ON THE USE OF BEHAVIORAL MODELS FOR THE INTEGRATED PERFORMANCE AND RELIABILITY EVALUATION OF FAULT-TOLERANT AVIONICS SYSTEMS

Alejandro D. Domínguez-García, John G. Kassakian, Joel E. Schindall
Laboratory for Electromagnetic and Electronic Systems, Massachusetts Institute of Technology, Cambridge, MA

Jeffrey J. Zinchuk
Systems Engineering and Evaluation, The Charles Stark Draper Laboratory, Inc., Cambridge, MA

Abstract—In this paper, we propose an integrated methodology for the reliability and performance analysis of fault-tolerant systems. This methodology uses a behavioral model of the system dynamics, similar to the ones used by control engineers when designing the control system, but incorporates additional artifacts to model the failure behavior of the system components. These artifacts include component failure modes (and associated failure rates) and how those failure modes affect the dynamic behavior of the component. The methodology bases the system evaluation on the analysis of the dynamics of the different configurations the system can reach after component failures occur. For each of the possible system configurations, a performance evaluation of its dynamic behavior is carried out to check whether its properties, e.g., accuracy, overshoot, or settling time, which are called performance metrics, meet system requirements. After all system configurations have been evaluated, the values of the performance metrics for each configuration and the probabilities of going from the nominal configuration (no component failures) to any other configuration are merged into a set of probabilistic measures of performance. To illustrate the methodology, and to introduce a tool that we developed in MATLAB/SIMULINK® that supports this methodology, we present a case-study of a lateral-directional flight control system for a fighter aircraft.

I. INTRODUCTION

Fault-tolerant systems that are required for aircraft avionics need a thorough understanding of their behavior, and require powerful evaluation tools during their development phase, in order to fully understand and quantify potential component or subsystem failures and design proper failure recovery mechanisms.

There are in the literature [1], [2] well-developed techniques in the field of control engineering to design effective and robust control systems and to analyze their performance in the presence of disturbances and model uncertainties. Applications of these techniques to the design of aircraft avionics can be found in the literature [3]. The application of these techniques relies on having a behavioral model of the system dynamics, which is usually a set of differential equations, but can be a more general mathematical model.

The analysis of system behavior in the presence of components or subsystem failures and therefore system reliability, availability, and safety, is usually carried out using a qualitative description of the system’s functionality. It describes how components and subsystems are interconnected to fulfill the functions they are designed for, and how component failures can propagate to other components through their interconnections and affect the system functionality [4], [5], [6]. For conventional systems, this approach is valid, as it is possible to evaluate how a system can fail to perform the function for which it was designed without paying attention to the dynamics. However, this is not the case for large complex systems or embedded software-intensive systems, which are characteristic of fault-tolerant systems [7]. For these kinds of systems, it is very difficult, if not impossible, to understand how the system behaves in the presence of hardware failures or software malfunctions without using a behavioral model of the system to quantify its performance under failure conditions.

In this paper, we propose a methodology that uses a behavioral model of the system dynamics, similar to the ones used by control engineers when designing the control system, but with additional features to model component failure behaviors. These features include component failure modes (and associated failure rates) and how these failure modes affect the dynamic behavior of the component. Starting from
the nominal configuration, the system will evolve to different system configurations, depending on which components failed, how these components failed, and the order in which the components failed (when more than one failed). For each of the possible system configurations, a performance evaluation of the system dynamic behavior is carried out to check whether some of its properties, called performance metrics, meet system operational requirements. After all system configurations have been evaluated, the values of the performance metrics for each configuration and the probabilities of going form the nominal configuration to any other configuration are merged into a set of probabilistic measures of performance. The proposed methodology allows one to assess not only system reliability and safety, but other important dynamic performance metrics that might be relevant in the design of a fault-tolerant system. The behavioral model not only includes the system architecture (the avionics) under evaluation, but it also includes a model of the physical system to be controlled (the aircraft). The performance evaluation is based on how the physical system behaves when failures in components within the control system occur.

The idea of using a dynamic model of the system under control was proposed first in the nuclear engineering field by Amendola in 1981 to study the likelihood of accident sequences in a nuclear reactor, referred to as the Event Sequences and Consequences Spectrum (ESCS) [8]. In 1987, Aldemir [9] formalized some of the ideas introduced in [8], and introduced the idea of using a Markov model to compute the likelihood of different accident sequences in the reactor. In 1992 Devooght and Smidts laid down a rigorous mathematical formulation of the problem [10]. This methodology is commonly referred in the nuclear engineering field as Dynamic Probabilistic Risk Assessment (DPRA).

The methodology proposed in this paper is similar to DPRA in the sense that it makes use of a dynamic (behavioral) system model, with additional information to model component behavior. It also makes use of a Markov chain to model the stochastic transitions between system configurations that take place when components fail. However, there are certain aspects of the kind of problems that DPRA tries to solve that makes the mathematical formulation of DPRA much more complex than the formulation of our methodology. In aerospace problems the dynamic time constants are on the order of seconds. Therefore, when a component failure occurs, the system either recovers within a short transient period, in the order of seconds, reaching a new stable steady-state (associated with a new system configuration), or it quickly becomes unstable. In this scenario, the likelihood of two components failing within a time on the order of magnitude of the system dynamic time constants is negligible relative to the likelihood of just one component failing. Therefore, the stochastic transitions among configurations can be modeled as a process independent of the system dynamics, and the resulting model is formulated in terms of just the probability of the system configurations at a given time. This is not the case in nuclear power plants, where the decoupling of the stochastic transitions between configurations and system state dynamic variables cannot be done [11]. This is due to the fact that in nuclear power plants the time constants of the transitions from one configuration to another are large enough that the likelihood of another failure taking place before the system reaches a new steady-state is relevant. Therefore, in DPRA the stochastic model is formulated in terms of the probability of the system dynamic variables in a given configuration at a given time.

Section II of this paper presents and explains in detail the elements of the methodology and lays out its mathematical foundations. Section III presents the main features of a MATLAB/SIMULINK® tool that supports the proposed methodology. In Section IV a lateral-directional flight control system for a fighter aircraft is presented as a case-study. Concluding remarks and future work are presented in Section V. Additionally, all the dynamic models and parameters of the components for the analyzed case-study presented in Section IV are collected in several appendices at the end of the paper. The notation used is listed at the end of the paper as well.

II. METHODOLOGY

The methodology uses a model of the system dynamics plus additional features to model component failure behavior. These features include component failure modes (and associated failure rates) and how these failure modes affect the dynamic behavior of the component. Depending on which components failed, how this components failed, and the order in which components failed (when more than one failed), the system will evolve from the nominal configuration (no failures) to other configurations with different
A. System Configurations

The system is composed of different interconnected components. When there are no failures in the system, the system is said to be in its nominal configuration, and its dynamics can be described by

\[
\begin{align*}
\dot{x}(t) &= f_i(x(t), u(t)) \\
y(t) &= g_i(x(t), u(t))
\end{align*}
\]

where the subindex \(i\) equals 1 for the nominal configuration. The system can evolve from the nominal configuration (no component failures) to other dynamic configurations depending on the status of the components within the system. These statuses are dictated by the component failure modes. Each system configuration in the presence of component failures, \(i = 2 \ldots N\), can also be described by (1).

The transitions between configurations occur stochastically and are triggered by random failures within the system components. Each component can have different failure modes, and each failure mode has an associated failure rate. The system will be in any of the configurations, \(i\), within system global evaluation time \(T\) (system life-span, preventive maintenance period or mission time), with certain probability \(p_i(T)\) given that the system was in the configuration \(i = 1\) at \(T = 0\). We assume that the system evolves between configurations in a Markovian fashion, i.e., the system configurations at future times will only depend on the configuration at the present time. Therefore the Chapman-Kolmogorov equations can be used to compute the system configuration probabilities as

\[
\frac{dp_i(t)}{dt} = -\lambda_k(t)p_k(t) + \sum_{i \neq k} \lambda_{ik}(t)p_i(t)
\]

where \(\lambda_k(t)\) results from adding all the individual failure rates associated with the different component failure modes that can trigger a system transition out of configuration \(k\). \(\lambda_{ik}(t)\) is the failure rate associated with the failure mode that triggers the transition from system configuration \(i\) to system configuration \(k\). As mentioned in the introduction, and shown here, the formulation of the stochastic model of the transitions between configurations is independent of the system dynamic variables, and, therefore, it is possible to compute the configuration probabilities \(p_k(t)\) independently of the system dynamics state-variables \(x\).

DPRA tries to compute the probability of the system dynamic variables in a given configuration at a given time \(p_k(x, t)\), and therefore the formulation of the Chapman-Kolmogorov equations becomes much more complicated [10].

B. Performance Metrics

Performance metrics \(Z_1, Z_2, \ldots, Z_m\) are measurable properties of the system that will quantify how well it performs the functions for which it was designed. Performance metrics can be system functionality related, e.g., in a tracking-system, relevant performance metrics are the tracking error, the overshoot or the settling time. However, non-functional metrics can be considered as well, e.g., in the tracking system, the power consumption may be important.

For each performance metric \(Z_i\) with \(i = 1 \ldots m\), a set of requirements must be defined. These requirements can be given in the form of bounds as

\[
L_1 < Z_1 < U_1 \\
L_2 < Z_2 < U_2 \\
\ldots \\
L_m < Z_m < U_m.
\]

C. Performance Evaluation and Probabilistic Measures of Performance

For each system configuration represented by (1), the behavior of the system is analyzed in order to quantify its performance metrics. For a given system configuration, \(\{f_i, g_i\}\), if any of the performance metrics \(Z_1, Z_2, \ldots, Z_m\), do not meet the specified requirements given in the form of bounds by (3), then that configuration is regarded as a system failure.

The performance metrics \(Z_1, Z_2, \ldots, Z_m\) are useful in determining whether each possible system configuration is declared as failed or non-failed, but it is necessary to define another set of measures to quantify the system as a whole, i.e., aggregated measures for all the possible system configurations. This set of measures, called probabilistic measures of performance, will be
helpful in identifying the weakest points in the design and therefore improving the system design as a whole. System reliability is an example of an aggregated measure of performance when the performance metric considers only whether the system is functional or not.

The two elements that compose the probabilistic measures of performance are

1) the system configuration probabilities \( p_i(T) \) given by (2), and

2) a reward model, given by

\[
\begin{align*}
    r_1 &= h_1(Z_1) \\
    r_2 &= h_2(Z_2) \\
    &\ldots \\
    r_m &= h_m(Z_m)
\end{align*}
\]  

where \( h_1(\cdot), h_2(\cdot), \ldots, h_m(\cdot) \) are real functions.

The probabilistic measures of performance are computed as the expectation of the reward model by

\[
E(R_1) = \sum_k h_1(z^k_1) \hat{\mu}_k(T) \\
E(R_2) = \sum_k h_2(z^k_2) \hat{\mu}_k(T) \\
&\ldots \\
E(R_m) = \sum_k h_m(z^k_m) \hat{\mu}_k(T)
\]

where \( z^k_1, z^k_2, \ldots, z^k_m \) are the values of the performance metrics for the non-failed system configurations \( k \in FC \), with \( FC \) being the set of system non-failed configurations, and \( \hat{\mu}_k(T) \) the conditional probability that the system has reached the non-failed configuration \( k \) given that the system is in any of its possible non-failed configurations \( FC \):

\[
\hat{\mu}_k(T) = \frac{p_k(T)}{\sum_{i \in FC} p_i(T)}.
\]

III. MATLAB/SIMULINK EVALUATION TOOL

In order to illustrate the proposed methodology, we developed a MATLAB/SIMULINK® based tool. This tool implements all of the elements of the methodology described in Section II, enabling the automatic evaluation of any system. The SIMULINK® environment enables the construction of the system behavioral model and the definition of performance metrics as well as requirements for them. A collection of scripts coded in MATLAB® carries out the performance evaluation and the calculation of probabilistic measures of performance.

A. System Behavioral Model Definition

The system behavioral model is defined in the SIMULINK® environment. In section II, it was mentioned that the system configurations dynamic models, \( (1) \), is one of the elements of the methodology. Theoretically it is necessary to have all the system configurations beforehand to carry out the analysis of a system. In practice, the system nominal configuration \( (i = 1 \text{ in equation } 1, \text{ no failures}) \) is modeled in SIMULINK® by interconnecting the nominal models of each component, and afterwards each component model is augmented with an artifact to model the component failure behavior. The realization of the remaining system configurations is carried out through simulation by injecting different sequences of component failure modes \( (i = 2 \ldots N \text{ in } (1)) \).

Figure 1. SIMULINK® Control Surface Position Sensor Model.

To illustrate how a system model is defined, we will show how to define one of the components within a model. The complete system model will result after connecting all the component models. Figure 1 shows a SIMULINK® model for one of the control surface position sensors of the case-study presented in Section IV. The mathematical model from which this example was developed is shown in Appendix VI. On the upper part of Figure 1, the elements labeled Transfer Fcn, Delay, and Resolution are common SIMULINK® blocks that allow the sensor functional properties to be modeled.

The block on the upper-right of Figure 1, named Failures, is just an artifact to model the component failure behavior. The content of the Failures block is displayed on the right bottom of Figure 1. The
failure behavior of the sensor depends on the Switch control input $U_f$, which is controlled by the simulation environment created in MATLAB®. For $U_f = 0$, the Switch block input 0 is passed to the output, meaning the sensor is failure-free. When the Switch control input $U_f$ is set to 1, 2, or 3, the inputs 1, 2, 3 of the Switch block are passed to the output, modeling, respectively, Omission, Gain-Change and Bias failure modes.

The bottom-left of Figure 1 shows the GUI through which the user interfaces with the Failures block, allowing the definition of failure rates for each failure mode—this case, Omission, Gain-change, and Bias. It is possible to define constant failure rates and time-dependent failure rates if component wear-out mechanisms are to be modeled. The Failures block GUI allows other parameters of the component failure model to be defined—in this case, the Gain-change factor and the Bias factor. The component can only go from the non-failed status to any of the failed statuses defined by the failure modes, but once it is in a failed status, it remains there—it cannot fail in a different mode. However, it is possible to refine the tool and allow transitions, not only from the non-failed status to any failure status, but between any failed status.

![Output of evaluated system](image)

Figure 2. SIMULINK® performance metric definition.

B. Performance Metrics Definition

The performance metrics, as well as their requirements, are defined in SIMULINK®. Figure 2 illustrates how to define a performance metric. The content of the Performance metric block is displayed on the right bottom of Figure 2, which defines the performance metric as the difference between the output of the system under evaluation and a reference. The bottom-left of the figure shows the GUI through which the user interfaces with the Performance metric block, allowing the definition of the upper and lower bounds.

C. Performance Evaluation and Probabilistic Measures of Performance

For each possible system configuration, the system is simulated in order to check whether or not the performance requirements are met. At the same time the performance of the system is evaluated for each sequence of failures, the state-transition matrix associated with (2) is built.

The steps followed by the tool can be summarized as:

1) Each possible single failure is injected in the SIMULINK® system behavioral model. Therefore $n_1$ new system configurations are realized, where $n_1$ is the sum of failure modes of all the system components.
2) For each system configuration realized in step 1 a new entry is added to the state-transition matrix associated with (2).
3) The behavior of the configurations realized in step 1 is simulated for a period of time, and based on the results of the simulation, the system is declared to be in a failed configuration or in a non-failed configuration.
4) From the component failures resulting in non-failed configurations in step 3, and the remaining non-failed components, sequences of two-failed elements are generated and injected in the SIMULINK® system behavioral model. There results $n_2$ new system configurations, where $n_2$ is the number of sequences with two different failed components (in a particular failure mode).
5) For each new system configuration realized in step 4 a new entry is added to the state-transition matrix associated with (2).
6) The behavior of the configurations realized in step 5 is simulated for a period of time, and based on the results of the simulation, the system is declared in a failed configuration or in a non-failed configuration.
7) From the sequences of failures resulting in non-failed configurations in step 6 and the remaining non-failed components, new sequences of three failed elements are generated and injected in the SIMULINK® system behavioral model.
This process lasts until all the sequences of component failures result in a failed system configuration or the analysis is truncated when the sequences of failed components have a predetermined size, i.e., number of failed elements within the sequence.

For a given sequence of failures $s_k$ of size $k$ ($k$ components are failed) the system behavior is simulated for a period of time $t_c$ (configuration evaluation time) and for a given set of system control inputs $u(t)$. At the beginning of the simulation period, the system state variables $x(t)$ are set to the final values of the simulation period for the sequence of failures $s_{k-1}$ that gave birth to the sequence $s_k$ by introducing an additional yet-to-fail $k^{th}$ component.

After the simulation is running, the failure of component $k$ is introduced at $t_f$ (failure injection time) with $t_f < t_c$. The values of $t_f$ and $t_c$ are chosen such that the system reaches a new steady-state in $t_c - t_f$ if the new configuration is stable, or the system behavior rapidly diverts from the expected nominal behavior if the configuration is unstable. If the system configuration is unstable, or it is stable but the performance metrics do not meet requirements, the configuration is declared as failed. If the configuration is stable and meets performance requirements, then it is declared as a non-failed configuration.

Once all the system configurations are evaluated, the probabilities $p_i(T)$ of reaching, at time $T$, each system configuration, $i$, from the nominal configuration at time $T = 0$ can be computed. A reward model, (4), is built with the values of the performance metrics for each system configuration. Finally, probabilistic measures of performance are computed using (5). The outputs for each system configuration are:

- The sequence of component failures leading to that system configuration.
- Probability of being in that system configuration at the global evaluation time $T$.
- The performance metrics values in that system configuration.
- A tag to indicate whether the configuration is failed or not.

Additionally, System reliability and unreliability values are collected, as well as the probabilistic measures of performance associated with each performance metric. The tool also allows each particular system configuration to simulated individually.

### IV. Case-Study

In this section, we present a case-study of a fault-tolerant architecture for a fighter aircraft lateral-directional flight control system [3]. For clarity, the system architecture under study is described in detail here, but the behavioral model equations for each component within the system are described in several appendices at the end of the paper.

The authors acknowledge the fact that the redundancy and the failure detection, isolation, and reconfiguration (FDIR) elements introduced in the system under study are, by no means, complete. However, one of the objects of the methodology presented in this paper is to uncover weak design points, e.g., not enough component redundancy, or insufficient FDIR to achieve fault-tolerance. Therefore, we use this architecture to show how the methodology presented in this paper can help guiding the redundancy allocation and the FDIR design.

#### A. System Behavioral Model Definition

The behavioral model of the architecture was defined using SIMULINK®. In this section, we present a case-study of a fault-tolerant architecture for a fighter aircraft lateral-directional flight control system [3]. For clarity, the system architecture under study is described in detail here, but the behavioral model equations for each component within the system are described in several appendices at the end of the paper.

The proposed architecture is composed of two redundant primary flight computers (PFC) (Appendix I) that receive information about aircraft attitude from three redundant inertial measurement units (IMU) (Appendix II) and from triple redundant control surface (rudder, right aileron and left aileron) position sensors (Appendix VI) and reconfiguration signals from the other PFC. Both PFCs have a voting algorithm (Appendix I) implemented to compute the actual aircraft attitude from the redundant IMU measurements. Both PFCs have also implemented control laws (Appendix I) that, based on the IMU measurements and the pilot inputs, compute the appropriate commands for the control surface actuation subsystems (Appendix III). Each control surface (Appendix V) is actuated by two redundant actuation subsystems commanded independently from each PFC, which are linked to the corresponding control surface by a mechanical combiner (Appendix IV). Each actuation subsystem is composed of a current-controlled electric motor. Therefore, there are three actuation subsystems per PFC, commanding both left and right ailerons, and the rudder.

As shown above, every component, except the control surfaces, is duplicated or triplicated in order to achieve fault-tolerance. However, though redundancy is an important element of any fault-tolerant system,
there is another equally important element—failure detection, isolation and reconfiguration (FDIR). In this case-study, some FDIR elements have been introduced to illustrate how the system will recover from some failures, and will not recover from others despite the presence of redundancy. The PFCs have a failure detection circuit able to detect a failure-by-omission in the computer. When this happens, the failure-detection circuit shuts down the main computer processor and sends a reconfiguration signal to the other computer. When this happens, the failure-detection circuit able to detect a failure-by-omission in the presence of redundancy. The PFCs have a failure detection, isolation and reconfiguration (FDIR). In this case-study, some FDIR elements have been introduced to illustrate how the system will recover from some failures, and will not recover from others despite the presence of redundancy. The PFCs have a failure detection circuit able to detect a failure-by-omission in the computer. When this happens, the failure-detection circuit shuts down the main computer processor and sends a reconfiguration signal to the other computer.

Table 1 collects information corresponding to the failure modes of the different components. The actual failure modes are part of the component behavioral models described in Appendices I-VI. The possible failure modes of each component are listed in column 2 of Table 1, while column 3 is an explanation of the effect of each failure mode on the component behavior. \( U_f \) in column 4 is the variable that assigns the corresponding failure mode to the component behavioral model equations (see Appendices I-VI). The last column of Table 1 collects the failure rates \( \lambda \) associated with each failure mode. These failure rates will allow the construction of the state-transition matrix associated with (2).

To complete the system behavioral model, a linear lateral-directional aircraft dynamic model [12], [13] interacting with the avionics architecture model described above is included. The state variables of this model are the sideslip \( \beta \), the body axis yaw rate \( r_b \), and the body axis roll angle \( \phi \). The control surface commands are both left and right aileron angles \( \delta_a^l \) and \( \delta_a^r \), and the rudder angle \( \delta_r \). The complete state-space representation of the aircraft lateral-directional dynamics is shown in Appendix VII.

### B. Performance Metrics Definition

The system nominal configuration (no failures) will be used to define the performance metrics. In the nominal configuration, the aircraft state variables are denoted by the super index 0, i.e., the sideslip is denoted by \( \beta^0 \), the body axis roll angle by \( p_b^0 \), the body axis yaw rate by \( r_b^0 \), and the body axis roll angle by \( \phi^0 \). The performance metrics are defined as maximum difference (positive or negative) between the actual state aircraft state variables and the nominal configuration ones and are denoted by

\[
Z_\beta = \beta(t) - \beta^0(t) \\
Z_{p_b} = p_b(t) - p_b^0(t) \\
Z_{r_b} = r_b(t) - r_b^0(t) \\
Z_{\phi} = \phi(t) - \phi^0(t). \tag{7}
\]

The requirements of these performance metrics are defined as

\[
L_\beta < Z_\beta < U_\beta \\
L_{p_b} < Z_{p_b} < U_{p_b} \\
L_{r_b} < Z_{r_b} < U_{r_b} \\
L_{\phi} < Z_{\phi} < U_{\phi} \tag{8}
\]

where \( U_\beta = 0.15\text{rad}, L_\beta = -0.15\text{rad}, U_{p_b} = 0.15\text{rad}, L_{p_b} = -0.15\text{rad/s}, U_{r_b} = 0.15\text{rad}, L_{r_b} = -0.15\text{rad/s}, \) and \( U_{\phi} = 0.15\text{rad}, L_{\phi} = -0.15\text{rad}.\n
### TABLE 1

<table>
<thead>
<tr>
<th>Component</th>
<th>Failure modes</th>
<th>Description</th>
<th>( U_f )</th>
<th>( \lambda/(\text{h}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Right Computer (i=1,2)</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>2 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Random</td>
<td>Random output between -5 and 5</td>
<td>2</td>
<td>10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Stuck</td>
<td>Output at last correct value</td>
<td>3</td>
<td>10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Delayed</td>
<td>Output delayed 0.2 s</td>
<td>4</td>
<td>10^{-1}</td>
</tr>
<tr>
<td>Left Aileron_actuation subsystem (i=1,2)</td>
<td>Omission</td>
<td>Output set to zero regardless the input</td>
<td>1</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>Left Aileron_actuation subsystem (i=1,2)</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>Right Aileron_actuation subsystem (i=1,2)</td>
<td>Omission</td>
<td>Output at last correct value</td>
<td>2</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>Left Aileron, Right Aileron, Rudder</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>Internal measurement unit (i=1,2,3)</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>4 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Gain change</td>
<td>Output scaled by a factor of 1.5</td>
<td>2</td>
<td>3 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Biased</td>
<td>Output biased by a factor of 0.3</td>
<td>3</td>
<td>3 \cdot 10^{-1}</td>
</tr>
<tr>
<td>Left Aileron angle sensor (i=1,2,3)</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>4 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Gain change</td>
<td>Output scaled by a factor of 1.5</td>
<td>2</td>
<td>3 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Biased</td>
<td>Output biased by a factor of 0.3</td>
<td>3</td>
<td>3 \cdot 10^{-1}</td>
</tr>
<tr>
<td>Right Aileron angle sensor (i=1,2,3)</td>
<td>Omission</td>
<td>Output set to zero</td>
<td>1</td>
<td>4 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Gain change</td>
<td>Output scaled by a factor of 1.5</td>
<td>2</td>
<td>3 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>Biased</td>
<td>Output biased by a factor of 0.3</td>
<td>3</td>
<td>3 \cdot 10^{-1}</td>
</tr>
</tbody>
</table>
In Figure 3. The maintenance period of the aircraft response for the nominal configurations are displayed. The system control inputs considered for evaluation must rapidly divert from the expected nominal behavior, the system must reach a new steady-state if appropriate. Therefore, in the remaining constants dictate that a configuration evaluation time $t_c = 20s$, and a failure injection time $t_f = 4s$ are appropriate. Therefore, in the remaining 16s of simulation, the system must reach a new steady-state if the new configuration is stable, or the system behavior must rapidly divert from the expected nominal behavior. The system control inputs considered for evaluating each configuration, as well as the aircraft dynamic response for the nominal configurations are displayed in Figure 3. The maintenance period of the aircraft was considered as the system global evaluation time $T$, taking a value of 500h.

Table 2 shows the probability of system failure (unreliability) at the end of the maintenance period for different levels of truncation. It can be seen that it is enough to evaluate up to system configurations with three components failed. Truncating after three component failure events yields a system unreliability of $3.20 \times 10^{-4}$ and a truncation error of $2.39 \times 10^{-7}$. There is a trade-off between achieving a higher accuracy in estimating reliability and computational time for performing the evaluation. By truncating the evaluation at the second level, only 67 system configurations are evaluated, and the evaluation takes less than 3 minutes. If the truncation is done at the third level, 3675 possible configurations are analyzed and the evaluation takes 3 hours and 15 minutes. The computation was carried out on a machine with a 2.1 GHz Pentium® M processor, and 1.5Gb of RAM.

Table 3 displays the values of the probabilistic measures of performance corresponding to different truncation levels. In this case, the probabilistic measures of performance are the expected values of the system performance metrics given the system is in a non-failed configuration. Therefore, these measures give an idea of the system average deviation from nominal behavior. It is important to note that the nominal configuration does not affect these results since $Z_\beta, Z_p, Z_r$, and $Z_\phi$ are equal to zero in the nominal configuration. This is relevant, otherwise the result would be heavily biased since the probability of being in the nominal configuration is $p_1(500) = 0.988$, whereas the probability of being in any configuration with one component failed is on the order of $10^{-4}$. For this same reason, the results $E[Z_\beta], E[Z_p], E[Z_r],$, and $E[Z_\phi]$ do not significantly change by truncating the evaluation at the second level of failure instead of at the third level. Therefore, it is necessary to investigate whether it is more appropriate to compute these expectations at each level of failure.

**TABLE 2**

<table>
<thead>
<tr>
<th>Trunc. level</th>
<th>Unreliability Lower bound</th>
<th>Unreliability Upper bound</th>
<th>Truncation Error</th>
<th># of configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$3.13 \times 10^{-4}$</td>
<td>$3.85 \times 10^{-4}$</td>
<td>$7.17 \times 10^{-5}$</td>
<td>67</td>
</tr>
<tr>
<td>3</td>
<td>$3.20 \times 10^{-4}$</td>
<td>$3.20 \times 10^{-4}$</td>
<td>$2.39 \times 10^{-7}$</td>
<td>3675</td>
</tr>
</tbody>
</table>

Figure 4 shows the system performance metrics $Z_\beta, Z_p, Z_r$, and $Z_\phi$ for a single failure in the left aileron, in which it fails by getting stuck at the position it was in when the failure occurred. Although the system performance is degraded, the performance...
metrics remain within their requirements (0.15rad or -0.15rad). In this case, there are not many things that can be done to improve the system performance since the aileron is non-redundant, and when it fails, it affects the aircraft aerodynamics. It might be possible to modify the control laws to account for a failure of this type.

Figure 5 shows the performance metrics for another failure of the left aileron. The failure mode is such that the aileron is now commanded by the aircraft dynamics, i.e., the aileron trails. In this case, the failure is catastrophic. It can be seen that 4s after the failure occurs, $Z_r$ rapidly increases. This means that the aircraft is rolling without any control. It can be seen that $Z_\phi$ increases as well, which is a natural result since $\phi$ is the aircraft rolling angle.

Figure 6 corresponds to a failure by omission of one of the left aileron actuation subsystems. This results in degraded system performance. Nevertheless the configuration is stable and meets the performance requirements. Therefore this configuration is considered as non-failed. It would be possible to have perfect system performance when failures occur within the actuation subsystems by implementing FDIR mechanisms to account for these failures.

With FDIR implemented when any of the primary flight computers fails by output omission, the performance metrics remain smaller than $3 \cdot 10^{-3}$ in absolute value. Therefore the system behavior is almost unaffected. This is expected since computer output omission failures can be detected and isolated with the self-detection circuit built in each computer, and the control laws in the remaining computer reconfigured to account for this type of failure.

Figure 7 shows the results for another type of failure within the computer. In this case, it is a failure that causes the computer to output a random command for the control surface actuation subsystems. Unlike the failure-by-output-omission, this failure is not
survivable, the system becomes unstable resulting in a failed configuration. A way to remove this failed configuration could be by improving the self-detection circuit of each computer to allow the detection and isolation of this type of failure within the computer.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we demonstrate that model-based performance evaluation of fault-tolerant systems is a powerful methodology for the design phase of fault-tolerant systems. By including other important system performance metrics and relying on behavioral and objective models of the system, the system performance and reliability is evaluated in a more thorough fashion than more traditional techniques based on system functional models. We prove the feasibility of the methodology by showing a tool developed using MATLAB/SIMULINK® that performs the system evaluation automatically.

There are still several issues that need to be investigated. As mentioned before, it is not clear if the probabilistic measures of performance must be calculated by aggregating configurations with the same number of components failed or as an aggregate of all non-failed configurations across all the levels of failure. So far, the methodology only addresses how to evaluate a given system architecture, but it does not address in a structured way how to improve the architecture. It is true that the methodology uncovers weak points in the design and might help improve certain parts to eliminate single points of failure or improve the performance of certain non-failed configurations. However, there is nothing in the methodology that points out the system unreliability drivers, i.e., which component failures are driving the system reliability or the performance of the non-failed configurations. Therefore it is necessary to investigate how to incorporate new elements to the methodology to help improve the design in an structured way.

ACKNOWLEDGMENT

Some of the ideas presented in Section III were developed in collaboration with Benjamin Sewell and Dr. Peter Miller from Ricardo UK. The authors would like to thank Dr. Philip S. Babcock at the Charles Stark Draper Laboratory for his invaluable ideas and fruitful discussions, and for his constant support of this research. Piero Miotto at the Charles Stark Draper Laboratory provided the IMU model and fruitful discussions. Thanks also to Thomas T. Myers at Systems Technology, Inc. for providing technical information regarding the linear lateral-directional aircraft dynamics model.

REFERENCES

2) Yaw Control:

\[ R_y(s) = K_{y1} \delta_y(s) + \frac{z_{y1} + z_{y2} \tau}{s(z_{y1} + s_{y2})} (K_{y2} R_\delta(s) + K_{y3} P_b(s)) \]

\[ \delta_{y}(r) = \begin{cases} \sigma_y(r), & \text{for } r = 0 \\ 2\sigma_y(r), & \text{for } r = 1 \\ \sigma_y(r), & \text{for } U_f = 0 \\ 0, & \text{for } U_f = 1 \\ \text{rand}(l, u), & \text{for } U_f = 3 \\ \sigma_y(\delta_{y}(r)), & \text{for } U_f = 4 \end{cases} \]

where \( K_{y1} = 0.001, K_{y2} = -0.7816s, K_{y3} = 0.172s, z_{y1} = -0.00125s^{-2}, z_{y2} = -0.001875s^{-1}, p_y = 1.5s^{-1}, P_y = 0.45A, I_y = 6A/s, D_y = 0.01As, K_y = -1.33, \) were taken from [3]; and \( l = -5A, u = 5A, \tau_1 = 0.2s, \) and \( \tau \) is the time at which the computer gets stuck.

APPENDIX II

INERTIAL MEASUREMENT UNIT

\[ x_{imu}(t + \Delta t) = x_{imu}(t) + \text{randn}(0, \sigma_{ss} \sqrt{1 - e^{-2\Delta t/\tau}}) \]

\[ y_{imu}(t + \Delta t) = y_{imu}(t) + \text{randn}(0, \sigma_{ss}) \]

\[ \hat{y}_{imu} = \begin{cases} y_{imu}, & \text{for } U_f = 0 \\ 0, & \text{for } U_f = 1 \\ G y_{imu}, & \text{for } U_f = 2 \\ y_{imu} + B, & \text{for } U_f = 3 \end{cases} \]

where \( \sigma_{ss} = 3.15 \cdot 10^{-6}, \Delta t = 0.01s, \tau = 100s, \mu_b = 4.85 \cdot 10^{-6} \text{randn}(0, 1), \sigma_b = 2.09 \cdot 10^{-5}, K_1 = 10^{-4}, K_2 = 9.7 \cdot 10^{-5}, G = 1.5, \) and \( B = 0.3\text{deg}. \)

APPENDIX III

ACTUATION SUBSYSTEM

3) Ailerons:

\[ G(s) = k_1 \frac{(s + \frac{1}{\tau})(s + \frac{1}{\tau G_s})}{(s + \frac{1}{\tau})(s + \frac{1}{\tau})(s + \frac{1}{\tau G_s})} \]

\[ I_{u_1(2)}(s) = k_{2} G(s) \frac{I_{a_1(2)}(s)}{s} \]

\[ I_{a_1(2)}(t) = \begin{cases} \tau_{a_1(2)}(t), & \text{for } U_f = 0 \\ 0, & \text{for } U_f = 1 \\ \tau_{a_1(2)}(t), & \text{for } U_f = 2 \end{cases} \]
where $k_1 = 5000s^2$, $k_2 = 50$, $\tau_c = 10s$, $\tau_m = 10s$, $\tau_1 = 4s$, $\tau_2 = 10s$, $\tau_3 = 100s$, and $\tau$ is the time at which the actuation subsystem gets stuck.

4) Rudder:

$$G(s) = k_1 \frac{s + \frac{1}{\tau_c}}{(s + \frac{1}{\tau_m})(s + \frac{1}{\tau_1})(s + \frac{1}{\tau_2})(s + \frac{1}{\tau_3})}$$

$$T_{T_1} (s) = k_2 \frac{G(s)}{s + \frac{1}{\tau_c}} \delta_r(s)$$

$$\dot{T}_{T_1} (t) = \begin{cases} T_{T_1} (\tau), & \text{for } U_f = 0 \\ T_{T_1} (t), & \text{for } U_f = 1 \\ T_{T_1} (\tau), & \text{for } U_f = 2 \end{cases}$$

where $k_1 = 5000s^2$, $k_2 = 50$, $\tau_c = 10s$, $\tau_m = 10s$, $\tau_1 = 4s$, $\tau_2 = 10s$, $\tau_3 = 100s$, and $\tau$ is the time at which the actuation subsystem gets stuck.

**APPENDIX IV**

**MECHANICAL COMBINER**

A. Ailerons

$$T_{a}^{\dot{L}, r} = G_a (\dot{T}_{a_1}^{(r)} + \dot{T}_{a_2}^{(r)})$$

where $G_a = 0.5$.

B. Rudder

$$T_r = G_r (\dot{T}_{r_1} + \dot{T}_{r_2})$$

where $G_r = 0.5$.

**APPENDIX V**

**CONTROL SURFACES**

A. Ailerons

$$\begin{align*}
\dot{x}_{a}^{(r)} &= -\frac{1}{\tau_a} x_{a}^{(r)} + \frac{1}{\tau_a} T_{a}^{\dot{L}, r} \\
\dot{\delta}_{a}^{(r)} &= \sigma_{a} \gamma_{a} (x_{a}) \\
\dot{\delta}_{a}^{(r)} &= \begin{cases} \delta_{a}^{(r)}, & \text{for } U_f = 0 \\ \delta_{a}^{(r)}, & \text{for } U_f = 1 \\ \delta_{a}^{(r)}, & \text{for } U_f = 2 \end{cases}
\end{align*}$$

where $\tau_a = 0.04s$ and $\tau_l = 0.11s$, $\alpha_0 = 0.216rad$, were taken from [3]. $\tau$ is the time at which the aileron gets stuck.

B. Rudder

$$\begin{align*}
\dot{x}_{r} &= -\frac{1}{\tau_r} x_{r} + \frac{1}{\tau_r} T_r \\
\dot{\delta}_{r} &= \sigma_{r} \gamma_{r} (x_r) \\
\dot{\delta}_{r} &= \begin{cases} \delta_{r}, & \text{for } U_f = 0 \\ \delta_{r}, & \text{for } U_f = 1 \\ \Psi, & \text{for } U_f = 2 \end{cases}
\end{align*}$$

where $\tau_r = 0.05s$ and $\tau_l = 0.1s$ were taken from [3]. $\tau$ is the time at which the rudder gets stuck.

**APPENDIX VI**

**CONTROL SURFACES POSITION SENSOR**

A. Ailerons

$$\begin{align*}
\dot{x}_{a}^{(r)} &= -\frac{1}{\tau_a} x_{a}^{(r)} + \frac{1}{\tau_a} \delta_{a}^{(r)} \\
\dot{y}_{a}^{(r)} &= \sigma_{a} (x_{a}) \frac{1}{\tau_a}
\end{align*}$$

$$\begin{align*}
\dot{y}_{a}^{(r)} &= \begin{cases} y_{a}, & \text{for } U_f = 0 \\ 0, & \text{for } U_f = 1 \\ G y_{a}, & \text{for } U_f = 2 \\ y_{a} + B, & \text{for } U_f = 3 \end{cases}
\end{align*}$$

where $\tau_a = 0.01s$, $\tau_l = 0.001s$, $R = 0.001$, $G = 1.5$, and $B = 0.3deg$.

B. Rudder

$$\begin{align*}
\dot{x}_{r} &= -\frac{1}{\tau_r} x_{r} + \frac{1}{\tau_r} \delta_{r} \\
\dot{y}_{r} &= \sigma_{r} (x_{r}) \frac{1}{\tau_r}
\end{align*}$$

$$\begin{align*}
\dot{y}_{r} &= \begin{cases} y_{r}, & \text{for } U_f = 0 \\ 0, & \text{for } U_f = 1 \\ G y_{r}, & \text{for } U_f = 2 \\ y_{r} + B, & \text{for } U_f = 3 \end{cases}
\end{align*}$$

where $\tau_r = 0.01s$, $\tau_l = 0.001s$, $R = 0.001$, $G = 1.5$, and $B = 0.3deg$.

**APPENDIX VII**

**LINEAR LATERAL-DIRECTIONAL AIRCRAFT DYNAMICS**

$$\begin{align*}
\begin{bmatrix}
\dot{\beta} \\
\dot{p}_b \\
\dot{r}_b \\
\dot{\phi}
\end{bmatrix}
&= 
\begin{bmatrix}
Y_b & \sin \alpha_0 & -\cos \alpha_0 & \frac{w_0 \cos \alpha_0}{\alpha_0} \\
L_\beta & L_\rho & L_\gamma & 0 \\
N_\beta & N_\rho & N_\gamma & 0 \\
0 & 1 & \tan \alpha_0 & 0
\end{bmatrix}
\begin{bmatrix}
\beta \\
p_b \\
r_b \\
\phi
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & Y_{\delta_r} & 0 & L_{\delta_r} \\
N_{\delta_r} & N_{\gamma} & N_{\delta_r} & 0 & L_{\delta_r} \\
L_{\beta} & L_{\rho} & L_{\gamma} & 0 & \delta_{\phi}
\end{bmatrix}
\begin{bmatrix}
\dot{\delta}_{r} \\
\dot{\delta}_{r} \\
\dot{\delta}_{r} \\
\dot{\delta}_{r}
\end{bmatrix}
\end{align*}$$
where $Y_0 = -50.69 m/s^2$, $V = 178 m/s$, $a_0 = 0.216 rad/s$,

\[
g = 9.81 m/s^2, \quad L_r = -32.38 s^{-2}, \quad L_p = -0.374 s^{-1}
\]

$N_f = 2.40 s^{-1}$, $N_\beta = 1.06 s^{-2}$, $N_p = -0.9406 s^{-1}$,

$N_{\delta f} = -0.0809 s^{-1}$, $Y_{\delta f} = 0.0719 s^{-1}$, $N_{\delta p} = -3.175 s^{-2}$,

$N_{\delta f} = 3.175 s^{-2}$, $N_{\delta f} = 6.66 s^{-2}$, $L_{\delta f} = -0.855 s^{-2}$,

$L_{\delta r} = 0.855 s^{-2}$, and $L_{\delta r} = -1.18 s^{-2}$, were taken from

[14].

**APPENDIX VIII**

**NOTATION**

$B$ : Bias factor for the IMUs and the control surface position sensors

$b_i(\cdot)$ : Reward model function for performance metric $i$

$f_i(\cdot)$ : State evolution function for system configuration $i$

$G$ : Gain change factor for the IMUs and the control surface position sensors

$G_x$ : Left and right mechanical combiner gain

$g$ : Acceleration of gravity

$g_{\delta f}(\cdot)$ : Instantaneous output function for system configuration $i$

$L_q$ : Dimensional variation of rolling moment about $X_q$ with $\beta$

$L_p$ : Dimensional variation of rolling moment about $X_p$ with $p$

$L_{\delta q}$ : Dimensional variation of rolling moment about $X_q$ with $\delta q$

$L_{\delta p}$ : Dimensional variation of rolling moment about $X_p$ with $\delta p$

$m$ : Number of system performance metrics

$N$ : Number of system configurations

$N_{\delta f}$ : Dimensional variation of yawing moment about $Z_{\delta f}$ with $\beta$

$N_{\delta p}$ : Dimensional variation of yawing moment about $Z_{\delta p}$ with $p$

$N_{\delta \beta}$ : Dimensional variation of yawing moment about $Z_{\delta \beta}$ with $\delta \beta$

$N_{\delta f}$ : Dimensional variation of yawing moment about $Z_{\delta f}$ with $\delta f$

$N_{\delta p}$ : Dimensional variation of yawing moment about $Z_{\delta p}$ with $\delta p$

$L_{\delta f}$ : Dimensional variation of yawing moment about $Z_{\delta f}$ with $\delta f$

$\eta_i$ : Number of system performance metrics

$p_{\delta f}, p_{\delta p}(\cdot)$ : Roll rate

$p_i(T)$ : Probability that the system configuration at time $T$ is $i$ given that at $T = 0$ is 1

$\tau$ : Primary Flight Computers reconfiguration signals

$P_{\delta f}, P_{\delta p}(\cdot)$ : Yaw rate

$\tau_i$ : Reward model associated with performance metric $j$

$R$ : Control surface position sensors resolution

$s$ : Laplace transform variable

$t$ : Time

$t_0$ : Configuration evaluation time

$t_f$ : Failure injection time

$T$ : System global evaluation time

$T_{\alpha f}(r), T_{\alpha p}(r)$ : Left (right) aileron torque command

$T_{\alpha f}(r), T_{\alpha p}(r)$ : Left (right) aileron action subsystem torque output

$T_{\alpha r}(\cdot)$ : Rudder torque command

$T_{\alpha r}(\cdot)$ : Rudder action subsystem torque output

$u_1, u_2, u_3$ : Voter inputs

$u_i$ : Voter output

$u(\cdot)$ : System input

$U_{\alpha f}$ : Component failure model switch control input

$V$ : Forward velocity

$x(\cdot)$ : System state variables

$x_{\delta f}, x_{\delta p}$ : Left (right) aileron state variable

$x_{\delta \beta}$ : Inertial measurement unit state variables

$x_{\alpha f}$ : Rudder state variable

$x_{\alpha p}$ : Rudder position sensor state variable

$X_{\alpha f}, Y_{\alpha f}, Z_{\alpha f}$ : Aircraft stability axis

$Y_{\alpha f}$ : System output

$y(\cdot)$ : Inertial measurement output

$y_{\delta f}$ : Left (right) aileron position sensor output

$\gamma_{\delta f}$ : Rudder position sensor output

$Y_{\beta}$ : Dimensional variation of $Y_{\beta}$ force with $\beta$

$Y_{\delta f}$ : Dimensional variation of $Y_{\delta f}$ force with $\delta f$

$z^T(\cdot)$ : Value of performance metric $j$ for system configuration $i$

$\alpha$ : Pitch angle

$\beta$ : Sideslip angle

$\phi$ : Roll angle

$\phi_i$ : Roll command

$\sigma_{\phi i}(\cdot)$ : $\tau$-shift operator

$\Psi$ : Heading angle

$\Gamma$ : Failure time

$\tau_r$ : Inverse of aileron bandwidth

$\tau_r$ : Inverse of rudder bandwidth

$\tau_r$ : Inverse of left, right, and rudder position sensor bandwidth

$\varepsilon_{\beta}$ : Ceiling function

$\varepsilon_{\beta}(l, u)$ : Uniformly distributed random generator between $l$ and $u$

$\varepsilon_{\beta}(\mu, \sigma)$ : Gaussian distributed random generator with mean $\mu$ and standard deviation $\sigma$

$k_1, k_2, r_1, r_2, \tau_1, \tau_2$ : Left, right, and rudder actuation subsystem parameters

$K_{\alpha f}, K_{\alpha p}, K_{\delta f}$ : IMU parameters

$K_{\alpha f}, K_{\alpha p}, K_{\delta f}$ : Roll control law parameters

$K_{\alpha f}, K_{\alpha p}, K_{\delta f}$ : Yaw control law parameters

25th Digital Avionics Systems Conference October 15, 2006
Figure 8. Lateral-Directional Flight Control System Fault-Tolerant Architecture.