Unmanned Helicopter Faults Diagnosis based on Petri Nets

Unmanned Helicopter Faults Diagnosis based on Petri Nets

Miguel A. Trigos[1], Antonio Barrientos[2], Jaime del Cerro[3]
Universidad Politécnica de Madrid, Madrid, España

Artículo recibido en mayo de 2016; artículo aceptado en junio de 2016

Citación del artículo: Trigos, M., Barrientos, A. & Del Cerro, J. (2016). Unmanned Helicopter Faults Diagnosis based on Petri Nets. I+D Revista de Investigaciones, 8(2), pp. 91 - 103

Abstract
This work presents a Fault Diagnosis application based on Petri Nets (PN) applied to a small unmanned helicopter. The first step of the research is the construction of the model and diagnoser for the Helicopter by using PN. A Data Acquisition System (DAQ) has been designed and built for providing the PN Diagnoser with data during the flights. Missions have been conducted with the aircraft configured to fly in both normal and fault operation. Thus, several common faults were intentionally generated during the test flights. This application allows the operator to perform the aerial vehicle health monitoring in order to prevent major damages in case of accident. Vehicle variables are monitored and thresholds adequate for the UAV defined. A summary of the validation results obtained during real flight tests are also included. An extensive use of this tool would allow improving preventive maintenance protocols for UAVs and establishing recommendations in regulations. UAVs accidents involve not only high economic cost but also serious restrictions for performing flights over populated areas. This work integrates Fault Diagnosis from theoretical and practical point of view. The use of the diagnoser by using Petri Nets is considered as novel approach.

Keywords: Fault Diagnosis; Petri Nets; LabVIEW; UAVs; Data Acquisition Systems DAQ; Helicopter.

1. Introduction

During 2007 -2008 more than 600 Unmanned Aerial System (UAS) from more than 250 different manufacturers have appeared in the market (Hughes, 2007). This growth and the new legislative trends (National RPAS Regulations, 2015) have required to extend the Fault Diagnosis (FD) techniques to the UAS fields. Moreover, not only the experience of any research team working with UAS but also data reported by Freeman (Freeman, 2014) about the reliability of UAVs (refers to alone aircraft) show that they are highly vulnerable to unforeseen situations. These can result from either equipment devices (control station and aircraft) or operational issues, highlighting in this manner the lack of methods to prevent the user from system breakdowns.

According to Hayhurst (Hayhurst et al., 2006), the dangers that may apply to an unmanned aircraft can be grouped in three main domains: the design domain, the domain of the flight crew (operators for UAS) and the operational domain. Collisions into terrain with or without collateral

[1] Ingeniero Electrónico, Universidad del Valle. PhD Automática y Robótica, Universidad Politécnica de Madrid. Investigador del grupo: Centro de Automática y Robótica. Universidad Politécnica de la ciudad de Madrid (España): Dirección, c/ José Gutiérrez Abascal, 2. 28006, PBX: +34 91 336 30 60. Correo electrónico institucional: matrigos@industriales.upm.es

[2] Ingeniero Industrial, Universidad Politécnica de Madrid. PhD Automática y Robótica, Universidad Politécnica de Madrid. Docente-Investigador del grupo: Centro de Automática y Robótica. Universidad Politécnica de la ciudad de Madrid (España): Dirección, c/ José Gutiérrez Abascal, 2. 28006, PBX: +34 91 336 30 60. Correo electrónico institucional: antonio.barrientos@upm.es

[3] Ingeniero Industrial, Universidad Politécnica de Madrid. PhD Automática y Robótica, Universidad Politécnica de Madrid. Docente-Investigador del grupo: Centro de Automática y Robótica. Universidad Politécnica de la ciudad de Madrid (España): Dirección, c/ José Gutiérrez Abascal, 2. 28006, PBX: +34 91 336 30 60. Correo electrónico institucional: jaime.delcerro@upm.es

damage to people and properties as well as mid-air collisions with other aircrafts are considered among these hazards. A classification of fault diagnosis and fault tolerant control methods for manned and unmanned helicopters is presented in (Qi et al., 2013). Moreover, FD techniques have been applied to many types of UAS, for example, (Davoodi et al., 2013) use a FD strategy for micro-air vehicles.

From the fault diagnosis point of view, most of the research works are focused on evaluating hardware failures located in the aircraft, i.e., sensors and actuators (Gaujens et al., 2014) (Wu et al., 2015), the proposed approach by (Jagadish and Chang, 2011) can identify the sensor failures in UAVs by mean of computing the Euler angles in multiple ways. On the other hand, few research efforts (Drozeski et al., 2005) are focus on identifying faults; seeking for a reconfiguration of the control system so as to bring the aircraft to a state of normal operation or, in the worst case, abort the mission. Most of the FD techniques are based on parameter estimation (Samar et al., 2006), Kalman Filter (Heredia and Ollero, 2011) or neural networks (Samy et al., 2010). In few cases, methods based on redundancy are proposed (Bateman et al., 2011).

In this work, Petri Nets have been applied for building a diagnoser, due to their ability to concurrently represent and analyze the system in an easy way while keeping the synchronization of the events (David and Hassane, 2010). Literature shows that Petri Nets are mainly used for modeling purposes in discrete event systems (Cabasino et al., 2010) (González and Cerrada, 2014), because they solve limitations of combinational explosion. Moreover, they are widely used in hybrid applications (Zhao et al., 2005) (Lefebvre, 2014). Nevertheless, they have been hardly applied in UAS applications (Liu et al., 2011).

Consequently this work, which proposes a FD algorithm based on Petri Nets (PN) applied to UAS (hybrid systems), can be considered as an innovative and significant contribution in this field. Moreover, a methodology for building FD algorithms based on Petri Nets, whose basis were previously introduced in (Trigos et al., 2009) is presented and evaluated in a real application.

Accordingly to this methodology, a FD diagnoser has been developed by splitting the aircraft into subsystems. Subsequently, a model of each component that takes into account normal and fault operations has been built. The construction process turns out to be quite simple and robust. Its main advantage is the simultaneous detection of faults as well as the flexibility to expand its application to other systems. As a main difficult found using the PN was the dependency in knowledge of the expert for build the net.

The evidences showed in this work are the result obtained while flying a helicopter in normal and fault conditions (forced), such as problems with vibrations and fuel supply system. Not only did the mentioned flights allow validate the FD tool but also setting some thresholds of system variables that were unknown so far.

The paper is organized as follows. The methodology of work carried out is presented in Section 2, which describes the algorithm designed to build the fault diagnoser, the Data Acquisition System (DAQ) used for applying the FD tool to UAV and the real flights framed under normal and fault conditions. Section 3 summarizes an analysis of the results obtained after applying the FD algorithm. Finally, Section 4 presents the conclusions of this research in the field of UAVs.

2. Methodology of work

2.1 FD Algorithm Applied to UAV

The basis of this work can be found in (Trigos and & Garcia, 2008), where PN were applied to fault diagnosis in Discrete Event System (DES's). The FD algorithm is a tool based on the Petri net model of each component of the system, a diagnoser (Petri Net) capable of assessing the state of system is constructed step by step, including the ability to diagnose intermittent faults.

Let $Q=(P,T,I,O,M_o)$ the PN that represents the discrete event model of the system to diagnose. P denotes the set of places, T represents the set of transitions, I/O is the input/output incidence application and M_o the initial marking. Transitions T can be classified as unobservable T_{uo} and observable T_o. A transition T is observable when is activated by a control event (e.g., command from supervisor or instrumentation), not observable concerns to transitions that happen but the diagnostic system is not able to detect normally. Fault transitions T_f are considered as unobservable transitions. They are classified into disjointed sets corresponding to the different types of failure that may occur in the system. It is imperative to distribute failures in groups to facilitate their identification to the diagnosis system. Therefore, the fault transitions set T_f is made up of different subsets of faults given in the process, $T_f=T_{f1}U\ldots UT_{fm}$.

The procedure to design a fault diagnoser (Figurefigure 1) can be summarized in several steps, according to the following methodology:

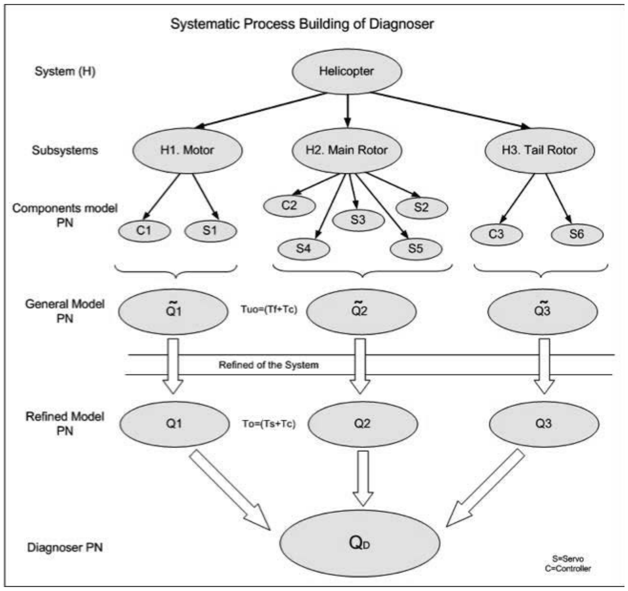

Figure 1.
Systematic building process of diagnoser.

Step 1. Classification in Subsystems. The system H has to be split into several sub-systems depending on their performances, thus $H=H_1UH_2U\ldots UH_n$. Although usually there are close relationships among them, this classification allows making better use of the FD algorithm.

Step 2. To build the PN model of the components of each subsystem, assuming that the system is split into N components, according to the expression:

$$Q\_i=(P,T,I,O,M_0) \tag{1}$$

$i=1,\ldots,N$, Q_i represents the PN of the i-component.

Step 3, Operation of Integration. Let $\tilde{Q}=(\tilde{P},\tilde{T},\tilde{I},\tilde{O},M_0)$ the denotation of the integrating operation of the PN models of N components of each subsystem. $\tilde{Q}$ refers to the representation the subsystem behavior by using a PN model, which includes different models of PN components. Since this model integrates the normal and fault behavior of the system, a transition can occur from any place of the model. Thus, a normal transition T_o or a failure transition T_{uo} can occur in any place of the PN. Let $\tilde{P}$ the union of individual places P_i, and $\tilde{T}$ the union of normal transitions T_o (i.e., transitions given by the supervisor or the process control system S) and the unobservable transitions T_{uo}.

$$\tilde{P}=\bigcup_i P_i \qquad \tilde{T}=\bigcup_i T_i \tag{2}$$

Step 4. To refine the General Model. It is necessary to consider only the observable part of Q, therefore, $\tilde{Q}=(\tilde{P},\tilde{T},\tilde{I},\tilde{O},M_0)$ has to be transformed to Q=(P,T,I,O). Thus, unobservable transitions are replaced by observable transitions. Moreover, considering that a place P is not achievable, when it will be never present by operating conditions of the system (i.e., marking the PN is not achievable) it can be written:

$$\forall p: p_i \in P \rightarrow M(p_i) \notin R(Q,M_o) \tag{3}$$

where $R(Q,M_o)$ is the set of all markings reachable system. The refinement is based on the construction of the integration table of the M sensors of the system. Thus, given a set of M sensors, the integrating sensors table $h_j=P\ \tilde{Y}_j$, $j=1,\ldots,M$, where Y_j denotes the discrete set of possible outputs for the j^{th} sensor, defined as follows:

$$Y=\prod_{j=1}^{M} Y_j \tag{4}$$

and $h=\tilde{P}\rightarrow Y$ denotes the integrating sensors table, defined accordingly to the following expression:

$$h(p)=\left(h_1(p),h_2(p),\ldots,h_M(p)\right) \tag{5}$$

Finally, the global model comprises the normal and fault places, $P=P_N+P_F$. Transitions include controller events S and resulting events from the integrating sensors table Σ, $T=SU\Sigma$. Accordingly, general model includes only observable transitions.

Step 5., Diagnoser. The diagnoser is a PN implemented considering the refined model of the system as starting point, conducting an on-line observation of the model, in order to perform a diagnostic on the system behavior.

Initially, it is required to define fault labels. $\Delta f=\{F_1,F_2,\ldots,F_m\}$, a label has to be assigned to each system fault. The set of labels Δ is made up of normal labels N and fault labels F, thus $\Delta=\{N\}U\{F\}$. Therefore, the diagnoser for system Q is a PN of the form $G_d=(P_d,T_d,I,O,P_0,t_0,t_{end})$. The sets of places, transitions, input and output arcs keep the same definitions of the PN, adding a starting place P_o, a starting transition t_o and an ending transition t_{end}. The starting place P_o always starts with the normal label. A transition t_o must be activated to start the PN diagnoser until t_{end} is manually activated to stop the diagnoser.

The set of places P_d of the diagnoser is an extension of the set of places of general model, a place p of G_d it is of the form (p_i,l_i) where places belong to observables places, $p_i \in P_0$ and the label l_i belonging to labels set Δ. The labels are of the form l_i={N} V {F}, a place P_d take the label of normal or fault operation.

An observer of Q provides an estimate of current location of the system. The label attached to each places indicates the status of the component (i.e., if it is in fault or normal mode).

The functions essential for building the diagnoser are:

Label Assigned function $LA: P_o \times \Delta \times T^* \rightarrow \Delta$. Thus given $p \in P_o$, $l \in \Delta$ and $s \in L(Q,p)$, LA assigns the label l over s (transitions sequences) starting from p and following the dynamics of Q, according to:

$$LA(p,l,s) = \begin{cases} \{N\}\ si\ \forall i | Tf_i \notin s| \\ \{F\}\ si\ \forall i | Tf_i \in s| \end{cases} \quad (6)$$

Sink places appear in the integration model Q. If marking fell into a sink place, the PN would be blocked. To correct this problem, the Fault Expanding function (FE) is created, taking advantage of the concurrence capabilities of the PN. Therefore, the Fault Expanding function is defined according to the following expression: $EF: R_N \times F_i \rightarrow R_F$ where R_N is the normal operating branch and R_F is the fault operating branch.

For each set of failure F_i, a new branch of failures in the PN will be created so as to fulfill the role of overseeing the failures individually. The diagnoser G_d will have as many branches as number of faults has the system. R_G denotes the total number of branches of the diagnoser.

$$R_{G_d} = \sum_{i=1}^{M} R_{f_i} \quad (7)$$

PN diagnoser evaluates possible changes in each branch. The diagnoser emerges in normal or failure operation according to LA function. The diagnoser evaluates each fault separately and takes into account in their transitions the failures that are caused by other failures, therefore it is able to detect failures simultaneously regardless the order in which failures occur.

An extended description of the basis of the FD algorithm and its implementation in UAV can be found in (Trigos et al., 2009). The final diagnoser algorithm is shown in Figure 2.

2.2 Rules to define the warning and fault states

After building a PN diagnoser, it is required to define the rules that allow determining if a warning or fault state can exist. These rules are based on the expertise of a proficient pilot, who is able to associate changes in the variables with their causes, providing the operator with the support to find the problem and act accordingly. These rules are defined by using two types of abnormal situations:

Warning, which is defined when a variable exceeds a failure threshold for a certain period of time (depending on the variables, a value of three seconds has been considered for high dynamic ones). If the variable returns to its normal value before this timeout, it will be not considered as a warning.

Fault, which is defined according to the following rules:

-Any warning that is activated and held for more than 5 seconds.

-Two simultaneous warnings occur and are maintained for more than 3 seconds.

-3 or more warnings occur simultaneously.

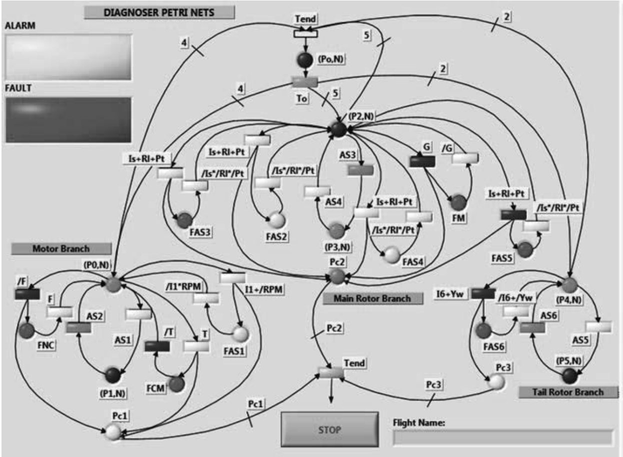

Figure 2.
Diagnoser Petri Net

Each variable, in isolation or together with one or more variables is classified according to their importance in the operation of the aircraft. The variables considered in the diagnoser are:

Voltage of the battery (V). Any jump in the voltage in minimum threshold of voltage becomes fault.

$$F_1 = \bar{V} \quad (8)$$

Attitude of the helicopter Ø (i.e., roll, pitch and yaw angles) as well as servo currents (Is) are signals that may exceed at any time their failure threshold,. The signals should exceed their value longer than the established timeout (3 seconds in this case);

$$F_2 = (\emptyset \cdot t) + (I_S \cdot t) \tag{9}$$

Temperature (T), Fuel Flow (FF), Revolutions per Minute (RPM) and Vibrations (G), are strongly related to each other in a helicopter, therefore they require a global analysis instead of an independent one. A truth table (table 1) has been constructed in order to define a set or rules that will be used to determine if the system falls into fault or not. An exception is considered on regard this rule: In the case that Fuel and Temperature variables overcome their failure threshold simultaneously, a warning is activated.

Based on the technique of Karnaugh maps, the equation for the fault in its simplest form is defined as follows:

$$F_3 = \bar{G}(\bar{T} + \bar{F} + \overline{RPM}) + \overline{RPM}(\bar{T} + \bar{F}) \tag{10}$$

According to the fault conditions defined in (8), (9) and (10), a global fault condition F can be created by using the OR operator:

$$F = \bar{V} + (\emptyset \cdot t) + (I_S \cdot t) + \bar{G}(\bar{T} + \bar{F} + \overline{RPM}) + \overline{RPM}(\bar{T} + \bar{F}) \tag{11}$$

Table 1
Discrete Performance Relationship of T,F,RPM,G.

T	*F*	*RPM*	*G*	*Fault*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0

A PN, which is in charge of representing the diagnoser, runs on real time during the flight and supports the pilot to detect a malfunction of the system in order to take the better decision. PN diagnoser identifies the number of times each variable overcomes the failure threshold, as well as the number of times that the system has fallen into fault for each subsystem (Motor, Rotor and Tail).

2.3. Data Acquisition System

A complete Data Acquisition System (DAQ) has been designed and built so as to implement the diagnoser. It is made up of two main components: the instrumentation onboard and the control station, which is responsible for presenting data and housing the DPN algorithm. The DAQ System is responsible for sensing the field variables of the helicopter and sending the information to the ground control station. This information is used to feed the fault diagnosis algorithm; which monitors the normal or fault operation of the aircraft by using the diagnoser petri net (DPN). Several variables were chosen to monitor the whole system according to the expertise of the researching group and pilot suggestions, namely:

- Engine temperature (T)
- Main battery voltage (V)
- Fuel flow (F)
- Engine revolutions per minute (RPM)
- Vibrations of the frame (G)
- Current (Is) of each servo that allows controlling the aircraft (i.e., throttle, four for the plate and tail)
- Attitude (roll (Rl), pitch (Pt) and yaw (Yw) angle)
- Position (latitude and longitude).

The general data acquisition scheme is represented in figure 3. Several sensors have been installed onboard. Each sensing module sends the information to a central computer based on a microcontroller. Once the data is collected, it is sent by radio to ground station by using an XBee communication system. The information is also stored onboard into a SD card (black box concept). The ground computer receives the info through an USB port in order to show it on the user screen by using an application developed in LabVIEW.

Currently, the diagnoser runs into the ground control station computer in order to maximize the flexibility of the system during the tests. Nevertheless it is planned to ship it in a CompactRIO module in order to reduce the bandwidth required and increase the acquisition frequency. In this manner, only fault and warning signals would send to ground station.

All other information (e.g., faults number, faults time, reading variables, etc.) will be used in subsequent work of preventive and predictive maintenance of the aircraft. A brief description of the main components of the data acquisition system is summarized in the table 2.

The DAQ system works in independently of the control system; it is fully autonomous and does not share any resources with the rest of the aircraft equipment. This allows using the system with any other UAV. Since there is a redundancy in the information (i.e., position and attitude of the vehicle), a simple comparison method could be useful to detect malfunctioning of the navigation systems.

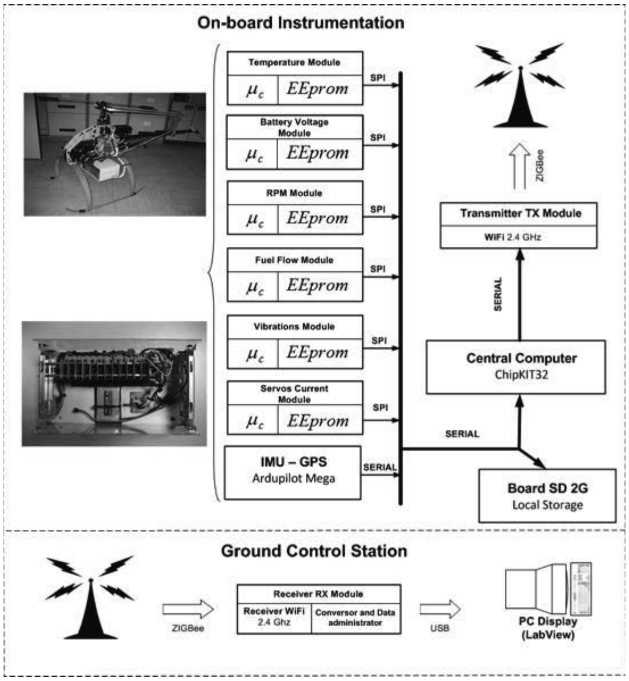

Figure 3.
General Scheme DAQ (UAV and Ground Station)

Table 2
Instrumentation table.

Variable/component	Sensor	Description
Temperature	ADS22100	Evaluates the engine temperature ranges from -50 to150 °C
Battery Voltage	No used	Is evaluated by directly in analog input. measuring the battery voltage
Revolutions per minute	UGN3503	Hall effect sensor
Fuel flow	FHK sensor	Measuring range from 0.041 to 6 L/min
Vibrations	accelerometer DE-ACCM2G2	Minimum is measured.
Servos Current	No used	It is measured through the supply voltage delivered by the current sensor and measured in mA
IMU - GPS	ArduPilot Mega 2.4 model	It provides information about both position and attitude of the aircraft.
Central Computer	microcontroller ChipKIT32	It performs data acquisition coming from the ArduPilot board by using the communication protocol MAVLINK
Transmitter/Receiver Module	XBee	It is the wireless interface between the aircraft and the Control station in the band of 2.4 GHz. A FTDI (Future Technology Devices International) module converts serial communication to introduce it into the computer via USB port.

Graphical Interface

As previously mentioned, a Labview graphical interface is responsible for acquiring, storing and displaying all data from the UAV. The data acquisition module is responsible for making data acquisition in real-time. Data is collected online and stored into a file for later use. The analysis and data representation module allows performing offline analyses of data from different missions. Finally, a module has been implemented in order to execute the Petri Net Diagnoser (PND), which is able to manage both real time and previously stored data from past missions. The PND registers the frequency with which any component on the aircraft falls into warning or fault.

Helicopter

For this study, a radio controlled helicopter model Vario Benzin Trainer was chosen, which relies on six servo-controllers for maneuvering. It has a length of 1,46 meters, a wingspan of 2 meters and a weight of 7.3 Kg. The Helicopter is made up of three main systems: motor system, the main rotor system (known as plate) and the tail rotor. If any of these three systems fails, the mission unavoidably has to be aborted, because the helicopter could crash with serious consequences rapidly.

The Motor System is made up of a small gasoline engine and a servo that controls the throttle. It is responsible for generating the rotation of helicopter blades (main and tail rotor in a mechanically-settle proportional relationship). An electronic controller is typically used to maintain the speed of the rotor constant independently of the attack angle of the blades. This angle is directly related to the drag that engine has to overcome.

The main rotor system is controlled by four servos. They allow modifying the lift and helicopter attitude (i.e., the roll and pitch angles) by varying the collective and cyclic pitch of the blades along their rotation. Thus, the collective pitch varies the lift and the cyclic one modifies the attitude.

The main component of the tail rotor system (anti-torque) is a servo controller, which receives commands from the pilot in order to modify the pitch angle of the tail rotor blades. The variation of this angle allows modifying the yaw angle of the helicopter, mainly because it varies the anti-torque that is required to compensate the one generated by the main rotor. .

The full system relies on additional devices, such as power supply, navigation sensors (i.e., inertial systems, tail gyro, GPS and electronic compass), controllers, communications (antennas and radios), and the ground control station.

2.4. Normal and Faults Flights

A series of field missions has been carried out in order to evaluate the performance of DAQ and PN diagnoser. Several lab tests were previously performed in order to set up the thresholds of the variables that determine the normal operation of the helicopter.

The faults generated were: To promote an increment in vibrations of the frame due to unbalances both in the main and tail rotor and to force problems in the engine carburetor, (i.e. the fuel flow was restricted to the engine).

Table 3
Discrete Performance Relationship of T,F,RPM,G.

ID	Mission	Duration	Description	Objective
1	Normal Flight	756 sec	Aircraft in optimal conditions	To verify that the diagnoser works properly in the different stages of the flight: idle (motor started on ground), takeoff, hovering, translation flights and landing
2	Flight with Vibrating Plate fault	181 sec	Flight with roll and pitch vibrations augmented	To force an over-vibration problem so as to check if diagnoser detect it
3	Flight with Tail vibrations Fault	238 sec	Flight with yaw vibrations augmented	To verify if vibrations due to different factors are detected by the diagnoser.
4	Flight with Fuel Mix Fault	657 sec	Flight with a wrong mix in gasoline	To verify that a problem in engine system is detected

3. UAV Behavior Analysis

It is worth noting that the test required to evaluate the performance of the diagnoser are extremely risky since they require forcing real failures during the flights. Table 4, shows a summary of the variables considered during the tests, including their thresholds for determining the normal and fault behavior.

Some difficulties are commonly found when working with real-time systems that have to be shipped on board of a vehicle with an extremely limited payload (e.g., data processing limitations, size of the information and consumption of processing elements). Additional problems arise when the information should be transmitted to ground station (noise, range of communications, limited bandwidth). Therefore, it is essential to be as pragmatic as possible, i.e. use the maximum sampling time and the essential discrete variables. Another important limitation is the reaction time of the pilot to decide if a mission has to be aborted when a warning/fault is detected considering the serious consequences that occur when an aircraft crashes into the ground. Thus, the UAS operator could observe a measured variable fall into alarm (warning) and in some cases into fault for short periods of time and recover the normal condition.

A summary of the conclusions observed in each of the variables are listed in the next section.

Table 1
Discrete Performance Relationship of T,F,RPM,G.

Variable	*Symbol*	*V. Normal (1)*	*V. Fault (0)*
Temperature	T	T amb - 90 ºC	100 ºC
Voltage	V	6,5 - 7V	0 - 6 V
Fuel Flow	F	100 - 500 mL	0 - 50 mL
RPM's	RPM	1200 - 1500	< 1200
Vibrations	G	0,5 - 2,5 G	> 2,5 G
Servos Current	Is	100 - 600 mA	800 mA
Roll/Pitch/Yaw	Ø	Rl/Pt/Yw	Rl/Pt/Yw
Angles		< 45º	> 45º

3.1 Analysis of the behavior of monitored variables during UAV Flights.

Engine Temperature maintains a value between ambient temperature and a maximum of 90 °C during normal operation, with a standard growing of 2.1 ºC/sec transient, although this value slightly depends on the environment in which the mission takes place. Either higher or lower values reveal problems during the flight. Thus, a value below the threshold usually points out that the engine may be getting more gas injection than necessary, but the contrary (i.e., a higher value) indicates problems in the engine such as poor combustion. A poor combustion may have different causes such as a lose of denotation synchronization, no optimal carburizing (low gas) and, in some cases, a poor fit of the engine output with respect to the pitch of the blades (wrong setup of the compensating tables in RC emitter). In this situation (i.e., limit exceeded) the mission should be aborted.

Moreover, considering the fault conditions generated in the test flights, it was shown that the delta temperature increases when the mix fault is generated, as see figure 5d highlights, but keeps normal values when the aircraft suffers from a vibration fault, as figures 5b and 5c illustrate.

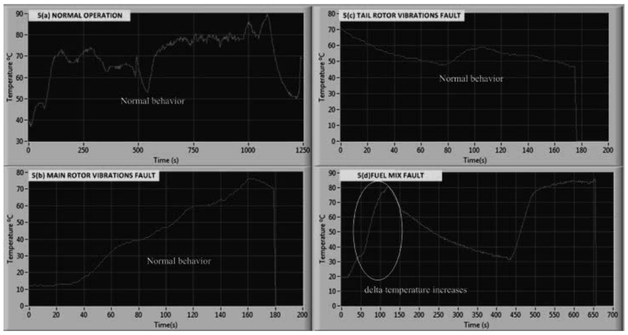

Figure 5.
Normal and Fault Temperatures in UAV

Battery Voltage. Usually, two power systems are used in large UAVs, namely a battery and a system generator. This generator, which is driven by the main engine, is responsible for generating the system voltage. Additionally, a voltage regulator that relies on multiple outputs provides the different voltages to the aircraft. When the aircraft does not rely on a generator, the battery has to be recharged before flying. The charge of the battery depends on the time of use and the number of charging cycles that has previously suffered. During the flights performed for system validation, the voltage of the battery turned out to be quite stable and in no case showed voltage drops.

A 6.5 volts threshold was setup in the system used in this work as minimum voltage allowed in operation. This value may differ depending on the technology of battery used and its variations in the discharge curve, therefore should be customized in any case.

Fuel Flow. The helicopter used in this work relies on a tank with a capacity of 500 mL approximately, which allows flying approximately 20 minutes. This time varies depending of the payload and altitude. There are three ways to track the storage and consumption of fuel in the aircraft: by using a low level sensor (discrete), which only detects that there is a minimum level that ensures the safety of mission; by measuring the pressure in the circuit (this allows detecting a leak in fuel hoses and connections) and finally by measuring the fuel flow in an analog way. This last solution was chosen in this work mainly because it allows detecting more malfunctioning conditions that previous ones. Therefore, a sensor was installed in the supply pipe.

Flight tests have allowed to define a threshold of 100 mL for the minimum fuel storage. Moreover, the test assessed that the fuel flow is affected in any fault of the engine system. Thus, the consumption usually varies significantly when the system requires major effort to the engine. According to this feature, an average fuel consumption of 0.50 mL/sec was settled for normal flight conditions. Furthermore, it was established that a fault in the engine system is detected if a consumption over 1 mL/sec is maintained for more than 3 sec.

Rotor speed. It is usually measured by counting the revolutions of the plate in a period of time. Its value can vary between upper and lower limits. A steady decrement in the revolutions indicates a mechanical failure due to any of the following parts: engine, transmission, gear or clutch among others. This variable is within the range of 1250-1450 RPM in normal flight for the helicopter used during the tests. Nevertheless, the speed exceeds 2000 RPM during taking off and landing maneuvers, as figure 6a shows. In vibration fault conditions, the pilot needs more time to reach the flight stability; therefore, the number of revolutions goes out of its normal range and oscillates at a higher frequency as figures 6b and 6c show. Moreover, it can be observed that the motor makes an additional effort, (i.e., the pilot has to increase the engine speed) to reach the normal flight. In the fuel gas fault, the range may vary during take-off to values near 1400 RPM and it changes to 1100-1200 RPM once in flight as figure 6d depicts.

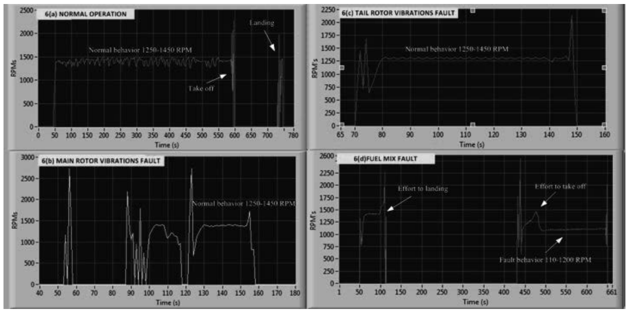

Figure 6.
Normal and Fault Revolutions per Minute in UAV

Vibrations of the frame. The behavior of the vibrations of the frame strongly depends on the type of maneuver (idle, takeoff, landing, hovering, and translation flight). These vibrations are measured in amplitude and their values are recorded. It's measured in the vertical axis amplitude. When the aircraft is turned off and there is no motion, this amplitude shows a constant value of 1g, due to the gravity, whereas value becomes larger if the aircraft is idle (motor started on ground). The values during normal flights (with amplitude near to 2g) are smaller than when idle (near to 2.5g).

Nevertheless, a considerable increment in the amplitude of vibrations is detected, (figures 7b and 7c), when the vibration fault is forced in the main rotor, with values similar to the ones when idle, reaching values of 2.3 G.

This comparison of amplitudes can be appreciated in figure 7. Fuel mix fault does not affect the vibrations behavior in the aircraft, as figure 7d demonstrates.

Servo Current. In practice, there are two ways to detect if a servo is damaged by reading its electrical consumption, thus, if the servo current is zero, the measurement denotes some kind of disconnection, whereas a steady increment in consumption usually indicates that the servo is not able to reach the reference commanded by the control. This situation usually highlights a mechanical jamming. As previously mentioned, the consumption of each of the six servos is monitored during the flights. No substantial changes were observed in the current consumption of the servos during the tests, neither in normal flights nor in failure ones. The normal consumption of servos is the range of 100 to 600 mA. Nevertheless, during the missions with fault (e.g., vibration and fuel mix failure), more maneuvers of the pilot were required to stabilize the helicopter, causing a higher number of changes in the current consumption of the servos.

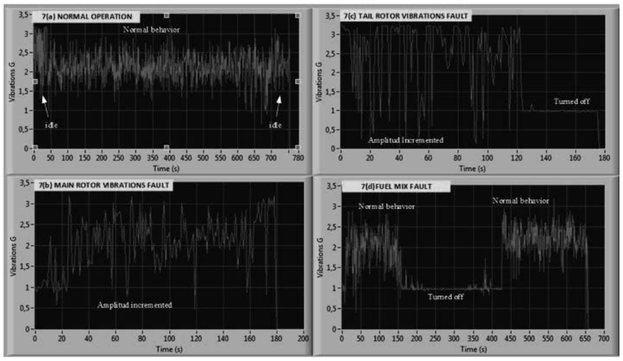

Figure 7.
Normal and Fault Vibrations in UAV

Attitude of the helicopter. Roll, pitch and yaw angles are registered in order to monitor the attitude of the helicopter. Furthermore, the radio emitter used during the test flights delimits the maximum plate angles allowed, which limits the expected roll and pitch angles under a 45 degrees threshold. Therefore, higher values are considered as a fault.

After evaluating the previously mentioned variables in the aircraft during the flight tests, it can be concluded that some of them (i.e., vibrations, speed of the rotor, fuel consumption and engine temperature) deserve special mention over the others. Moreover, the helicopter safety strongly depends on its suitable mechanical performance, and the expertize of the pilot is the best indicator to decide whether a mission should be aborted or not.

3.2 Results of Real Flights

1. Normal Flight Analysis: As figure 8 shows, although the RPM and vibration thresholds overcome intermittently their failure threshold during the flight, no warning was activated. Namely, the diagnoser PN found that RPM and vibrations variables exceeded their normal operation threshold in 6 and 93 times respectively. For more details about the flight in optimal conditions see the table 5.

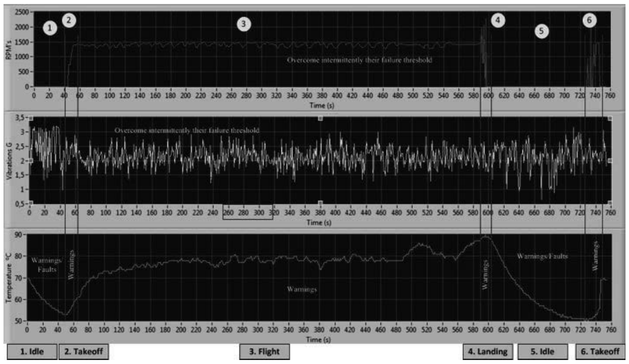

Figure 8.
UAV variables flying in normal conditions.

Table 5
Behavior of the helicopter in flight of optimal conditions

Stage	*Descripción*	*Behavior*	*Source*
1,5	Idle (Motor started UAV on ground)	Warnings / Faults	Vibrations / RPM
2,6	Takeoff	Warnings	RPM
3	Flight (Hovering, Translation)	Warnings	Vibrations
4	Landing	Warnings	Vibrations / RPM

2. Flight with Vibrating Plate fault: Small strips of tape were added to the blades of the main rotor to force an imbalance of the blades, which results in a loss of synchrony when they rotate at high speed. This therefore increases significantly the vibrations in the aircraft. Since the assumed risk was very high, only short flights were carried out under these circumstances. In fact, the performed flight caused problems in the silencer (i.e., it loosened repeatedly and in a certain moment tail rotor was lost). As figure 9 shows, during short parts of the flight, the vibration amplitude exceeded its threshold and it was impossible to sense a time of normal behavior in the aircraft. Although the speed of the rotor in flight is under normal conditions, the temperature increases quickly. Vibrations fell in fault 28 times and the value of RPMs did it in 10 occasions. More details about the flight in fault vibrations plate are provided in table 6.

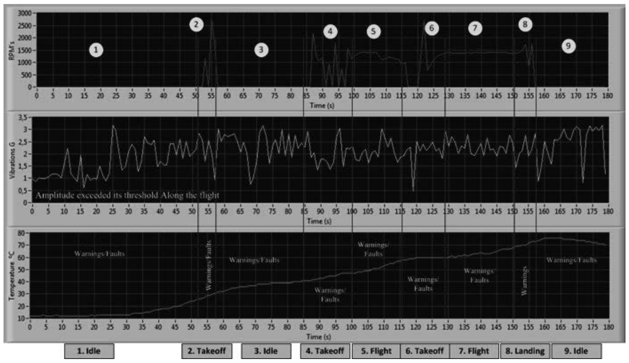

Figure 9.
UAV variables Flying in main rotor vibrations fault.

Table 6
Behavior of the helicopter in flight fault vibration plate.

Stage	*Descripción*	*Behavior*	*Source*
1,3, 9	Idle (Motor run in ground)	Warnings / Faults	Vibrations / RPM
2, 4, 6	Takeoff	Warnings / Faults	Vibrations / RPM
5, 7	Flight (Hovering, Translation)	Warnings/ Faults	Vibrations
8	Landing	Warnings	Vibrations / RPM

3. Flight with Tail vibrations Fault: The procedure to create the vibrations was similar to previous case but the tape was applied to the blades of the tail rotor. The risk in this case is even higher than previous one, since the vibrations in tail rotor are amplified due to the length of the tail. Moreover, the vibrations directly affect to the tail scope, creating additional problems for the heading stabilization, therefore, only short flights were carried out. In this mission, the pilot made big corrections to elevate the aircraft. As shown in figure 10, the vibrations are out of their normal threshold and the PN Diagnoser informs the pilot of a fault continuously. The rotor speed was under normal behavior, although an abrupt change at the time required to takeoff and landing was obtained. The PN Diagnoser detected that engine subsystems failed 3 times and the main rotor subsystem failed 26 times. The most critical variable was the vibrations with 26 drops, followed by the motor temperature that, although did not fall into fault, suffered an increment in its slope due to the efforts demanded to the engine by the abrupt commands of the pilot. Table 7 provides the reader with more details about the flight.

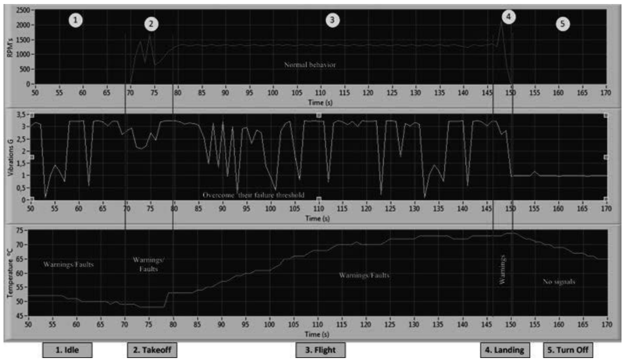

Figure 10.
UAV variables Flying in tail vibrations.

Table 7
Behavior of the helicopter in flight fault vibration tail.

Stage	*Descripción*	*Behavior*	*Source*
1	Idle (Motor run in ground)	Warnings / Faults	Vibrations / RPM
2	Takeoff	Warnings / Faults	Vibrations / RPM
3	Flight (Hovering, Translation)	Warnings/ Faults	Vibrations
4	Landing	Warnings	Vibrations / RPM
5	Turn off	No signals	

4. Flight with Fuel Mix Fault: This fault was generated by closing the inlet manifold. This affects the input of the fuel into the carburetor of the Helicopter. During the first part of the flight, the pilot had to force the engine to obtain revolutions enough to perform hovering, consequently the silencer went out of its place, and the aircraft had to be reassembled and the flight re-started. During this flight, the speed of the rotor went out of their normal threshold and the vibrations raised more often of normal behavior. Accordingly, the engine temperature was also affected and its slope increased as figure 11 illustrates. The PN Diagnoser detected a failure in engine subsystem 25 times, the subsystem Main Rotor failed 53 times. The most critical variables were the speed of the rotor and the vibrations, followed by temperature that, although did not fall in fault, its slope was increased by the high number of maneuvers required to perform the flight. More details about the flight in fuel mix can be found in table 8.

Table 8
Behavior of the helicopter in flight fuel mix fault.

Stage	*Descripción*	*Behavior*	*Source*
1	Idle (Motor run in ground)	Warnings / Faults	Vibrations / RPM
2,6	Takeoff	Warnings / Faults	Vibrations / RPM
3	Flight (Hovering, Translation)	Warnings/ Faults	Vibrations
4,8	Landing	Warnings	Vibrations / RPM
5	Turn off	No signals	
7	Flight (Hovering, Translation)	Warnings / Faults	Vibrations

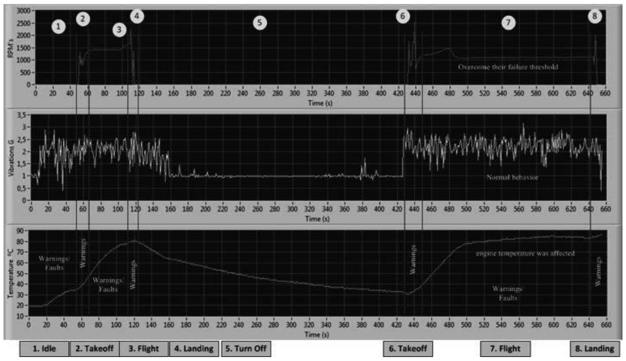

Figure 11.
UAV variables Flying in fuel mix fault.

4. Conclusions

This work summarizes the results obtained by using an innovative method, which consists in a Petri net diagnoser, to perform the health monitoring of a small helicopter.

The design process has been fully presented, including the building process, rules and variables involved in the diagnoser. A thorough analysis that justifies the variables selected to be monitored and their connections with the most frequent causes of failure in helicopters has been also included. Moreover, the data acquisition system is briefly described.

A special section dedicated to analyze the different flights carried out to validate the fault diagnosis system by forcing some malfunctioning into the aircraft has been included into this work. It must be stressed that forcing several failures during flights turned out to be extremely risky and therefore difficult to repeat intensively. Nevertheless, a valuable amount of flights have been carried out in order to assets the performance of the algorithm and methodology.

Finally, the performed tests have demonstrated that the Petri net diagnoser is able to detect not only permanent but also intermittent failures, providing the pilot with a set of health monitoring warnings. This can be considered the main contribution of this work, since no previous work in the literature shows the use of this kind of diagnoser based on Petri nets; reducing the combinational explosion of the factors took in consideration. Furthermore, the use of a commonly used tool such as Labview allows providing the diagnoser with additional features that turn out of great importance for preventive and predictive maintenance. Thus, using continuously this tool allows detecting gradual faults by comparing the values of the monitored variables in time to detect degradations.

As future work, a spectral analysis of vibration is considered, aiming at finding more types of mechanical damage, generation of mathematical models of normal and fault operation of the aircraft, and application of other FD techniques using data obtained from this research.

References

Bateman, F., Noura, H. and Ouladsine, M., (2011) "Fault Diagnosis and Fault-Tolerant Control Strategy for the Aerosonde UAV," Journal of Aerospace and Electronic Systems, IEEE Transactions on, vol.47, no.3, pp.2119,2137, July 2011. doi: 10.1109/TAES.2011.5937287

Cabasino, M., Giua, A., and Seatzu, C., (2010). "Fault detection for discrete event systems using Petri nets with unobservable transitions". Journal of Automatica 46, 9 (September 2010), 1531-1539. DOI=10.1016/j.automatica.2010.06.013 http://dx.doi.org/10.1016/j.automatica.2010.06.013

David, R, and Hassane A., Discrete, Continuous, and Hybrid Petri Nets. Springer Science & Business Media, 2010.

Davoodi, M.R., Khorasani, K., Talebi, H.A. and Momeni, H.R., (2013) "A robust semi-decentralized fault detection strategy for multi-agent systems", International Journal of Intelligent Unmanned Systems, pp 21-35, Doi 10.1108/20496421311298125, 2013, available in: http://www.emeraldinsight.com/doi/abs/10.1108/20496421311298125.

Drozeski, G., Saha, B. and Vachtsevanos, G. (2005), "A fault detection and reconfigurable control architecture for unmanned aerial vehicles", Proceedings of Aerospace Conference, 2005, IEEE. DOI: 10.1109/AERO.2005.1559597, 2005.

Freeman, P. M. (2014). Reliability assessment for low-cost unmanned aerial vehicles (Order No. 3684180). Available from ProQuest Dissertations and Theses A&I: The Sciences and Engineering Collection. (1659817259). Retrieved from http://search.proquest.com/docview/1659817259?accountid=14712.

González, O. and Cerrada, M. (2014), "Diagnosis of controlled discrete event systems: An approach based on chronicles and modular analysis by using automata models". Revista Iberoamericana de Automática e

Informática industrial 11 (2014) 191–201 Vol. 11. Núm. 02. Julio 2014.

Hayhurst, K., Maddalon, J. and Miner, P. (2006), "Unmanned aircraft hazards and their implications for regulation", Proceedings of the 25th Digital Avionics Systems Conference, NASA Langley Research Center, Hampton, Inc., Eastsound, WA, October 2006.

Hughes, David, February 12, (2007), "A Second Kitty Hawk?" Aviation Week and Space Technology, pp. 49-55.

Jagadish, C., Chang, B.-C., (2011). Fault-tolerant attitude computation for unmanned aerial vehicles. Transactions of the Institute of Measurement & Control 33, 752–779. doi:10.1177/0142331209342210.

Liu, J., Chen, K. and; Wang, Z. (2011), "Fault analysis for flight control system using weighted fuzzy petri nets". Journal of Convergence Information Technology, Volume 6, Number 3. March 2011.

Gaujens, A., A. Benini, A. Mancini, and S. Longhi., (2014) "Testing of Cooperative Tasks for Unmanned Aerial and Ground Platforms." Proceedings of In 2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), 1–6, 2014. doi:10.1109/MESA.2014.6935621.

Heredia, G, and Ollero. A, (2011), "Detection of Sensor Faults in Small Helicopter UAVs Using Observer/ Kalman Filter Identification." Mathematical Problems in Engineering 2011 (September 21, 2011): e174618. doi:10.1155/2011/174618.

Lefebvre, D., (2014) "On-Line Fault Diagnosis With Partially Observed Petri Nets," Automatic Control, IEEE Transactions on, vol.59, no.7, pp.1919, 1924, July 2014. doi: 10.1109/TAC.2013.2294617.

National RPAS Regulations. (2015), available online: https://www.eurocontrol.int/articles/national-rpas-regulations (accessed on 23 January 2015).

Qi, X., Theilliol, D., Qi, J., Zhang, Y., Han, J., Song, D., Wang, L. and Xia, Y. (2013), "Fault diagnosis and fault tolerant control methods for manned and unmanned helicopters: A Literature Review". Proceedings of Conference on Control and Fault-Tolerant Systems (SysTol). Nice, France, October 9-11, 2013.

Samar, S., Gorinevsky, D. and Boyd, S. (2006), "Embedded estimation of fault parameters in an unmanned aerial vehicle", Proceedings of the 2006 IEEE International Conference on Control Applications, Munich, Germany,, October 4-6, 2006.

Samy, I.; Postlethwaite, I.; Gu, D.-W.; Fan, I.S., (2010) "Detection of multiple sensor faults using neural networks- demonstrated on a unmanned air vehicle (UAV) model," Control 2010, UKACC International Conference on , vol., no., pp.1,7, 7-10 Sept. 2010. doi: 10.1049/ic.2010.0403

Trigos M., Barrientos, A., Del Cerro, J. and; Lopez, H. (2009), "Modelling and fault diagnosis by means of petri nets. Unmanned aerial vehicle application", In Book Petri Nets, Theory and Applications, Editorial IN-TECH Austria. 2009. pp 353-378.

Trigos, M. and Garcia, E. (2008), "Fault diagnosis and modelling of the liquids packaging process. A research based on petri nets", in Proceedings of 10th IEEE International Conference on Control, Automation, Robotics and Vision, Hanoi – Vietnam. 2008.

Wu, C., Qi, J., Song, D., Qi, X. and Han, J., (2015). Simultaneous state and parameter estimation based actuator fault detection and diagnosis for an unmanned helicopter. International Journal of Applied Mathematics and Computer Science, 25(1), pp. 175-187. Retrieved 8 Apr. 2015, from doi:10.1515/amcs-2015-0013

Zhao, F., Koutsoukos, X., Haussecker, H., Reich, J. and Cheung, P. (2005), "Monitoring and Fault Diagnosis of Hybrid Systems", Proceedings of IEEE Transaction, Actions on Systems, Man, and Cybernetics-: Cybertnetics, 2005. Vol. 35, no. 6.