Reliability Engineering and System Safety 193 (2020) 106609

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Event-based diagnosis of flight maneuvers of a fixed-wing aircraft

Alejandro White[a], Ali Karimoddini*,[b]

[a] CCDC Army Research Laboratory, Vehicle Technology Directorate, U.S. Army Research Laboratory, United States

[b] Electrical and Computer Engineering Department, North Carolina A&T State University, United States

ARTICLE INFO

Keywords:
Fault diagnosis
Flight control
Discrete event systems

ABSTRACT

Despite all efforts to improve the reliability of aircraft, the possibility of failure occurrences cannot be completely eliminated, and hence, airplane safety forever remains as a post-developmental concern. In times of peril, flight crews have a minimal amount of time to detect and determine an occurred fault. An automated fault diagnosis tool can help timely detect occurred faults and increase the flight crew's odds of recovering the plane before a possible crash. As a solution, this paper proposes a diagnosis tool that provides diagnostic information based upon the pilot's corrective inputs to level an aircraft's wings. For this purpose, this paper models a fixed-wing aircraft as a discrete event system (DES). The constructed model is then used to develop a diagnosis tool, a so-called diagnoser, for detecting, isolating, and identifying fault occurrences in the ailerons (left and right) of the aircraft. The developed diagnoser can be asynchronously activated during the aircraft's operation and provides diagnostic information in real time.

1. Introduction

Measures such as the rigorous process of certification for the airworthiness of an aircraft, the assemblage of the Federal Aviation Administration (FAA), the improved reliability and readability of aircraft flight instruments, and robust flight crew training have had a positive impact on flight safety. Given all of the aforementioned efforts for flight safety, faults in powered flight still can occur. These faults can lead to an upset airplane flight mode that unless properly diagnosed and accommodated, might lead to the crashing of the aircraft. Survivability of an upset aircraft is directly correlated to the competence of the flight crew's diagnosis and chosen recovery actions [32]. The relationship between survivability and crew training impacts certain aspects of pilot certification carried out by the Flight Standards District Office (FSDO). This relationship has also motivated the release of FAA supported auxiliary literature such as the *Airplane Flying Handbook* [3], the *Pilot's Handbook of Aeronautical Knowledge* [4], the *Airplane Upset Recovery Training Aid* [32], and guidance material for the implementation of upset prevention and recovery training [7].

During training, a flight instructor prepares the pilot to be able to execute basic flight maneuvers at a level acceptable by the FAA Practical Test Standards (PTS) [4], or by the Airman Certification Standards (ACS)[4]. A pilot must be able to perform these basic flight maneuvers under two types of meteorological conditions: visual meteorological conditions (VMC) and instrument meteorological conditions (IMC). Under visual meteorological conditions (VMC) (e.g., clear sky in the daytime), the pilot is able to fly "visually" and may reference the earth's natural horizon for flight guidance. When flying under instrument meteorological conditions (IMC) (e.g., night over dark ocean or dense fog), the pilot must fly the plane solely by referencing the aircraft's mechanical or electronic instruments. When flying under VMC, it is suggested that only 10% of the pilot's attention be directed inside the flight deck, and 90% of the time, the pilot should be referencing outside of the flight deck [23,24]. Beginner pilots are to get a feel for the plane, whether it is the sound the plane engine makes during cruise flight versus climbing or diving, or the feel of different forces the pilot experiences transitioning from level flight to a banked turn, etc. This training of site and feel helps to provide the pilot with a deeper understanding of the pilot's control actions, and how they affect the attitude and flight pattern of the plane. A pilot's control inputs are based upon an understanding of what the outcome will be. This is what makes flying under IMC conditions more challenging. Sometimes there is a lag in the output of the instruments. It is stated that *most flight failures are survivable if correct responses are made by the flight crew* [2,10,32].

This highlights the responsibility of the flight crew during the flight failures to properly diagnose and react to the occurred failures to recover the plane. However, since fault occurrences during a flight are rare, even the most prepared and well-trained flight crews may become startled upon experiencing upset flight conditions. This many times

* Corresponding author.
E-mail addresses: apwhite3@aggies.ncat.edu, alejandro.p.white2.civ@mail.mil (A. White), akarimod@ncat.edu (A. Karimoddini).

https://doi.org/10.1016/j.ress.2019.106609
Received 18 November 2018; Received in revised form 10 June 2019; Accepted 11 August 2019
Available online 23 August 2019

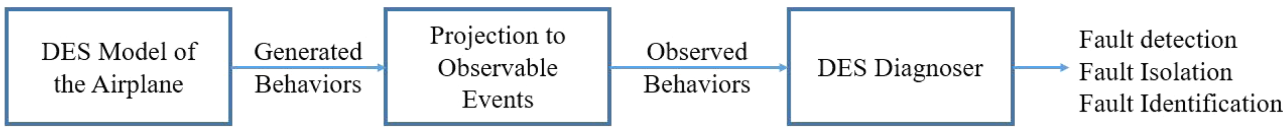

Fig. 1. The general structure of the DES fault diagnosis process.

causes the pilot to react before fully analyzing the situation. In some cases, the fault may only be diagnosed by an in-flight visual inspection by a flight crew member going and looking at the wing, as it was the case for a stuck aileron in Continental Airlines Flight 1659 [39]. When flying strictly using instruments, fatigue and spatial disorientation may play a factor in the pilot misinterpreting instrument readouts, or produce sensory illusions where the pilot erroneously determines the aircraft's attitude due to the pilot's inability to correctly orient him/her self with the aircraft. Fatigue and spatial orientation are believed to have played a role in the fatal crash of Air Transport International, Inc., Flight 805 [38]. These examples highlight the fact that there is always the need for a tool to automatically diagnose occurred faults and aid the pilots to promptly react to correctly accommodate faults.

There are different techniques for diagnosing faults including (but are not limited to) fault tree analysis [6,28], model-based approaches [16,20,48], development of expert systems [54,56], template structures [33], [19], and bayesian networks [29]. Applying the diagnosis techniques to flying vehicles, [21] reviews the techniques that are used in the Airbus 380, which are primarily based on threshold monitoring of residual signals combined with logical conditions. In [23], a model-based diagnosis technique is developed by employing an extended Kalman filter (EKF) to diagnose malfunctioning of an airspeed sensor when wind speed and propulsion dynamics are unknown. [8] adopts a sliding-mode observer method to diagnose sensor bias faults in inertial measurement units (IMUs) of a UAV. [55] develops nonlinear adaptive estimation technique for diagnosis of an aircraft engines. [41] develops a set-valued observer (SVO) for diagnosing sensor and actuator faults of an aircraft. In [27], a model-based technique is employed based on multiple linearized models, which together approximate the nonlinear dynamics of an aircraft to diagnose its actuators' faults. [43] and [45] use a sliding mode technique and [10] employs a signal processing approach to diagnose UAVs' actuators. Other examples include the use of EKF for diagnosing actuator failures [5] and residual-based fault diagnosis of control surfaces [13]. A comparison of diagnosis techniques for small aircraft is provided in [47]. While these techniques provide a great deal of progress toward enhancing the safety of aircraft systems, in most of these techniques the focus of the developed diagnosis techniques is on fault diagnosis of a single component of the aircraft. Further, most existing methods apply model-based or data-driven techniques to continuous/discrete-time signal readings. The fact is that complex engineered systems are usually composed of interconnected subsystems with hybrid structures which are monitored and regulated by both discrete event-driven logical rules and (continuous/discrete) time-driven estimators and controllers. Therefore, in addition to the aforementioned time-driven techniques, alternative complementary diagnosis techniques are required to infer fault occurrences in a system from observation of its high-level event-driven behaviors, governed by discrete logical rules embedded in its decision-making unite.

A very effective framework to capture event-driven dynamics of a system, particularly its high-level decision-making, is Discrete Event Systems (DESs) framework [15,37]. DES can effectively model a complex system in an abstract way, as opposed to other commonly used methods (e.g., difference/differential equations). Unique about the DES framework is that it naturally captures faults as abrupt changes (events) in the system, which facilitates the analysis of the system's faulty behaviors. More importantly, the structure of a DES model is similar to the human cognitive process in correlating the systems' interactions and the effect(s) of sequences of events. This will help the pilot or the autopilot system to properly manage normal/faulty situations of the plane toward a desired/safe sequence of events/actions.

Within DES framework, there are different diagnosis approaches ranging from off-line [31] to automated on-line [44] techniques. Further, it is possible to extend DES diagnosis technique to different decentralized [36,49] or modular/distributed [17,46] architectures. In addition, robust and safe techniques exist that can handle imperfect communications [14,30,34] as well as uncertainties in a system or its initial conditions [25,50–52]. A comprehensive review of fault diagnosis techniques for discrete event systems can be found in [26,53]. Despite the fact that DES provides an effective, abstract, and manageable tool for the diagnosis of complex systems, applying the DES framework to the modeling and analysis of practical systems is limited to a few cases such as power transmission networks [9], automated manufacturing systems [35], and communication networks [11,12].

This paper, therefore, derives a DES model for the flight mechanism of an aircraft, which is an enabling step toward its fault diagnosis. The derived DES model will be able to explain normal and faulty behaviors of the aircraft and can be used for analyzing fault events. We will use this model to build a diagnosis tool, a so-called diagnoser, that can detect the occurred faults and identify their types and their location in the system in a timely manner. Fig. 1 shows the general structure of the proposed DES diagnoser. As a model-based technique, the developed diagnosis tool requires a model of the system including its faulty and normal behaviors. Once derived, the developed diagnoser looks at the external (observable) behaviors of the plane (modelled by natural projection to the observable event set of the system) and diagnoses the faulty behaviors which involves fault detection (detecting the fault occurrences), fault identification (identifying the type and nature of the occurred faults), and fault isolation (locating the occurred faults). Moreover, the online implementation of the developed diagnoser is investigated. Unique about the developed diagnoser is that it is not required to be activated synchronously with the control structure of the plane. This will be achieved using only the plane's behaviors, observed after diagnoser activation, enabling the diagnoser to be asynchronously activated to estimate the status of the system whether it is normal or faulty. Moreover, if for any reason, the diagnoser misses an observation, it can be restarted to be able to track the plane's behaviors, without requiring the synchronous resetting of the control structure of the plane itself. The developed diagnoser is capable of diagnosing faults, which are included in the derived DES model. If a new fault is introduced to the system, a new model is required to revise the developed diagnoser.

The rest of the paper is organized as follows. Section 2 discusses the information about basic flight maneuvers that a fixed-wing aircraft may conduct including the straight-and-level flight, and right and left bank turns. Section 3 provides the necessary background about the discrete event systems modeling and develops a DES model for fix-wing flight maneuvers. In Section 4, a DES fault diagnoser is developed for a fixed-wing aircraft to detect and isolate the occurred faults in its flight-maneuvering mechanisms. Section 5 discusses online implementation of the developed fault diagnoser, and Section 6 concludes the paper.

2. Basic flight maneuvers

2.1. Rotation axes

An aircraft has three axes of rotation. These axes are the lateral axis,

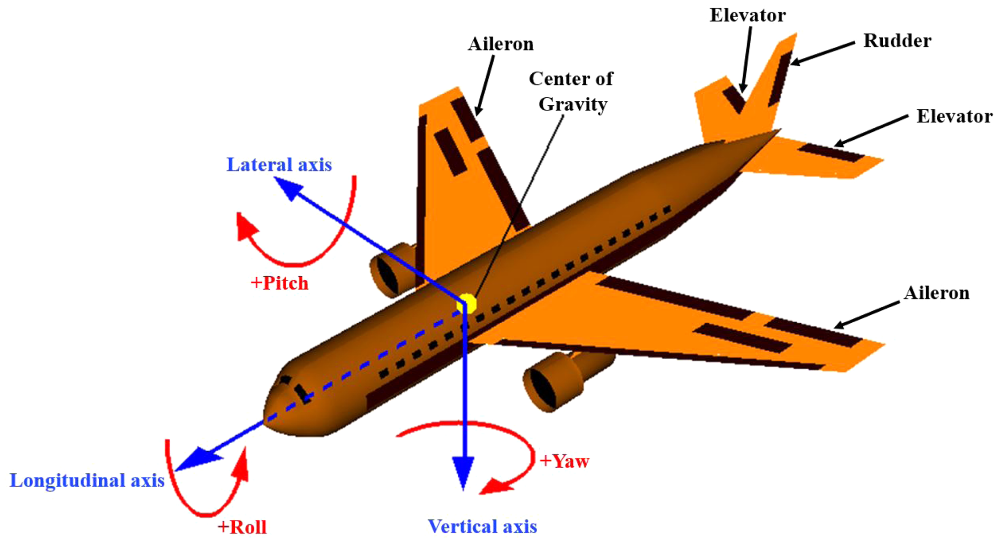

Fig. 2. Aircraft axes of rotation and control surfaces.

the longitudinal axis, and the vertical axis, whose origin is located at the center of gravity of the plane, forming together the body frame as shown in Fig. 2. The lateral axis is parallel to the wingspan of the plane and is directed towards the right wing tip. The longitudinal axis is parallel to the aircraft's fuselage and is extended from the tail to the nose of the aircraft. The vertical axis of the aircraft passes vertically through the aircraft's center of gravity and is directed towards the bottom of the plane. The angle formed between an aircraft's particular axis of rotation and the artificial horizon determines an aircraft's attitude. Using the gyroscope, an aircraft's pitch attitude is measured as the angle between the longitudinal axis and the artificial horizon, and the bank attitude is measured as the angle between the lateral axis and the artificial horizon. Yaw attitude is measured as the angle of rotation about the vertical axis, relative to the magnetic north direction. Movable airfoils, called control surfaces are utilized to orient the attitude of an aircraft.

An aileron is a control surface that is typically hinged horizontally to the trailing edge of an aircraft's wings. Ailerons are used to rotate the aircraft about its longitudinal axis (roll). The rudder is a control surface that is typically hinged vertically to the aft end of an aircraft and is used to rotate the aircraft about its vertical axis (yaw). An elevator is a control surface, which is typically attached horizontally to the aft end of the aircraft that rotates the aircraft about its lateral axis (pitch).

Relative wind is the air that flows opposite to the direction of aircraft movement. When the aircraft is flying, the relative wind produces various pressures on the control surfaces. The pressure distribution over the control surfaces (aileron, rudder, elevator) due to the relative wind flow is influenced by the speed and density of the air moving over the control surfaces. To produce a desired moment of force on the aircraft, the pilot will either leave a control surface in its neutral streamlined position or deflect the control surface to a desired degree of angle. In this paper, the airspeed, measured in knots (kn), will be the reference for the speed of the aircraft. When deflecting a control surface, the aircraft rotates in the same direction regardless of the attitude of the aircraft. Whether the aircraft is flying with a zero degree bank angle, or upside down, the required commands of the pilot to deflect control surfaces in order to rotate the aircraft in a desired direction remains the same [32]. Thus in this paper, the pilot's actions are the center of reference and are used to diagnose the performance of the aircraft's control surfaces. An aircraft's attitude is described as upset if any of the following conditions are true [32]:

- Pitch attitude is greater than 25° nose up,
- Pitch attitude is greater than 10° nose down,
- Bank angle is greater than 45°,
- Not within the above range of parameters, but flying at airspeeds

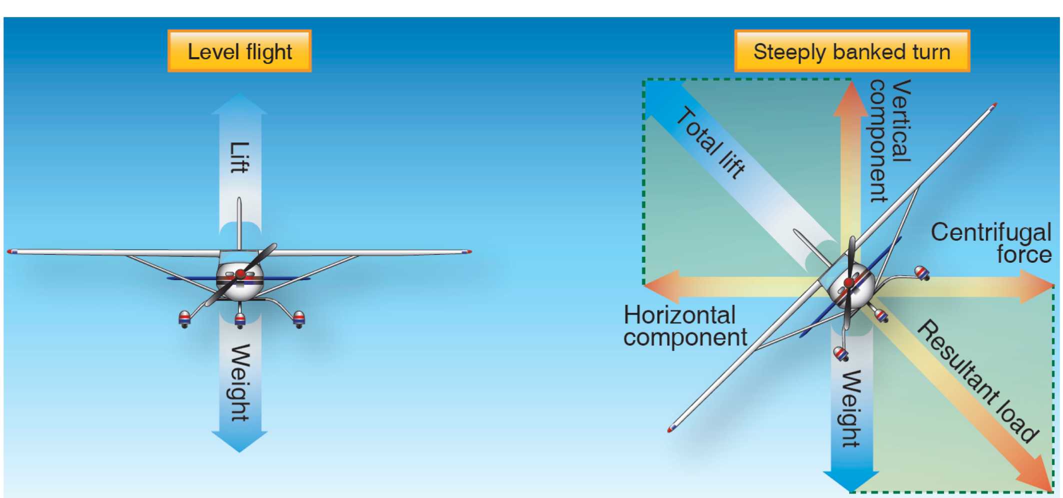

Fig. 3. Level flight and banked turn [1].

inappropriate for the conditions.

2.2. Flight maneuvers

Various versions of a fixed-wing aircraft may have different feel and response characteristics in reference to a pilot's flight control inputs. A modern jet aircraft may be more responsive to the deflection of control surfaces. Some aircraft are more streamlined than others and can maintain a constant airspeed. These aircraft may require less rudder deflection when banking into a turn. Some aircraft have more powerful rudders than others, or have different tail configurations (e.g., T-Tail or V-Tail). Rather than presenting flight maneuvers for a particular model of fixed-wing aircraft, this paper considers flight maneuvers based upon the fundamental principles of flight to provide a pedagogical baseline of flight management that is transferable to most fixed-wing aircraft.

In this paper, it is assumed that the understudy system is a multi-engine aircraft with engines that never fail, and that the engines' thrust is always symmetrical. This allows the aircraft to continue the operation even in case of failures in rotational actuators (ailerons, elevator, and rudder), without having an immediate crash, providing sufficient time to the diagnosis tools in order to diagnose occurred faults. Without loss of generality, our focus in this paper is to diagnose left and right aileron failures based on rolling behaviors. However, it is straightforward to extend the result of this paper to the case that elevator or rudder are faulty as well.

In straight-and-level flight, the heading and altitude of the aircraft are being held constant. This implies a bank attitude of 0°. A properly trimmed aircraft will require the pilot to apply little to no deflection of control surfaces. In order to bank an aircraft to the right, the pilot raises the right aileron and lowers the left aileron. This is commonly referred to as the right aileron. To roll the aircraft to the left, the pilot must raise the left aileron and lower the right aileron. Analogously this is referred to as the left aileron. If the aircraft is rolling in a certain direction (e.g., left) and the pilot desires to roll the aircraft in the opposite direction (e.g., right), the pilot must apply what is referred to as opposite aileron.

When ailerons are deflected to bank an aircraft, the lowered aileron increases drag and lift, thus producing a rising wing and the raised aileron decreases the drag and lift, thus producing a lowering wing. This difference in drag across the wings causes the aircraft to yaw in the direction opposite to the desired direction of turn. This is commonly referred to as the adverse yaw. To counteract this adverse yaw, in addition to deflecting the ailerons, the pilot must simultaneously deflect the rudder in the desired direction of the turn. A right turn requires the coordinated application of the right rudder, and a left turn requires the coordinated application of the left rudder. If the rudder is not applied correctly, then the aircraft will produce what is called a sideslip angle. A sideslip angle is the angle between the aircraft's longitudinal axis and the relative wind. When applied correctly, a deflected rudder will eliminate adverse yaw, producing a zero degree angle of sideslip. This implies that during the turn, the aircraft's longitudinal axis is aligned with the relative wind, meaning that the aircraft is flying straight into the wind during the turn. This is called a coordinated turn. An aircraft under uncoordinated flight may become unstable and enter into a fatal spin [3,32]. The coordinated use of aileron and rudder during a turn is a common procedure on many types of fixed-wing aircraft including modern jet transport airplanes. Some modern jet airplanes have systems that utilize the combination of ailerons and spoilers with yaw dampers and turn coordinators to diminish the effects of adverse yaw felt by the pilot, in turn, eliminating the use of rudder input from the pilot during a normal turn. However, in situations where the yaw damper system is inoperative and the spoilers are unable to reduce all effects of adverse yaw, rudder deflection by the pilot is required for a coordinated turn [32].

A single deflection of the rudder results in sideslip of the aircraft. During sideslip, the wing facing into the relative wind experiences an increase in lift, while the wing away from the relative wind receives a decrease in angle of attack. This produces a differential lift between the wings, resulting in the aircraft rolling in the direction of the deflection of the rudder. This is referred to as a rolling moment due to sideslip. If the rudder solely is deflected to the right, the aircraft will yaw to the right and place the left wing into the relative wind. This results in the aircraft rolling to the right. Another scenario of an aircraft rolling due to the application of the rudder is called rolling moment due to a yaw rate. When the rudder deflection creates a yaw rate on the aircraft, the outboard wing experiences greater airspeed relative to the inboard wing. This produces a rolling moment towards the direction of the inboard wing. Rudder application to the right will cause a yaw rate to the right, causing the left wing (outboard wing) to rise and produce a rolling moment to the right. Analogously, a sole deflection of the rudder to the left will produce a rolling moment to the left [18,22,32,42]. In our DES model of the aircraft, we assume that the rudder is capable of overpowering both deflected ailerons. In other words, full extension of the rudder is capable of providing a rolling moment greater than the rolling moment of fully extended ailerons. An example of this situation was reported by the National Transportation Safety Board (NTSB) during its investigation of the US Air Flight 427 [40]. Maintaining a precise rolling moment about the aircraft via rudder deflection is extremely difficult, and should only be attempted under dire circumstances.

Based upon the presented information, we consider two flight control modes:

1. *Straight-and-level flight*: For straight-and-level flight control, it is assumed that the aircraft is properly trimmed and its wings are leveled with respect to the artificial horizon. To maintain a straight-and-level flight path, a pilot simply applies no pressure to the aileron and rudder control surfaces, placing them in their neutral position.
2. *Bank turn*: Completing a banked turn is a four-step process by the pilot. To initiate a turn, the pilot must first bank the wings into the direction of the turn. Upon reaching the desired angle of the bank, the pilot then releases pressure on all control surfaces (neutral ailerons and neutral rudder). Releasing pressure neutralizes the control surfaces to prevent increasing the bank angle, thus allowing the aircraft to turn at a constant bank angle. When the aircraft reaches the completion of the turn, the pilot must then apply opposite aileron to roll the aircraft in the opposite direction. This action rolls the wings back to level with respect to the artificial horizon. Once the wings become level, the pilot again releases pressure on all control surfaces to maintain a straight-and-level flight. Therefore, a bank turn is a process consisting of four commands:
(1) Roll to desired bank angle
(2) Release pressure on control surfaces to hold bank angle
(3) Roll wings back to level the wings by applying opposite aileron
(4) Release pressure on control surfaces

3. DES modeling of the flight maneuvering mechanism

A DES model of the system utilizes event transitions and states to provide an abstract representation of a real world system's behaviours. Representing a DES system using a finite-state automaton, we can pictorially sketch the system using a directed graph in which the states are shown as the nodes and the transitions are shown as the edges. An event is an input (e.g., an input command or a notable sensor feedback) which results in a distinct change in the system. A system state may represent a particular mode of operation. We model the aircraft maneuvers by an automaton represented by the four-tuple: $G = (X, \Sigma, \delta, x_0)$, where X, Σ, δ, and x_0 include the system's state space, event set, state transition relation, and initial state, respectively. In G, each event $e \in \Sigma$ can cause a transition from one state to another one via the non-deterministic state transition relation δ. The statement $x' \in \delta(x, e)$ says that it is feasible for G to transition from state x to state x' if event $e \in \Sigma$ occurs when G is in state x. The event set Σ is the union of disjoint sets of observable Σ_o and unobservable events Σ_u. Observable events, typically are system signals that are captured or measured by sensors, whereas unobservable events are not captured or measured by the sensors. Unobservable events may be resulted from the lack of a sensor or even sensor damage. In our DES modeling approach, we model fault events as unobservable events $f \in \Sigma_u \subseteq \Sigma$, whose occurrences must be diagnosed from the observation of observable events. A concatenation of events is called a *string of events*, or simply a *string*. The concatenation of two strings s_1 and s_2 is shown by $s_1.s_2$. Extending the transition rule, δ, to the strings, it can be recursively defined as $\delta(x, s.\, e) = \bigcup_{y \in \delta(x,s)} \delta(y, e)$. The set of strings that can be generated by G from the state x is $\mathcal{L}_G(x) = \{s \in \Sigma^* | \delta(x, s) \neq \emptyset\}$. The language of the system, $\mathcal{L}_G$, is the set of all sequences of strings that can be generated by the automaton G from the state x_0, which can be captured by $\mathcal{L}_G(x_0)\{s \in \Sigma^* | \delta(x_0, s) \neq \emptyset\}$. The Kleene closure of an event set Σ is denoted by Σ^*, which consists of all possible strings formed by the concatenation of events $e \in \Sigma$, including the zero-length string ε. Let $s \in \Sigma^*$. A system trace s and its unobservable extensions originating from the state x are represented by $UE(s, x) = \{s.\, t | t \in \Sigma_u^* \text{ and } s.\, t \in \mathcal{L}_G(x)\}$.

Table 1 lists and describes the pilot commands which are observable discrete events for the aircraft. Each event should be read as such: 'Left Aileron Position|Right Aileron Position|Rudder Position'. Each subscript represents the direction of deflection for its corresponding control surface. For the interpretation of the subscripts, 'U' means up-deflection, 'D' means down-deflection, and 'N' means no-deflection (neutral). For example, $A_UA_DR_L$ describes left aileron up, right aileron down, and left rudder. The events 'Bank Left' and 'Bank Right' represent a coordinated left bank and a coordinated right bank, respectively, and events 'Rudder Left' and 'Rudder Right' represent the sole deflection of the rudder.

3.1. Normal flight mode

Fig. 4 shows the DES model for the normal operation of an aircraft, which includes the following flight maneuvers:

- **Straight-and-Level Flight:** Assuming the bank attitude is 0°, the heading and altitude of the aircraft is constant, the DES representation of the straight-and-level flight is shown by a self-loop of $A_NA_NR_N$ on state 1 of Fig. 4.
- **Banked Right Turn:** The right side of Fig. 4 shows the process for a right banked turn maneuver, which includes the following four steps:
  1. $A_DA_UR_R$ for rolling right to reach the desired bank angle,
  2. $A_NA_NR_N$ for releasing pressure on control surfaces to hold the bank angle,
  3. $A_UA_DR_L$ for rolling wings back to level the plane by applying opposite aileron,
  4. $A_NA_NR_N$ for releasing pressure on control surfaces once wings become level.
- **Banked Left Turn:** Similar to the bank right turn, but in the opposite direction, the left side of Fig. 4 shows the process for a left banked turn maneuver.

Table 1
Pilot flight commands (observable events).

Description	Neutral	Bank Right	Bank Left	Rudder Right	Rudder Left
Event	$A_NA_NR_N$	$A_DA_UR_R$	$A_UA_DR_L$	$A_NA_NR_R$	$A_NA_NR_L$
Symbol	α	β	δ	τ	λ

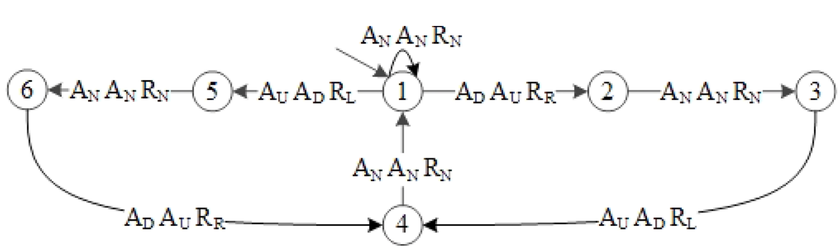

Fig. 4. DES model of normal aircraft flight maneuvers.

3.2. Faulty flight mode

In cases during the flight a fault occurs in flap actuators of control surfaces, the pilot must adapt flight procedures to maintain an attitude that allows for the safe continuation of the flight. In this paper, for the sake of simplicity of the presentation of the model, the rudder is assumed to be non-faulty and we only consider faults that may occur in the left and right ailerons. Therefore, we consider two types of fault: left aileron and right aileron. The faulty events of each fault type are left/right aileron stuck up or stuck down. A fault does not occur when ailerons are in the neutral position. Also, these faults are assumed to be permanent.

We first discuss the faults that may occur during a right bank turn maneuver. When performing a right turn, there are three possible fault occurrence scenarios for the pilot: (i) left aileron stuck down, (ii) right aileron stuck up, and (iii) left and right ailerons simultaneously stuck down and up, respectively.

Case I. Left Aileron Stuck Down Fault (f_{LD}): Upon reaching the desired angle of the bank, the pilot attempts to neutralize all control surfaces. Due to the failure in the left aileron, it is stuck down and the pilot will notice that the plane continues to roll undesirably to the right. At this point in the process of carrying out the banked turn, the pilot is not yet aware that a fault exists, for the excess roll may simply result from a gust of wind, turbulence, ill-trimmed aileron, etc. There is no readout that tells the pilot that the aileron is stuck. To handle the undesirable roll rate the pilot applies the opposite aileron ($A_UA_DR_L$). Since the left aileron is permanently stuck in the down position it will continue to produce lift and roll the plane to the right, thus the pilot's command ($A_UA_DR_L$) applies opposite aileron in order to maintain constant bank when turning. Upon completing the turn, the pilot may apply a greater opposite aileron to roll the airplane left to level the wings. Once the plane is leveled, again the pilot attempts to neutralize control surfaces to resume the straight-and-level flight ($A_NA_NR_N$). Due to the left aileron being stuck down (right aileron is currently neutral), the airplane will unexpectedly roll to the right, and yaw to the left (adverse yaw). With a stuck down left aileron that continuously tries to roll the airplane to the right, and yaw the aircraft's nose to the left; the pilot may again apply opposite aileron ($A_UA_DR_L$) to level the wings. This is now the new flight trim for the faulty plane. To keep the wings as level as possible with respect to the artificial horizon, the left rudder and the right aileron are used to produce a left rolling moment to counteract the right rolling moment constantly being produced by the stuck down left aileron. The pilot may also turn the aircraft to the right using only right rudder ($A_NA_NR_R$). Since the left rudder is stuck down, upon applying only the right rudder, the plane will bank to the right and the rudder will cancel out the adverse yaw created by the downward deflected left aileron. The right aileron will remain neutral as to provide resistance to the rate of roll when banking to the right. This is

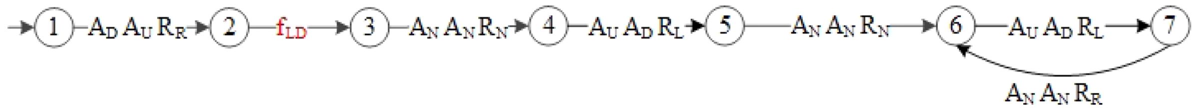

Fig. 5. Flight commands for Left Aileron Stuck Down, f_{LD}.

done for the simple fact that the left aileron may no longer be placed in the neutral position during the banked turn. The pilot does not necessarily attempt to bank the plane left as it may require overexertion of the rudder and left aileron to overcome right rolling moment being produced by the left aileron. For this scenario, a right turn with the right rudder and a neutral right aileron has a very close resemblance to a coordinated right turn. Also, an aircraft enters a spin (which may lead to a downward spiral) in the direction of its yaw. Right rudder counteracts the left yaw being produced by the stuck down left aileron. This sequence of pilot commands after the occurrence of the fault f_{LD} is shown in Fig. 5.

Case II. Right Aileron Stuck Up (f_{RU}): Upon reaching the desired angle of bank, the pilot attempts to neutralize all control surfaces ($A_NA_NA_N$). In case of Right Aileron Stuck Up fault,f_{RU}, like the previous case, the pilot is unaware of a fault occurrence due to the lack of a provided readout in the flight instruments. Therefore, to handle the undesirable roll rate the pilot applies opposite aileron ($A_UA_DR_L$). Since the right aileron is permanently stuck in the up position, it will produce less lift than the left wing, causing the aircraft to continue to roll to the right. The pilot must continuously apply opposite aileron in order to maintain constant bank throughout the turn. Upon completion of the turn, the pilot increases the amount of opposite aileron to roll the aircraft wings. When the aircraft wings are leveled, the pilot neutralizes the flight surfaces to maintain a straight-and-level flight ($A_NA_NR_N$). Due to the right aileron being stuck in the up position, the right wing will unexpectedly drop, and the aircraft will roll right. The pilot must apply left aileron command ($A_UA_DR_L$) to raise the right wing and roll the aircraft level. Since the right wing has less lift, it will continuously become lower during flight. Treating the right wing similar to a dead engine (or spoiler), the pilot attempts to fly the plane such that the right wing is raised at or above the level of the left wing at all times. Thus, the new trim of the aircraft is held by left rudder $A_NA_NR_L$. If needed, the pilot may choose to turn the aircraft to the left by the command $A_UA_DR_L$. This may be accomplished using opposite aileron and full left rudder. Since the right wing is producing more drag and less lift, the aircraft is more stable when the right wing is at or above the level of the left wing. The pilot does not attempt a right turn in this case, as a right bank would entail the lowering of the right wing and would require great control exertion to maintain constant bank angle and raise the right wing upon completion of a right turn. This sequence of actions to recover the aircraft from the failure f_{RU} is shown in Fig. 6.

Case III. Left Aileron Stuck Down and Right Aileron Stuck Up (f_{LDRU}): Upon reaching a desired angle of bank and after attempting to neutralize all control surfaces ($A_NA_NA_N$), in case that both left aileron is stuck down and right aileron is stuck up, the pilot will notice that the plane continues to roll undesirably to the right. Similar to previous cases, the pilot is unaware of a fault occurrence due to lack of provided readout from flight instruments. To manage the undesirable roll rate, the pilot applies opposite aileron ($A_UA_DR_L$) to increase the pressure of left rudder to level the wings. This provides a command readout of ($A_UA_DR_L$). Upon leveling the wings, the pilot attempts to neutralize the control surfaces ($A_NA_NR_N$) for straight-and-level flight. Once the flight command to neutralize the control surfaces is issued, the airplane will roll right due to the left and right ailerons being stuck down and stuck up, respectively. To counter this roll, the pilot must apply opposite aileron ($A_UA_DR_L$) to prevent the aircraft from rolling. Since the ailerons are permanently stuck, this control action will not be adequate to stop the aircraft from continuing to roll right. The pilot will be forced to apply full left rudder ($A_NA_NR_L$) in order to roll the airplane into the left direction. This will be the new flight trim. After rolling the airplane left, the pilot is forced to maintain flight stability with the sole application of the left rudder. The sequence of actions in Case iii is represented in Fig. 7.

Cases I–III cover faulty behaviours of the aircraft during a right bank turn as well as corrective actions of the pilot that position the aircraft into a new trim that allows for sustainable flight. During a bank left turn maneuver, the following fault occurrence scenarios may happen: (i) right aileron stuck down, f_{RD}, (ii) left aileron stuck up, f_{LU}, and (iii) left and right ailerons simultaneously stuck up and stuck down, f_{LURD}. The fault cases for bank left turn are analogous to the fault cases of a bank right turn in that the pilot experiences undesired roll and yaw rates in directions opposite to those experienced during a bank right turn.

3.3. DES model of an aircraft during the banked turn maneuvers

Putting all together, Fig. 8 presents a DES model of the aircraft consisting of normal and faulty aircraft banked turn maneuvers. For the sake of simplicity of presentation, we have used symbols to represent the events. The list of observable events and (unobservable) fault events are shown in Tables 1 and 2.

The left side of the DES model in Fig. 8 shows the normal bank right maneuver and the faulty behaviors when failures f_{LU}, f_{RD}, and f_{LURD} occur. Symmetrically, on the right side, the normal bank left maneuver

Table 2
Fault events (unobservable events).

Right Up	Left Down & Right Up	Left Down	Right Down	Left Up & Right Down	Left Up
f_{RU}	f_{LDRU}	f_{LD}	f_{RD}	f_{LURD}	f_{LU}
f_1	f_2	f_3	f_4	f_5	f_6

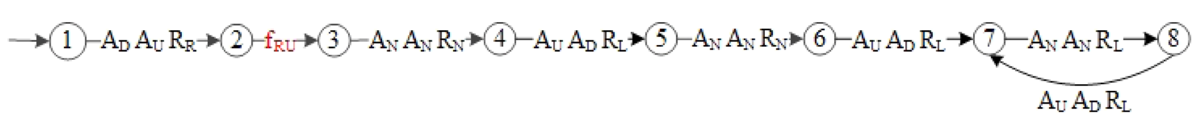

Fig. 6. Flight Commands for Right Aileron Stuck Up, f_{RU}.

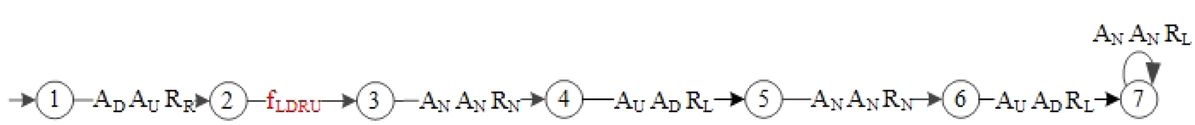

Fig. 7. Flight commands for Left and Right Aileron simultaneously stuck down and stuck up, f_{LDRU}.

is shown as well as the faulty behaviors of the plane when failures f_{LD}, f_{RU}, and f_{LDRU} occur. We use this DES model later for fault detection and analysis in the next section.

4. Asynchronous diagnosis of flight maneuvers

Having the DES model of the flight maneuvers of the aircraft, we can now develop a diagnosis tool to detect, identify, and isolate the faults in the ailerons, based upon the observed maneuvers of the aircraft system. The general structure of a DES diagnosis process is shown in Fig. 1. The Diagnoser G_d uses the original system's observed behaviors as input and provides an estimation of the system's state and condition (faulty or non-faulty) as output. We assume that the number of successive unobservable events is bounded, otherwise, the system may be stuck in a cycle of unobservable events without providing the chance to estimate the status of the system from its observable behaviors.

The condition of the system is faulty if a fault event has occurred in the system. This paper assumes that the fault in the system under diagnosis does not stop its operation and we have this opportunity to estimate the status of the system from its post-fault behaviors. We partition the system's faults, Σ_f, as the union of m different types Σ_{f_1}, Σ_{f_2}, ...,Σ_{f_m}. Correspondingly, for these fault types, we use labels in the form $F = \{f_1, f_2, ..., f_m\}$, where f_i is a label for fault type of Σ_{f_i}. We also use the label N for non-faulty operation condition. The set $L = \{N\} \cup 2^F$ represents all possible condition labels for each system state $x \in X$.

The proposed diagnoser can be represented by a deterministic finite-state DES in the form of the four-tuple $G_d = (Q_d, \Sigma_d, \delta_d, q_0)$, where $Q_d \subseteq 2^{X \times L}$ is the state space, $\Sigma_d = \Sigma_o$ is the event set, δ_d is the state transition rule, and q_0 is the diagnoser's initial state. Each diagnoser state $q \in Q_d$ is a set of ordered pairs providing an estimation of the original system's current state and corresponding condition. During online operation, whenever the system outputs an observable event $e \in \Sigma_o$, the state transition function δ_d, updates the diagnoser's estimation of the system's state and condition.

Assume that the current estimation of the system is $q = \{(x_1, \ell_1), ...,(x_k, \ell_k)\}$, where $x_j \in X$ and $\ell_j \in L$, $j \in \{1, ..., k\}$. Upon the occurrence of a string of one or more events $t \in \Sigma^*$, the diagnoser's state transition function δ_d updates the estimate of the system's state as:

$$\delta_d(q, t) = \bigcup_{\substack{(x,\ell)\in q \\ t\in\Sigma^*}} \{(y, \nabla(\ell, t)) \,|\, y \in (\delta(x, t), \nabla(\ell, t))\} \tag{1}$$

where the system's state transition function δ is used to provide the updated state estimation, and the *Label Updating Function*, $\nabla: L \times \Sigma^* \to L$, is used to update and provide an estimate of the system's condition for the corresponding state estimate produced by δ. Consider the current condition of the system represented by the label $\ell \in L$, observing the new string t, the updated condition of the system will be $\ell' = \nabla(\ell, t)$=:

$$\begin{cases} \{N\}, & \text{if } \ell = \{N\} \text{ and } \forall f \in \Sigma_f,\ f \notin t, \\ \{f_i \in F \,|\, f_i \in \ell \text{ or } \exists f \in \Sigma_{f_i},\ f \in t\}, & \text{Otherwise} \end{cases} \tag{2}$$

Unlike traditional diagnosers that are synchronously initialized with the system under diagnosis, here we use our asynchronous diagnoser developed in [51], which can be activated asynchronously with the system, and hence, it relaxes the synchronous initialization constraint and eases the implementation of the diagnoser. Since the proposed diagnoser is asynchronously activated at any time instance of the system's operation, initially the diagnoser is unknowing of the system's state and condition, and it could be in any of the system's operation modes. Therefore, the diagnoser's initial state will be:

$$q_0 := \{(x, \ell) = (\delta(x_0, t), \nabla(\{N\}, t)) \,|\, \forall\, t \in \mathcal{L}_G(x_0)\} \tag{3}$$

This initial state of diagnoser takes into account all possible system conditions for every reachable system state. Starting from this wide estimation, the asynchronous diagnoser narrows down its estimate of the system's state and condition as it sequentially observes events $e \in \Sigma_o$, evolving by the diagnoser's state transition relation explained in (1).

To implement Algorithm 1, we form a state reachability table (SRT). SRT is a data table which contains all system diagnostic information needed to construct the diagnoser. This diagnostic information can be readily retrieved in each table entry. The SRT is constructed row-by-row. The layout for each row in the SRT is as follows. The first column entry of each row contains a state $x \in X$ and its corresponding diagnostic label $\ell \in L$. All remaining column entries of each row independently corresponds to $e_i \in \Sigma_o$, and provides diagnostic information for all states $x' \in \delta(x, t)$, $t \in UE(e, x)$. The collection of all entries in the first column, forms q_0 for G_d. The rest of the columns are the states that

Initialization:
$q_0 := \{(x_0, N)\}$;
Step 1: Constructing q_0
$q_0 := q_0 \bigcup\{(x, \ell)|x \in \delta(x_0, u),\ u \in \Sigma_u^* \cap \mathcal{L}_G(x_0),\ \ell = \nabla(\{N\}, u)\}$;
repeat
  for $(x, \ell) \in q_0$ and $e \in \Sigma_o$ **do**
    if $\delta(x, e)$ is defined, $\exists t \in UE(e, x)$, and $(\delta(x, t), \nabla(\ell, t)) \notin q_0$ **then**
      $q_0 = q_0 \cup \{(\delta(x, t), \nabla(\ell, t))|t \in UE(e, x)\}$;
    end if
  end for
until There is no new pair (x, l) in q_o.
Step 2: Constructing Q_d
$Q_d := q_0$;
repeat
  for $q \in Q_d$ and $e \in \Sigma_o$ **do**
    if $\delta_d(q, e)$ is defined and $\delta_d(q, e) \notin Q_d$ **then**
      Add $\delta_d(q, e)$ to Q_d;
    end if
  end for
until There is no new state $\delta_d(q, e)$ for all $e \in \Sigma_o$.

Algorithm 1. Constructing an Asynchronous Diagnoser.

Table 3
State Reachability Table for aircraft DES model in Fig. 8.

State	α	β	δ	τ	λ
$1N$	$1N$	$2N$, $41f_1$, $40f_2$, $39f_3$	$18N$, $35f_4$, $34f_5$, $33f_6$	–	–
$2N$	$3N$	–	–	–	–
$41f_1$	$13f_1$	–	–	–	–
$40f_2$	$9f_2$	–	–	–	–
$39f_3$	$5f_3$	–	–	–	–
$18N$	$19N$	–	–	–	–
$35f_4$	$29f_4$	–	–	–	–
$34f_5$	$25f_5$	–	–	–	–
$33f_6$	$20f_6$	–	–	–	–
$3N$	–	–	$4N$, $36f_4$, $37f_5$, $38f_6$	–	–
$13f_1$	–	–	$14f_1$	–	–
$9f_2$	–	–	$10f_2$	–	–
$5f_3$	–	–	$6f_3$	–	–
$19N$	–	$45N$, $44f_1$, $43f_2$, $42f_3$	–	–	–
$29f_4$	–	$30f_4$	–	–	–
$25f_5$	–	$26f_5$	–	–	–
$20f_6$	–	$21f_6$	–	–	–
$4N$	$1N$	–	–	–	–
$36f_4$	$29f_4$	–	–	–	–
$37f_5$	$25f_5$	–	–	–	–
$38f_6$	$20f_6$	–	–	–	–
$14f_1$	$15f_1$	–	–	–	–
$10f_2$	$11f_2$	–	–	–	–
$6f_3$	$7f_3$	–	–	–	–
$45N$	$1N$	–	–	–	–
$44f_1$	$13f_1$	–	–	–	–
$43f_2$	$9f_2$	–	–	–	–
$42f_3$	$5f_3$	–	–	–	–
$30f_4$	$31f_4$	–	–	–	–
$26f_5$	$27f_5$	–	–	–	–
$21f_6$	$22f_6$	–	–	–	–
$15f_1$	–	–	$16f_1$	–	–
$11f_2$	–	–	$12f_2$	–	–
$7f_3$	–	–	$8f_3$	–	–
$31f_4$	–	$32f_4$	–	–	–
$27f_5$	–	–	$28f_5$	–	–
$22f_6$	–	–	$23f_6$	–	–
$16f_1$	–	–	–	–	$17f_1$
$12f_2$	–	–	–	–	$12f_2$
$8f_3$	–	–	–	$7f_3$	–
$32f_4$	–	–	–	–	$31f_4$
$28f_5$	–	–	–	$28f_5$	–
$23f_6$	–	–	–	$24f_6$	–
$17f_1$	–	–	$16f_1$	–	–
$24f_6$	–	$23f_6$	–	–	–

are reachable from q_0, immediately by an observable event $e_i \in \Sigma_o$. The rows of the table also help find the rest of the states of the diagnoser, reachable from already extracted diagnoser's states. In Table 3, we have constructed the SRT for the DES model of the aircraft in Fig. 8, whose first column is q_0, and second, third, fourth, and fifth ones are diagnoser states q_7, q_8, q_6, q_{11}, and q_3, reachable by events α, β, δ, τ, and λ, respectively. From these states, we can build the rest of states. For example, from $q_3 = \{17f_1, 12f_2, 31f_4\}$ with the event δ the diagnoser transits to $\{16f_1\}$ which is called q_4. This is due to the fact that among the states $31f_4$, $12f_2$, and $17f_1$, in rows 35th, 39th, and 44th, only $17f_1$ goes to $16f_1$ by the event δ.

5. Online fault diagnosis

The constructed diagnoser in Fig. 9 provides fault diagnosis by detecting, isolating, and identifying a fault's occurrence without requiring the restarting of the system. When the diagnoser is activated, observing each observable event of the aircraft, the diagnoser proceeds to a new state in the diagnoser based on the transition rules in (1).

Definition 1. Consider $q = \{(x_1, \ell_1), \ldots, (x_M, \ell_M)\} \in Q_d$. Then, q is said to be

- Normal if $\ell_k = \{N\}$ for all $k = 1, \ldots, M$.
- F_i-certain if $f_i \in \ell_k$ for all $k = 1, \ldots, M$.
- F_i-uncertain if $\exists n, m$ such that $f_i \in \ell_n$, but $f_i \notin \ell_m$.

If the system faults that occur pre and/or post diagnoser activation can be definitively diagnosed, then the system is called asynchronously diagnosable.

Lemma 1. *If there is no cycle of F_i-uncertain state in the asynchronous diagnoser, the system under diagnosis is asynchronously diagnosable and the diagnoser will eventually reach an F_i-certain state during the diagnosis process [51].*

As it can be seen in Fig. 9, the constructed diagnoser G_d does not have any cycle of F_i-uncertain states and hence can diagnose any fault of type f_i upon the diagnoser reaches an F_i-certain state. The developed diagnoser can be asynchronously activated, and is able to diagnose faults that occur pre- and post-diagnoser activation.

Next example shows the fault diagnosis for a fault that is occurred before the activation of the diagnoser:

Example 1. *Fault Occurrence Pre-diagnoser activation:* Consider the aircraft DES model in Fig. 8 where $\Sigma = \{\alpha, \beta, \delta, \tau, \lambda, f_1, f_2, f_3, f_4, f_5, f_6\}$, $\Sigma_o = \{\alpha, \beta, \delta, \tau, \lambda\}$, $\Sigma_u = \Sigma_f = \{f_1, f_2, f_3, f_4, f_5, f_6\}$, and $\Sigma_{fi} = \{f_i\}$, $i = 1, \ldots, 6$. Let's assume the aircraft has completed a left turn, and commands all control surfaces to the neutral position (α). Unbeknownst to the aircraft, the right aileron is stuck up and f_1 has occurred. Upon initialization of the diagnoser, the diagnoser is at its initial state q_0 and the aircraft is at state 13. As the aircraft applies commands to level its wings, and place the aircraft in coordinated flight by going through the following sequences:

$$\begin{tikzcd} 13 \arrow[r, "\delta"] & 14 \arrow[r, "\alpha"] & 15 \arrow[r, "\delta"] & 16 \arrow[r, "\lambda"] & 17 \arrow[l, bend left, "\delta"] \end{tikzcd}$$

, which ends with the aircraft arbitrarily cycling states 16 and 17. The diagnoser G_d, shown in Fig. 9, observes the sequence of events $\delta\alpha\delta(\lambda\delta)^*$, and will go through the following sequences:

$$\begin{tikzcd} q_0 \arrow[r, "\delta"] & q_6 \arrow[r, "\alpha"] & q_{18} \arrow[r, "\delta"] & q_{17} \arrow[r, "\lambda"] & q_{15} \arrow[r, "\delta"] & q_4 \arrow[r, "\lambda"] & q_5 \arrow[l, bend left, "\delta"] \end{tikzcd}$$

. Based upon its observations of the aircraft, G_d enters an arbitrarily cycling between states q_4 and q_5. States q_4 and q_5 are F_1-certain, concluding that f_1 has been diagnosed. For this case, upon a finite number of observations of the aircraft by G_d, the occurrence of f_1 is detected, and the state location of the aircraft is estimated to be either state 16 or 17. Indeed, the diagnoser G_d uses the observed sequence of aircraft commands and eventually reaches an F_1-certain state. From this example one can observe how G_d starts off with a wide estimation of the aircraft's state and condition and narrows down its estimation as it gathers information from the observed aircraft system commands to diagnose the fault occurrence. Reaching an F_i-faulty state, the diagnoser determines both which fault has occurred, and which state the system is in. This can be also checked for all other fault events and aircraft state s.

In contrast to the previous example, the following example shows the fault diagnosis for a fault that is occurred after the activation of the diagnoser.

Example 2. *Fault Occurrence Post-diagnoser activation:* Consider the aircraft DES model in Fig. 8 where $\Sigma = \{\alpha, \beta, \delta, \tau, \lambda, f_1, f_2, f_3, f_4, f_5, f_6\}$, $\Sigma_o = \{\alpha, \beta, \delta, \tau, \lambda\}$, $\Sigma_u = \Sigma_f = \{f_1, f_2, f_3, f_4, f_5, f_6\}$, and $\Sigma_{fi} = \{f_i\}$, $i = 1, \ldots, 6$. Let's assume again that the aircraft is performing a bank left turn, and that the

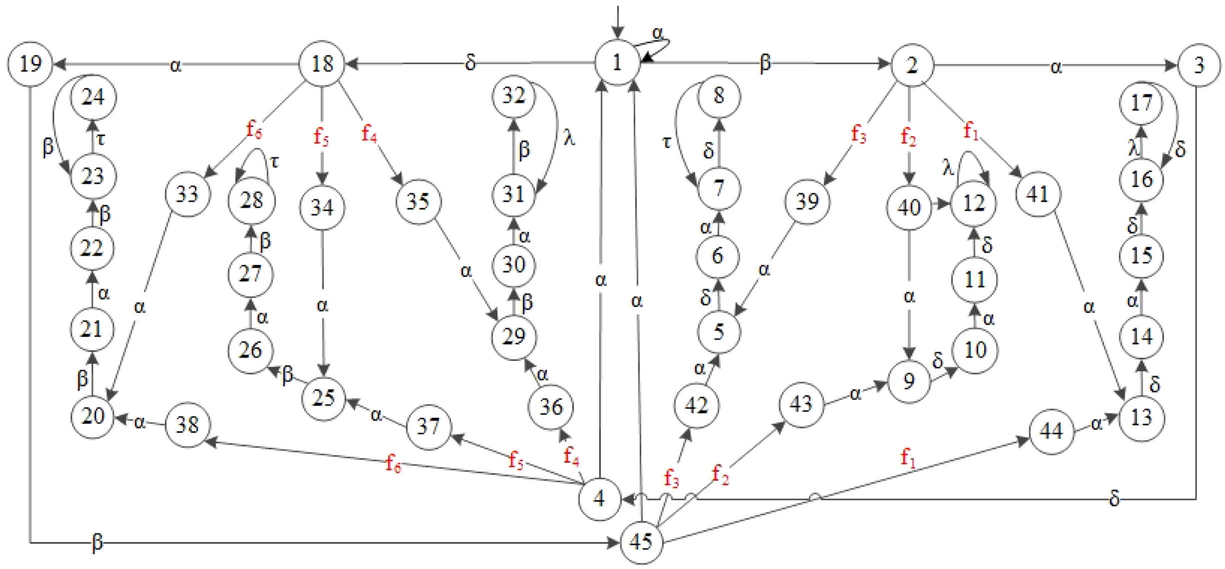

Fig. 8. DES model of a Fixed Wing aircraft including faulty and normal behaviors during the bank turn.

aircraft is completing the turn and applying right aileron command (β) to roll the wings back to level. The aircraft system is located at state 19, and is under normal operation when G_d is activated. The aircraft then goes through the following sequences based on the commands received from the pilot: $19 \xrightarrow{\beta} 45 \xrightarrow{f_3} 42 \xrightarrow{\alpha} 5 \xrightarrow{\delta} 6 \xrightarrow{\alpha} 7 \underset{\tau}{\overset{\delta}{\rightleftarrows}} 8$

. This trajectory ends with the aircraft arbitrarily cycling states 7,8. Upon activation, the diagnoser G_d in Fig. 9, observes the sequence of observable events $\beta\alpha\delta\alpha(\delta\tau)^*$ and starts from the initial state q_0, and will go through the following sequence:

$$q_0 \xrightarrow{\beta} q_8 \xrightarrow{\alpha} q_{20} \xrightarrow{\delta} q_{19} \xrightarrow{\alpha} q_{18} \xrightarrow{\delta} q_{17} \xrightarrow{\tau} q_{14} \underset{\tau}{\overset{\delta}{\rightleftarrows}} q_{13}$$

. The observed behavior of the aircraft results in G_d arbitrarily cycling between states q_{14} and q_{13}. States q_{14} and q_{13} are F_3-certain, concluding that f_3 has occurred. Again, upon a finite number of observations of the aircraft by G_d, the diagnoser was able to detect the occurrence of f_3, and estimate the state location of the aircraft to be either state 7 or 8.

Lemma 1 provides a condition, which if meets, the diagnoser can eventually transit to an F_i-certain state and diagnose the occurred fault. To realize the diagnosis delay (the number of transitions that it take diagnoser to reach an F_i-certain state), one can count the number of events that may occur in the system under diagnosis which leads the diagnoser to transit to an F_i-certain state. For this purpose, we can identify all sequences of F_i-uncertain state and their corresponding transitions in the original system under diagnosis. For example, in the diagnoser G_d in Fig. 9, it can be verified that faults f_1 and f_6 upon their occurrence, require the greatest number of system observations for diagnosis. Both fault events are diagnosed upon the 6th observation following their occurrence in the plant. All other faults may be diagnosed within an equal or fewer number of observations. Let's consider the fault f_6. The following sequence consists of the f_6-faulty sequence in the original system under diagnosis:

$$\xrightarrow{f_6} 33 \xrightarrow{\alpha} 20 \xrightarrow{\beta} 21 \xrightarrow{\alpha} 22 \xrightarrow{\beta} 23 \underset{\beta}{\overset{\tau}{\rightleftarrows}} 24$$

. This corresponds to two sequences in the diagnoser G_d

$$G_d \xrightarrow{\alpha} q_7 \xrightarrow{\beta} q_8 \xrightarrow{\alpha} q_{20} \xrightarrow{\beta} q_{28} \xrightarrow{\tau} q_{27} \xrightarrow{\beta} q_9 \underset{\beta}{\overset{\tau}{\rightleftarrows}} q_{10}$$

or $\xrightarrow{\alpha} q_{29} \xrightarrow{\beta} q_{25} \xrightarrow{\alpha} q_{26} \xrightarrow{\beta} q_{28} \xrightarrow{\tau} q_{27} \xrightarrow{\beta} q_9 \underset{\beta}{\overset{\tau}{\rightleftarrows}} q_{10}$

. Both sequences end with the diagnoser state q_9, which is F_6-certain. Therefore, in both possible observed sequences in the diagoser, the faulty event f_6 is diagnosed once diagnoser G_d reaches q_9 after at most 6 transitions in the system under diagnosis. Note that with the available information (the DES model of the system under diagnosis and the observed observable events), this is the fastest way that one can diagnose a fault in the system, unless there is more information available about the system under diagnosis.

6. Conclusion

This paper developed a DES model for the flight mechanism of a fixed-wing airplane, which is capable of explaining normal and faulty behaviors of the plane, including faults in left/right ailerons (stuck up, stuck down). The derived DES model is then used to develop an asynchronous diagnoser for detecting the occurred faults, the type of the occurred faults, and estimating the state of the system under diagnosis. External observations of the aircraft system and the pilot commands are used to determine whether a fault has occurred in the aforementioned control surfaces. A unique feature of the developed diagnoser is that it can be activated any time before or after occurrences of a fault, and can be restarted any time without requiring the system under diagnosis to be restarted. Two cases of fault occurrences before and after activation of the diagnoser were investigated and it was shown that in both cases the developed diagnoser is able to determine the occurred faults in a finite number of observations. The developed model can be extended to include rudder (stuck left, stuck right) faults as well as engine and elevator failure, when the pilot is planning for straight-level flight, and banking an airplane to the left or right. Future research, therefore, includes the extension of the developed method to other system's faults in other sensors and actuators as well as engines of an aircraft.

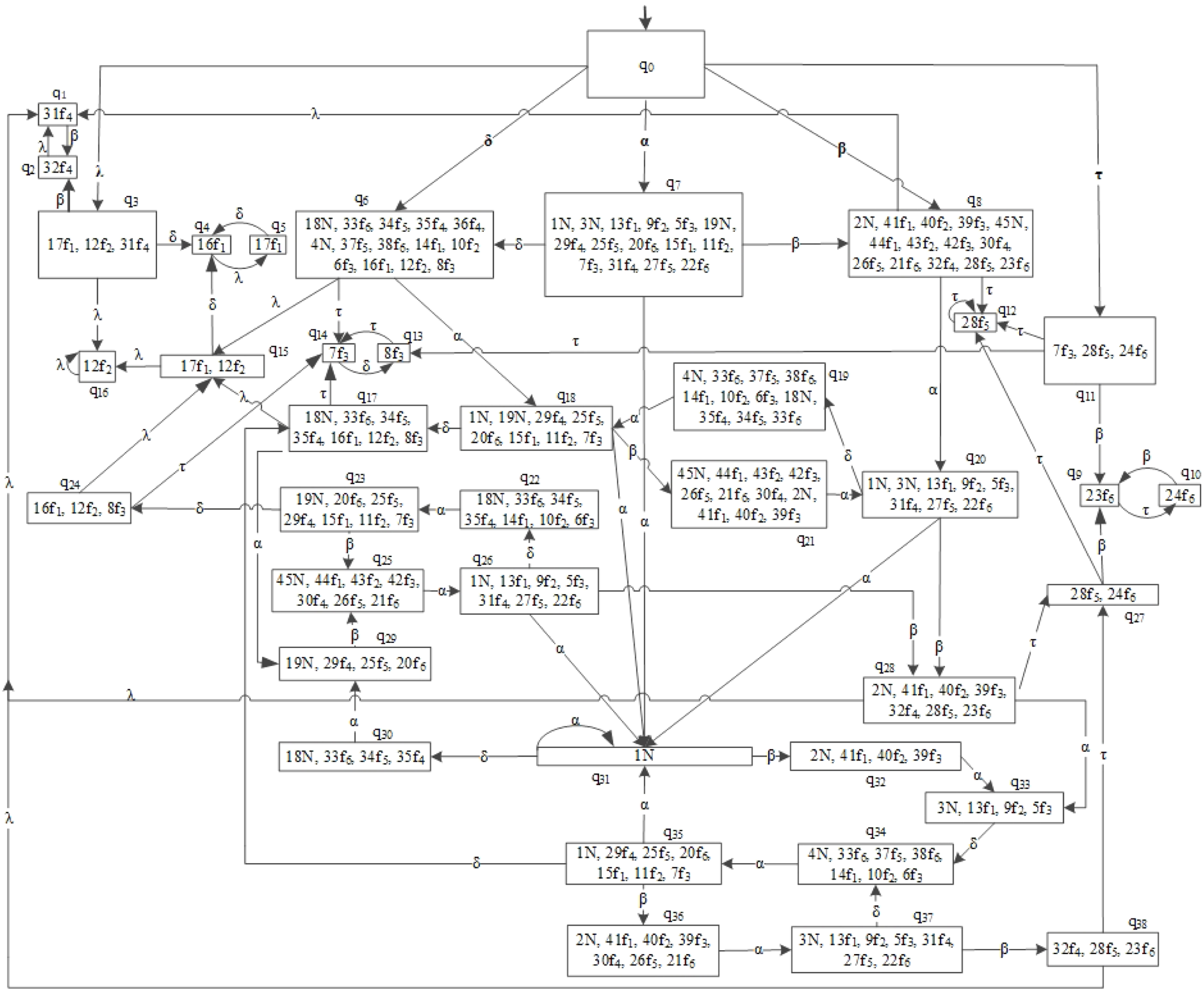

Fig. 9. The diagnosr G_d for a fixed-wing aircraft DES model provided in Fig. 8.

Acknowledgment

We would like to thank Dr. Navid Goudarzi who provided us with technical feedback on the aerodynamic modelling of the planes. Also, we thank the anonymous reviewers for their insightful comments and suggestions, which helped us greatly improve the paper. Further, we thank the support from the National Science Foundation under the award number 1832110 and Air Force Research Laboratory and OSD under agreement number FA8750-15-2-0116. The first author would like to acknowledge the support from Title III HBGI grant from the U.S. Department of Education. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, AFRL, OSD, DoED, or the U.S. Government.

References

[1] Federal aviation administration. https://www.faa.gov.
[2] Federal Aviation Administration. Advanced avionics handbook. US Department of Transportation 2009.
[3] Federal Aviation Administration. Airplane flying handbook. US Department of Transportation 2016.
[4] Federal Aviation Administration. Pilots handbook of aeronautical knowledge. US Department of Transportation 2016.
[5] Amoozgar MH, Chamseddine A, Zhang Y. Experimental test of a two-stage kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. J Intell Robot Syst 2013;70(1–4):107–17.
[6] Andrews JD, Dunnett SJ. Event-tree analysis using binary decision diagrams. IEEE Trans Reliab 2000;49(2):230–8.
[7] IATA. Guidance material and best practices for the implementation of upset prevention and recovery training. Montréal: IATA; 2018.
[8] Avram RC, Zhang X, Campbell J, Muse J. IMU Sensor fault diagnosis and estimation for quadrotor uavs. IFAC-PapersOnLine 2015;48(21):380–5.
[9] Baroni P, Lamperti G, Pogliano P, Zanella M. Diagnosis of a class of distributed discrete-event systems. IEEE Trans Syst Man Cybern Part A: Syst Hum 2000;30(6):731–52.
[10] Beard RW, Kingston D, Quigley M, Snyder D, Christiansen R, Johnson W, et al. Autonomous vehicle technologies for small fixed-wing UAVs. J Aerosp Comput Inf Commun 2005;2(1):92–108.
[11] Bhattacharyya S, Huang Z, Chandra V, Kumar R. A discrete event systems approach to network fault management: detection amp; diagnosis of faults. Proceedings of the American control conference. 6. 2004. p. 5108–5113 vol.6.
[12] Bhattacharyya S, Kumar R, Huang Z. A discrete event systems approach to network fault management: detection and diagnosis of faults. Asian J Control 2011;13(4):471–9.
[13] Blanke M, Hansen S. Towards self-tuning residual generators for UAV control surface fault diagnosis. Proceedings of the conference on control and fault-tolerant systems (SysTol). IEEE; 2013. p. 37–42.
[14] Carvalho LK, Basilio JC, Moreira MV. Robust diagnosis of discrete event systems

against intermittent loss of observations. Automatica 2012;48(9):2068–78.
[15] Cassandras CG, Lafortune S. Introduction to discrete event systems. Springer Science & Business Media; 2009.
[16] Chen J, Patton RJ. Robust model-based fault diagnosis for dynamic systems. 3. Springer Science & Business Media; 2012.
[17] Contant O, Lafortune S, Teneketzis D. Diagnosability of discrete event systems with modular structure. Discret Event Dyn Syst 2006;16(1):9–37.
[18] Cook MV. Flight dynamics principles: a linear systems approach to aircraft stability and control. Butterworth-Heinemann; 2012.
[19] Das SR, Holloway LE. Characterizing a confidence space for discrete event timings for fault monitoring using discrete sensing and actuation signals. IEEE Trans Syst Man Cybern Part A: Syst Hum 2000;30(1):52–66.
[20] Gao Z, Cecati C, Ding SX. A survey of fault diagnosis and fault-tolerant techniquespart I: fault diagnosis with model-based and signal-based approaches. IEEE Trans Ind Electron 2015;62(6):3757–67.
[21] Goupil P. Airbus state of the art and practices on FDI and FTC in flight control system. Control Eng Pract 2011;19(6):524–39.
[22] Gudmundsson S. General aviation aircraft design: applied methods and procedures. Butterworth-Heinemann; 2013.
[23] Hansen S, Blanke M. Diagnosis of airspeed measurement faults for unmanned aerial vehicles. IEEE Trans Aerosp Electron Syst 2014;50(1):224–39.
[24] Instrument Procedures Handbook (IPH) H. FAA-H-8083-16B. Federal Aviation Administration, US Department of Transportation 2017.
[25] Karimi MM, Karimoddini A, White AP, Bates IW. Event-based fault diagnosis for an unknown plant. Proceedings of the IEEE 55th conference on decision and control (CDC). London: Springer-Verlag; 2016. p. 7216–21.
[26] Lafortune S. Diagnosis of discrete event systems. Encyclopedia of systems and control. London: Springer-Verlag; 2015. p. 268–75.
[27] Lee H, Snyder S, Patterson A, Hovakimyan N. Aircraft actuator fault detection and isolation using piecewise constant fault estimation scheme. Proceedings of the AIAA guidance, navigation, and control conference. 2016.
[28] Lee W-S, Grosh DL, Tillman FA, Lie CH. Fault tree analysis, methods, and applications - a review. IEEE Trans Reliab 1985;34(3):194–203.
[29] Lerner U, Parr R, Koller D, Biswas G. Bayesian fault detection and diagnosis in dynamic systems. Proceedings of the AAAI/IAAI. 2000. p. 531–7.
[30] Lima STS, Basilio JC, Lafortune S, Moreira MV. Robust diagnosis of discrete-event systems subject to permanent sensor failures. Proceedings of the 10th IFAC workshop on discrete event systems. 2010.
[31] Lin F. Diagnosability of discrete event systems and its applications. Discret Event Dyn Syst 1994;4(2):197–212.
[32] Manual on Aeroplane Upset Prevention and Recovery Training. Doc 10011. International Civil Aviation Organization 2014.
[33] Pandalai DN, Holloway LE. Template languages for fault monitoring of timed discrete event processes. IEEE Trans Autom Control 2000;45(5):868–82.
[34] Paoli A, Lafortune S. Safe diagnosability for fault-tolerant supervision of discrete-event systems. Automatica 2005;41(8):1335–47.
[35] Philippot A, Sayed-Mouchaweh M, Carré-Ménétrier V. Unconditional decentralized structure for the fault diagnosis of discrete event systems. Dependable control of discrete systems. 1. 2007. p. 67–72.
[36] Qiu W, Kumar R. Decentralized failure diagnosis of discrete event systems. IEEE Trans Syst Man Cybern Part A Syst Hum 2006;36(2):384–95.
[37] Ramadge PJ, Wonham WM. The control of discrete event systems. Proc IEEE 1989;77(1):81–98.
[38] Aircraft Accident Report. NTSB-AAR-92-05 (PB92-910496). National Transportation Safety Board (NTSB) 1992.
[39] Aircraft Accident Report. NTSB-A-94-064. National Transportation Safety Board (NTSB) 1994.
[40] Aircraft Accident Report. NTSB-AAR-99-01 (PB99-910401). National Transportation Safety Board (NTSB) 1994.
[41] Rosa P, Silvestre C. Fault detection and isolation of LPV systems using set-valued observers: an application to a fixed-wing aircraft. Control Eng Pract 2013;21(3):242–52.
[42] Russell J. Performance and stability of aircraft. Butterworth-Heinemann; 1996.
[43] Saied M, Shraim H, Francis C, Fantoni I, Lussier B. Actuator fault diagnosis in an octorotor UAV using sliding modes technique: theory and experimentation. Proceedings of the European control conference (ECC). IEEE; 2015. p. 1639–44.
[44] Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D. Diagnosability of discrete-event systems. IEEE Trans Autom Control 1995;40(9):1555–75.
[45] Sharifi F, Mirzaei M, Gordon BW, Zhang Y. Fault tolerant control of a quadrotor UAV using sliding mode control. Proceedings of the conference on control and fault-tolerant systems (SysTol). IEEE; 2010. p. 239–44.
[46] Su R, Wonham WM. Global and local consistencies in distributed fault diagnosis for discrete-event systems. IEEE Trans Autom Control 2005;50(12):1923–35.
[47] Venkataraman R, Bauer P, Seiler P, Vanek B. Comparison of fault detection and isolation methods for a small unmanned aircraft. Control Eng Pract 2019;84:365–76.
[48] Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: part I: quantitative model-based methods. Comput Chem Eng 2003;27(3):293–311.
[49] Wang Y, Yoo T-S, Lafortune S. Diagnosis of discrete event systems using decentralized architectures. Discret Event Dyn Syst 2007;17(2):233–63.
[50] White A, Karimoddini A. Semi-asynchronous fault diagnosis of discrete event systems. Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC). 2016. p. 003961–6.
[51] White A, Karimoddini A. Asynchronous fault diagnosis of discrete event systems. Proceedings of the American control conference (ACC). 2017. p. 3224–9.
[52] White A, Karimoddini A, Su R. Fault diagnosis of discrete event systems under unknown initial conditions. IEEE Trans Autom Control 2019. https://doi.org/10.1109/tac.2019.2912712.
[53] Zaytoon J, Lafortune S. Overview of fault diagnosis methods for discrete event systems. Annu Rev Control 2013;37(2):308–20.
[54] Zhang X, Parisini T, Polycarpou MM. Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. IEEE Trans Autom Control 2004;49(8):1259–74.
[55] Zhang X, Tang L, Decastro J. Robust fault diagnosis of aircraft engines: a nonlinear adaptive estimation-based approach. IEEE Trans Control Syst Technol 2013;21(3):861–8.
[56] Zheng Y, Fang H, Wang HO. Takagi–Sugeno fuzzy-model-based fault detection for networked control systems with Markov delays. IEEE Trans Syst Man Cybern Part B Cybern 2006;36(4):924–9.