Automated Contingency Management for Propulsion Systems

Abhinav Saxena, *Member, IEEE*, Marcos E. Orchard, *Member, IEEE*, Bin Zhang, *Member, IEEE*, George Vachtsevanos, *Senior Member, IEEE*, Liang Tang, *Member, IEEE*, Youngjin Lee, and Yorai Wardi, *Member, IEEE*.

***Abstract*—Increasing demand for improved reliability and survivability of mission-critical systems is driving the development of health monitoring and Automated Contingency Management (ACM) systems. An ACM system is expected to adapt autonomously to fault conditions with the goal of still achieving mission objectives by allowing some degradation in system performance within permissible limits. ACM performance depends on supporting technologies like sensors and anomaly detection, diagnostic/prognostic and reasoning algorithms. This paper presents the development of a generic prototype test bench software framework for developing and validating ACM systems for advanced propulsion systems called the Propulsion ACM (PACM) Test Bench. The architecture has been implemented for a Monopropellant Propulsion System (MPS) to demonstrate the validity of the approach. A Simulink model of the MPS has been developed along with a fault injection module. It has been shown that the ACM system is capable of mitigating the failures by searching for an optimal strategy. Furthermore, few relevant experiments have been presented to show proof of concepts.**

I. Introduction

Growing demand for improving the reliability and survivability of safety-critical aerospace systems has led to many health management (HM) and fault-tolerant control approaches. Such systems are capable of detecting the occurrence of faults while still retaining acceptable performance in the presence of faults. In recent years, numerous propulsion health monitoring technologies have been developed by NASA/DoD to aid in the detection and classification of insipient propulsion failures for various military and space propulsion applications [1]-[4]. These technologies have focused on the detection and diagnosis of developing propulsion and instrumentation faults.

The Automated Contingency Management (ACM) system provides a framework to accommodate these technologies and leads to the design of high confidence propulsion systems with robust fault accommodation and adaptive engine operation reconfiguration necessary for the next generation aero propulsion systems. The proposed ACM technology performs a multi-objective constrained optimization to accommodate impending failure conditions, and provides such potential benefits as [5]:

- Reduced design safety margin generally equating to improved performance
- Higher effective reliability to accomplish mission objectives
- Reduced human burden due to increased autonomy
- Technically accurate contingencies (e.g. engine will recover from surge event; do not shutdown)
- Ability to optimize maintenance intervals for specific components of propulsion and prioritization of tasks to be performed during the planned maintenance interval.

This paper presents an approach for the design of a generic test bench for fault insertion, simulation and ACM algorithm evaluation for propulsion systems. A Matlab/Simulink platform has been used to demonstrate proof-of-concept of the architecture for a monopropellant propulsion system. The paper is organized as follows. The ACM philosophy and a generic approach to design an ACM system are introduced in Section II. Section III presents a proof-of-concept case for a ACM for Monopropellant Propulsion System (MPS). Demonstration scenarios and simulation results are provided in Section IV followed by validation of the proposed approach in Section V. The paper concludes with remarks on future work in Section VI.

Manuscript received October 12, 2001. This work was supported in part by NASA, Ames Research Center. The work reported in this paper was conducted under the SBIR Phase II program, Automated Contingency Management for Advanced Propulsion Systems, with Impact Technologies (www.impact-tek.com) as prime contractors.

A. Saxena is a PhD candidate in the school of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332 USA (Phone : 404-457-2756; fax: 404-894-4130; e-mail: asaxena@ece.gatech.edu).

M. E. Orchard, is a PhD student in the school of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: morchard@ece.gatech.edu).

B. Zhang, is a postdoctoral fellow in the school of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: bin.zhang@gatech.edu).

G. Vachtsevanos, is a professor in school of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: gjv@ece.gatech.edu).

L. Tang is with Impact Technologies LLC, 200 Canal View Blvd. Rochester, NY 14623 USA (e-mail: liang.tang@impact-tek.com).

Y.J. Lee, is an assistant professor in the Department of Electrical Measurement and Control at Korea Aviation Polytechnic College (e-mail: airlee011@hotmail.com).

Y. Wardi, is a professor in school of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: yorai.wardi@ece.gatech.edu).

II. Automated Contingency Management

Background

Engineered systems are subject to failures which may result in potential hazards that must be addressed in a timely manner. Reliability and availability are two key issues to assure dependability of a system. A system is dependable when it is trustworthy enough that reliance can be placed on the service that it delivers [6]. For a system to be dependable, it must be available (e.g., ready for use when is needed), reliable (e.g., able to provide continuity of service while in use), safe (e.g., does not have a catastrophic consequence on the environment), and secure (e.g., able to preserve confidentiality) [7].

Although these system attributes can be considered in isolation, in fact they are interdependent [8]. For instance, a system that is not reliable is also not available (at least when it is not operating correctly). Achieving the goal of dependability requires efforts in all phases of a system's development. Steps must be taken at design time, implementation time, and execution time, as well as during maintenance and enhancement. There are mainly four approaches that are taken based on what phase the system is currently in.

Fault Avoidance: during design phase through validation and verification methodologies so that most known faults are taken care of from the very beginning.

Fault Removal: after the design phase verification is carried out to remove the faults still remaining in the system.

Fault Tolerance: employed during the operational phase, and hence should be able to tackle the problems real-time. This requires the system to be able to diagnose, isolate and identify the faults at the earliest.

Fault Evasion: steps taken in advance to avoid a predicted fault before it actually happens. This requires advanced prognostic capabilities that use observed behavior to predict likely scenarios for near future.

Automated Contingency Management (ACM) mostly falls under *Fault Tolerance* with a slight overlap with *Fault Evasion.* The Following section describes in detail how ACM can be defined followed by a methodology that must be used in designing such system so that it has the attributes of all four approaches as discussed above.

ACM Requirements Definition

Although performance measures for any type of system may vary from one application domain to the next, the system must meet certain general requirements when designed and implemented for critical military or industrial processes. A candidate list of such requirements may include some of the following examples [5]:

- The system must ensure safety and reduced O&S costs over the life of critical system/processes.
- It must be designed as an open system architecture that maximizes ease of subsystem and component changes, upgrades and replacement while minimizing system/process interface changes.
- System Reliability, Availability, Maintainability and Durability (RAM-D) requirements must be met.
- Scalability requirements
- Cost requirements
- User requirements (display, GUI, etc.)
- Compatibility requirements with existing sensors, components, devices, etc.

Other general requirements may be considered specific to the system at hand, its operating environment, etc.

As depicted in Fig. 1, any ACM system should aim for *Mission Success* while satisfying *Safety* requirements. These concepts can further be defined in terms of more concrete specifications that will be used in setting up the problem. Furthermore, this figure shows the approaches that ACM might take to mitigate the effect of a fault. First, it should try to achieve reconfiguration either by changing the mission objectives or by reconfiguring the control modules. If in some situation neither of these is feasible, a more conservative approach of switching to Fail-Safe mode can be taken to avoid any loss [9]-[10].

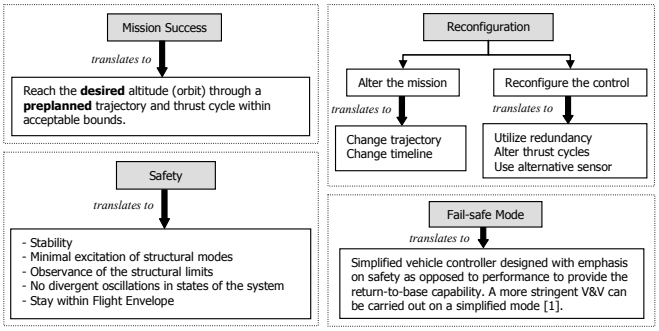

Fig. 1 Objectives and characteristics of an ACM system

The concept of reconfiguration can be viewed as an optimization of available resources given the constraints of mission objectives and safety. We detail a basic methodology that must be followed in order to develop such systems.

We formulate the ACM algorithm as a constrained optimization problem as stated below:

"*Given the current state of the system, and subject to predefined system constraints, find the optimal action series that will bring the system to the desired state with a minimal cost and a highest probability of success."*

In order to formulate this optimization problem we propose a methodical approach that will help in identifying and including all important parameters and constraints that must be taken care of.

ACM Criteria Identification

Identifying important criteria is the key step for accommodating all critical processes that ACM should

address. Some of these criteria have been identified and categorized in Table 1. This list can be expanded as more information about the system is available. The first column lists generic criteria and the second lists some examples related to aircraft type systems, just for illustration purposes. Once an exhaustive list of such criteria has been compiled, we can proceed towards framing an optimization problem.

TABLE 1

IMPORTANT CRITERIA TO IDENTIFY BEFORE ACM DESIGN

Fault modes and their relative severity	- What actuators and valves can get stuck - Pressure sensors may be broken - Leakage of gas - Excessive temperature - Abnormal pressure increase - Structural failures (crack on surface, failed mechanical component) - Short circuit in electrical wiring or an open circuit
Mission critical components	- Constant fuel pressure in ignition chamber - Adequate thrust generation for the planned trajectory - Correct position sensing - Engine related hardware
Safety critical parameters	- Roll rate within bounds - Pitch rate within bounds - Smooth mode transitions - Sustainable body velocities (longitudinal, lateral and vertical) - Safety related hardware
Design specifications (sustainable material property limits for temperature, shear and compressive stress etc)	- Aircraft surface temperature should not rise beyond a limit. Thus speed may be altered based on atmospheric conditions - Inside air pressure should be bounded - Excitation frequencies for various critical structural modes - Rate of consumption of resources in different operating modes - Most efficient rates of consumption in nominal modes
Consumable resources and their availabilities	- Available fuel or any other energy sources - Available time to accomplish mission without running out any of the resources - Available computational memory, computational power and bandwidth of data channels
Available alternatives for mission critical and safety critical operations and their costs/rewards	- An alternative trajectory planning: may require longer time and more fuel - Re-routing a gas path to avoid stuck valve - Switching of some less important modules to conserve energy - Employ analytical redundancy to overcome dead sensors, may need more computations & bandwidth Alternative to abort the mission and come back

Framing the optimization problem

The objective function: Objectives of the ACM system (Fig 1) can be translated into several criteria, which can be characterized by analytical equations and computed using sensor measurements. For example:

- Reaching desired altitude → *min*{`distance from the desired orbit`}
- Adhere to preplanned trajectory → *min*{`trajectory tracking error`}
- Stability → *min*{`vibrations (jerks), change in orbit radius, change in sign of acceleration,`}, max{`smooth mode change operations`} etc.
- Minimal excitation of structural modes → *max*{`difference between excited frequencies and natural frequencies of the system`}
- No out-of-range values

Constraints: Conditions with which above objectives should be met translate into constraints

- No out-of-range values for the measured variables
- Observance of structural limits → {temperature of ignition chamber is range bounded}, {speed (drag force) while in atmosphere is range bounded}, {vibration amplitude is range bounded}
- Time to complete mission is upper bounded
- Rate of fuel consumption is upper bounded
- Only selective operational modes are permissible for the controller in a given stage of flight
- Certain controls can be of on-off (1-0) nature
- Decision must be taken before anticipated time-to-failure

These types of optimization problems may typically be non-linear and involve integer variables. However, there are several ways like linearity approximations and mixed-integer optimization which can be used to reduce these problems to simple ones for real-time applications.

ACM Strategy

An ACM guarded system can be represented by a Finite State Machine (FSM) as shown in Fig. 2. There can be multiple states in each of the three state spaces, but the general nature of transitions between different states can be described as mentioned below and depicted in Fig. 2.

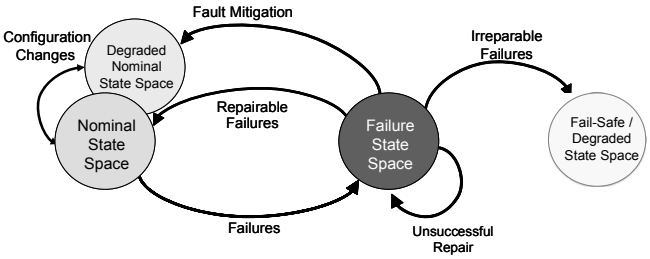

Fig. 2 ACM strategy for failure mitigation

Contingencies move the system to failure state. Repairable failures bring the system to the normal state, whereas irreparable failures will force the system to a fail-safe state to avoid further catastrophes and buy some extra time before external help can be sent, if possible. However, in case of faults that may not be completely repairable, ACM tries to find alternatives that will still let the system perform within acceptable limits but with degraded performance.

III. A CASE STUDY

Figure 3 depicts the overall scheme conceptualized for proof-of-concept demonstration using a Monopropellant Propulsion System (MPS).

Mission level objectives are translated into external commands, e.g. Move forward by *x* distance, increase speed, stop, etc., which will provide inputs to various components in the system model. Once a fault is detected, the stateflow model indicates the failure to the decision maker, which in turn requests the ACM simulator to provide possible corrective action sequences along with associated costs. The decision maker makes a decision based on specified criteria (currently the minimum cost). The corrective action is communicated and applied to the system model. Various fault injection options have also been included using a fault simulator that can simulate various faults like stuck valves, malfunctioning regulator or gas leakage, etc.

A Simulink® model for a Monopropellant Propulsion System (MPS) has been developed as a test bench for developing CBM/PHM methodologies with particular emphasis on ACM. This MPS model has been taken from NASA's Fault Tree Handbook [11] and has been slightly modified to suit the requirements of health management scenarios. The simulink model is equipped with a fault simulator to allow injecting various types of faults so that the ACM strategies can be validated and verified. Although this model is hypothetical and primarily qualitative, it incorporates most of the functional aspects that can be found in a propulsion system. Furthermore, its simplicity allows for quick implementation and experimentation to test and validate new algorithms.

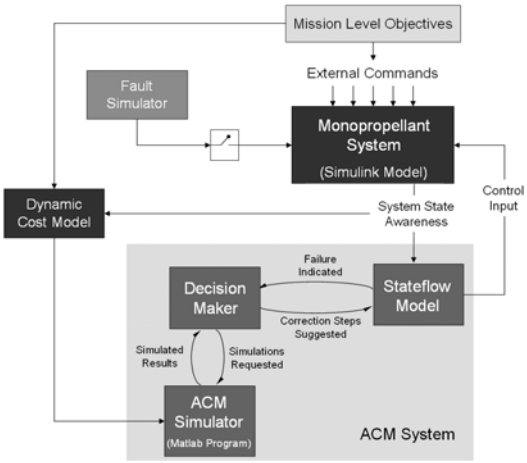

Fig. 3 PACM Test bench using MPS

A. Monopropellant Propulsion System (MPS)

The system uses hydrogen peroxide (H_2O_2) that passes over a catalyst and decomposes into oxygen, water, and heat, creating an expanding gas that produces the required thrust. The system includes a reservoir tank of inert gas that feeds through an isolation valve IV1 to a pressure regulator RG.

The pressure regulator senses the pressure downstream and opens or closes a valve to maintain the pressure at a given set point. Separating the inert gas from the propellant is a bladder that collapses as the propellant is depleted. The propellant is forced through a feed line to the thruster isolation valve IV2 and then to the thrust chamber isolation Valve IV3. For the thruster to fire, the system must first be armed, by opening the IV1 and IV2. After the system is armed, a command opens the IV3 and allows H_2O_2 to enter the thrust chamber. As the propellant passes over the catalyst, it decomposes producing oxygen, water vapor and heat. The mixture of hot expanding gases is allowed to escape through the thruster nozzle, which in turn creates the thrust. The relief valves RV1-4 are available to dump inert gas/propellant overboard should an overpressure condition occur in any corresponding part of the system.

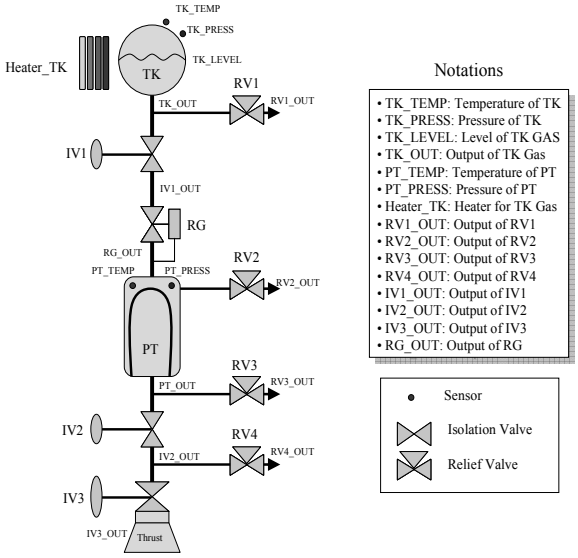

Fig. 4 NASA monopropellant propulsion system schematic

B. Simulink Modeling of the MPS

For simulation purposes the MPS model has been developed using MATLAB Simulink®, which provides an overall flexible structure to allow incorporating the state transitions associated with the ACM Strategies. This model consists of three modules that contribute to the purpose of simulation and testing during experimentation:

- System Model
- Fault Simulator
- Indicator (warnings and alarms)

The system model simulates the hypothetical monopropellant system. It consists of four subsystems namely the heater system, the tank system, the regulator system, and the valve system. Dynamic models of these components assume a simple first order response to any changes in their inputs. These equations can be later modified to represent more realistic dynamics, if needed.

Various parameters in these models can be modified easily to change the response of the components and thus, of the entire system. The purpose of the Fault Simulator Block is to inject fault signals and provide system inputs to the MPS. The Indicator Block displays the status of the system as inferred from various sensors such as temperature, pressure, etc. Some assumptions were made while designing the Simulink® model to simplify the modeling task. These assumptions have been listed in Table 2.

Two faults were simulated using the fault injection module, and the ACM system was expected to compute and apply an optimal fault mitigation strategy to overcome the situation. Several steps undertaken in this process are briefly described below.

TABLE 2
ASSUMPTIONS FOR MPS SIMULINK® MODEL

Components	Elements	Assumptions	Functional Description
Heater system	TK heater	▪ Current(I) = constant ▪ resistance(R) = constant ▪ heat(Q) depends on on/off time t	$Q = I^2Rt$
		▪ Transfer function is used for temperature system	$\frac{1}{s+1}$
Tank system	TK level TK pressure TK temp	▪ Tank volume(V)=constant ▪ Tank pressure(P) depends on temperature(T) and number of moles of gas(n)	$PV = nRT$
		▪ Tank level(L_T) is related to propellant consumption($\dot{n}$) by opening of IV3 and RV1-4. ▪ Initial Tank level(L_{ini}) =100	$L_T = (L_{ini} - \sum \dot{n})$
Regulator system	RG	▪ Output gas pressure(P_o) depends on input gas pressure(P_i) and set point(P_{SP})	$P_o = P_{SP}, P_i \geq P_{SP}$ $P_o = P_i, otherwise$
Valve system	IV1-3 RV1-4	▪ Each valve only has simplified as on/off status and the flow(n) depends on on/off time t ▪ The flow of gas or propellant (f_c) = constant	$\dot{n} = Pf_c$

C. Fault Scenarios

Two simple scenarios, consisting of two faults occurring in succession, were considered. These scenarios are described next.

1) Regulator failure

In a healthy condition the regulated pressure (P_{out}) is expected to follow the setpoint (P_{set}) as long as there is enough gas pressure (P_{in}) in the gas tank. As soon as P_{in} falls below P_{set}, the regulator can not maintain the desired P_{out} and it starts falling as input gas pressure depletes. Thus,

$P_{in} > P_{set} \rightarrow P_{out} = P_{set}$

$P_{in} < P_{set} \rightarrow P_{out} = P_{in}$

It can be assumed that the gas tank contains enough gas to provide the desired regulated pressure until the completion of the mission under normal conditions. Figure 5 shows that P_{in} should be always larger than P_{set} until the final mission time. The regulator failure occurs during the mission and the regulated pressure P_{out} drops to a lower value. The suggested fault mitigation strategy here is to increase the setpoint to a level that brings the regulated pressure back to the desired level.

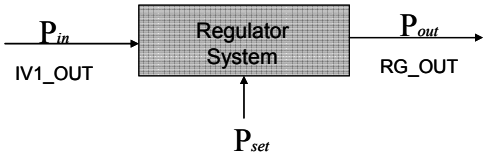

Fig. 5 Schematic of the simulated regulator system

Assuming that there are consequences (costs) associated with altering the set point, it may not be economical to alter the set point immediately. Therefore, depending on various costs such as stage (% completion) of the mission, amount of gas left in the tank, etc., a decision needs to be taken about when and by how much to raise the set point.

As can be seen from Fig. 6, increasing the set point leads to another problem where the tank gas pressure drops below the setpoint and the regulator can not maintain the pressure. The ACM strategy for this situation is to raise the temperature and hence the pressure of the gas in the inert gas tank. However, turning the heater ON is also an energy consuming process and therefore, the corresponding cost must be considered while deciding *when* to take the action so that the heater is not used for any extra amount of time than necessary. Furthermore, ACM must also decide when to switch off the heater once a safe level of gas pressure has been attained to accomplish the mission.

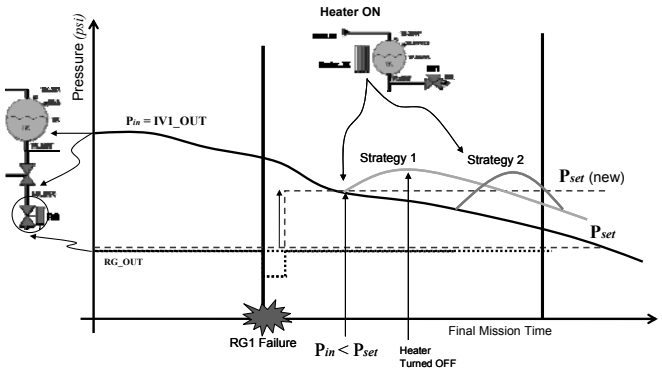

Fig. 6 Simulated regulator failure scenario

2) Heater failure

Figure 7 shows the heater failure scenario that follows the regulator failure. The failure occurs when the heater is commanded OFF by the ACM system but gets stuck in the ON position. As a consequence, the tank pressure increases to dangerously high levels. In this situation the ACM reacts by opening one of the relief valves (RV) to release the excess pressure. Once again the decision here involves deciding which RV to open, when to open it and for how long.

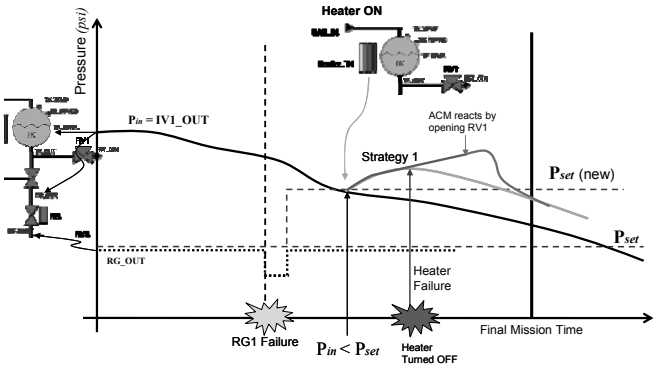

Fig. 7 Simulated heater failure scenario

D. Cost Modeling

For the above scenario, a simple cost model was developed. This model takes two factors into account in calculating the total costs. Whenever a fault occurs, these costs are calculated for all future time instants, and the action is taken whenever the sum total of these costs is the minimum. As shown in Fig. 8, the two factors considered are the cost of the time when the heater is "ON" and the cost of extra time required for completing the mission.

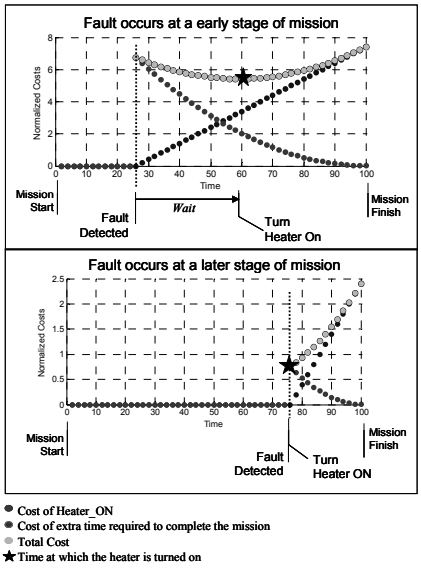

Fig. 8 Costs are dynamically computed and change depending on the current mission stage

After the failure occurs, the thrust level can not be maintained due to low pressure and the mission progress is slowed down. If a corrective action is not taken soon enough the mission would not be completed in the designated time. The corrective action itself incurs some cost and now the decision must minimize the sum of these low costs.

$$Total\ Cost = 0.0009 \cdot t^2_{Heater\_ON} + 0.05 \cdot (t_{final} - t_{scheduled})^2 \qquad (1)$$

Figure 8 shows two scenarios each with a fault occurring at an early and a later stage of the mission. As can be seen, if the fault occurs in the early stage, the heater need not be turned on immediately whereas if the fault occurs towards the end heater can be immediately turned on.

Once other cost factors are available, a composite cost function can be formulated and incorporated in the decision making process.

E. ACM Model

The ACM model has been developed using Matlab Stateflow® toolbox. Figure 9 shows the stateflow diagram for the fault scenarios described above. Whenever the system makes a transition from the normal mode to a fault mode the costs are computed and the action is taken whenever the total costs are minimized.

The stateflow model directly interacts with the Simulink® model to assess the state of the system. Variables like pressures, temperatures, percentage of mission completion, fuel level, etc., are continuously monitored and as soon as something goes beyond normal levels an action is applied until the abnormal behavior is sufficiently corrected.

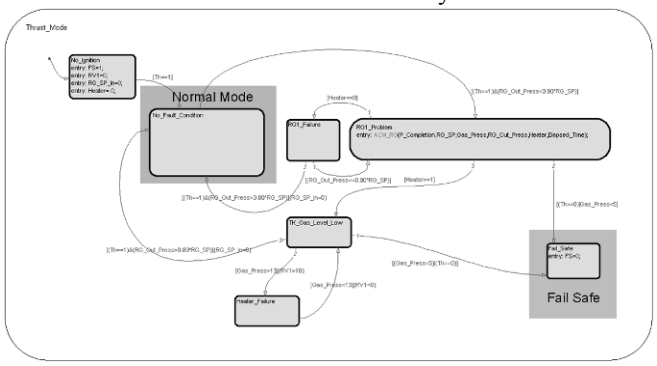

Fig. 9 ACM model in Stateflow

IV. Simulation and Results

For the purpose of demonstrating various attributes of the ACM system, three simulations scenarios were considered.

A. Simulation Scenario #1

Figure 10 shows six plots to visualize changes occurring in the system for the first simulation scenario. The first plot shows that system commanded to move forward and hence the thrust mode is turned on at $t = 10$. Mission progress is shown in plot 2 (from 0% to 100%). Plot 3 shows the inert-gas tank pressure which starts depleting as soon as the thrust is on. The regulator fault occurs at $t = 26$, causing the regulated pressure to drop (plot 4). Plots 5 and 6 show gas temperature and gas pressure respectively, in the tank. The ACM reacts immediately ($t = 27$) by increasing the set point from 9 (psi) to 11.5 (psi) as seen in plot 3 and hence, the regulated pressure is corrected. Around $t = 42$ the gas pressure falls below the new set point and the regulated pressure starts dropping again. The ACM reacts by turning the heater on at $t = 59$. This shows that in this situation the ACM prefers to wait for sometime before the heater is

turned on. The heater is turned off again at $t = 69$. However, at $t = 92$ gas pressure again falls below set point. In this situation the heater is turned on almost immediately ($t = 94$).

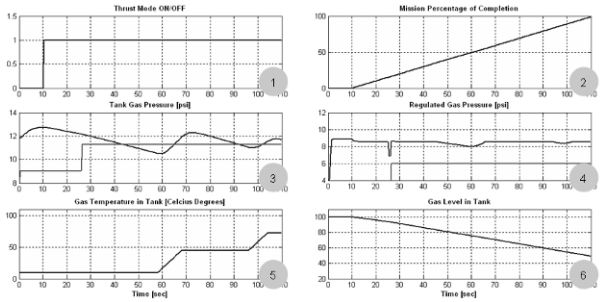

Fig. 10 Regulator fault scenario simulation

This simulation scenario shows that, depending upon the stage of the mission when the fault occurs, ACM decisions will vary based on total costs calculated there on.

B. Simulation Scenario #2

This simulation shows in this case that the ACM continuously monitors the system, and reacts as soon as the fault occurs. But it also illustrates the fact that the ACM retracts its actions quickly, if somehow the fault is removed from the system. Furthermore, if the fault occurs towards the end of the mission it may not even apply the corrective action, if this is more expensive than the cost of extra time required for completing the mission. The latter, mainly because only a very small percentage of mission completion remains towards the end, and therefore the extra time that may be required (due to degraded performance) may not be too much when compared to the explicit cost of applying the corrective action (altering the set point in this case).

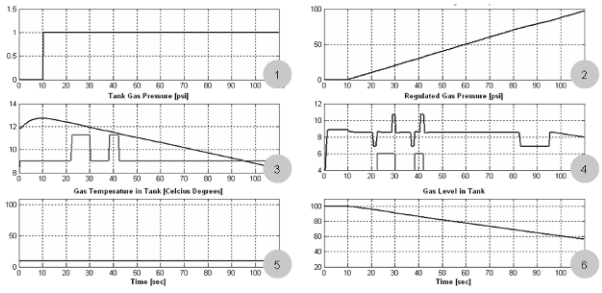

Fig. 11 ACM continuously monitors the system

C. Simulation Scenario #3

This simulation shows the performance of the system in a scenario where a heater failure occurs after the regulator fault. The expected outcome of this situation is an elevated temperature (and pressure) inside the inert-gas tank. As can be seen in Fig. 12, plot 3, the heater fails to turn off at $t = 69$, which is recognized as a critical failure by the ACM. As a result, the ACM reacts soon ($t = 72$) by opening the relief valve 1 (RV1) for two seconds. The corresponding effect can be seen in plot 6 where the gas level decreases quickly when RV1 is opened. Since the temperature is still rising, the inert-gas pressure shoots beyond safe levels for a second time and RV1 is opened again ($t = 82$) to release some gas.

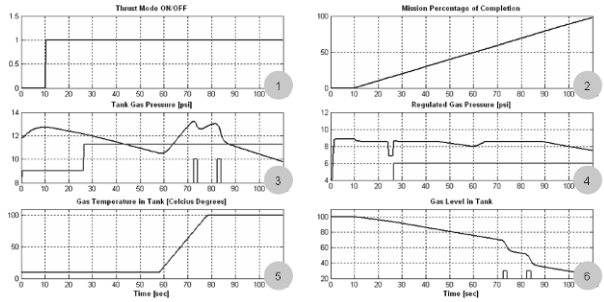

Fig. 12 Regulator and heater faults in succession scenario

V. ACM VALIDATION

Although the ACM was designed to perform optimally for the described fault scenarios, results provided in Section IV cannot guarantee that the optimality criteria have been met. In order to solve this issue, a validation procedure has been applied, randomizing the time instant when the fault occurs, and comparing the performance of an optimal ACM strategy (grey line) – implemented by using (1) as an objective function – against another ACM system (black line) lacking such optimization.

Figure 13 shows the obtained results in terms of the *cost of extra time* (needed to complete the mission). Clearly, both ACM systems (the optimal and the non-optimal) are able to mitigate the effect of the pressure regulator failure. Since the optimal contingency strategy balances the cost of extra time and heater usage, it is expected that the cost associated to the former would be higher. It is also important to mention that the effect of the optimal strategy is more significant for faults occurring in the early stages of the mission, see Fig. 13.

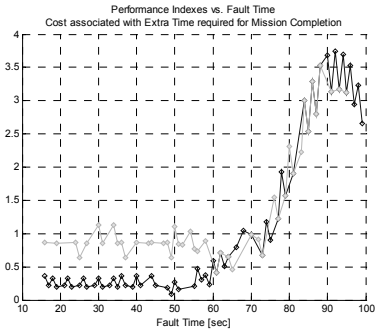

Fig. 13 Cost of extra time, given a failure in the pressure regulator

Figure 14 shows the validation results in terms of the *cost of heater usage*. It is evident that the optimization routine is able to finish the mission in approximately the same time, but saving a significant amount of energy.

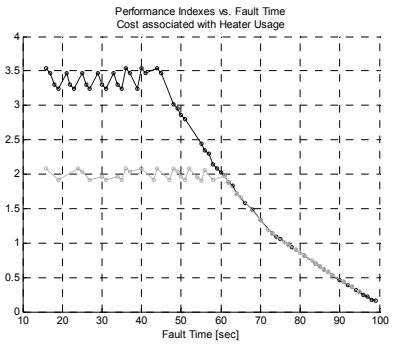

Fig. 14 Cost associated with heater usage, given a failure in the pressure regulator

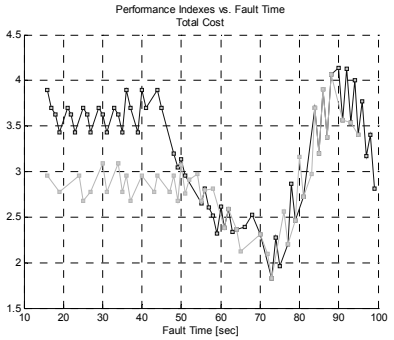

Fig. 15 Total cost, given a failure in the pressure regulator

As the final result, Fig. 15 shows the total cost for both the non-optimal (black line) and optimal (grey line) ACMs; in that sense, validation results indicate that, generally speaking, the optimization routine is fulfilling its purpose. Last but not the least, if the weight of the cost associated to extra time in the objective function were to increase, then we would observe that the performance of the optimal ACM would converge to the non-optimal one.

VI. Conclusion

In this paper we presented a generic prototype test bench software for developing and validating ACM systems for advanced propulsion systems called the Propulsion ACM (PACM). A proof-of-concept case was implemented for a Monopropellant Propulsion System (MPS) to show the validity of the approach. It was shown that ACM system is capable of mitigating the failures by searching for an optimal mitigating strategy and that the strategies change with time depending on the stage of the mission. A generic approach to develop such systems has been described and supported with the above mentioned example. Further, the concepts of Validation for such systems were introduced with relevant examples.

Acknowledgment

We gratefully acknowledge the support from NASA, Ames Research Center. The work reported in this paper was conducted under the SBIR Phase II program, Automated Contingency Management for Advanced Propulsion Systems, with Impact Technologies as prime contractors.

References

[1] Tumer, I. and Bajwa, A., "A Survey of Aircraft Engine Health Monitoring Systems", *the 35th AIAA/ASME/ SAE/ASEE Joint Propulsion Conference,* Los Angeles, CA., June 20-23, 1999

[2] Liang Tang, Gregory J. Kacprzynski, Michael J. Roemer, George Vachtsevanos and Ann Patterson-Hine, "Automated Contingency Management Design for Advanced Propulsion Systems", *infotech@Aerospace*, Arlington, Virginia. 26 - 29 September 2005.

[3] Byington, C. S., Watson, M., Roemer, M. J., Galie, T. R., and McGroarty, J. J., "Prognostic enhancements to gas turbine diagnostic systems," *IEEE Aerospace Conference*, Big Sky, Montana, March 2003

[4] Boller, C., "Next generation structural health monitoring and its integration into aircraft design," *International Journal of Systems Science*, Vol. 31(11), 2000. pp. 1333-1349

[5] Jianhua G., Roemer, M.J. Vachtsevanos, G., "An automated contingency management simulation environment for integrated health management and control", *Proceedings of the IEEE Aerospace Conference*, 2004.

[6] Carter W. C., "A time for reflection", *In Proceedings of the 12th IEEE Int. Symposium on Fault Tolerant Computing (FTCS-12)*, page 41, Santa Monica, CA, USA, June 1982.

[7] Laprie, J. C. (ed.), *Dependability: Basic Concepts and Terminology*, Vienna, Springer-Verlag, 1992.

[8] Heimerdinger, W. L., and Weinstock, C. B., "A conceptual framework for system fault tolerance", *CMU/SEI-92-TR-33*, October 1992, Software Engineering Institute, Carnegie Mellon University

[9] William, P.C., and Nayak, P.P., "A Model-based Approach to Reactive Self-Configuring systems". *Workshop on Logic-Based Artificial Intelligence,* Washington, DC, June 14--16, 1999

[10] Bateman, A.J., Elks, C.R., Ward, D.G., Schierman, J.D, "New Verifcation and Validation Methods for Guidance - Control of Advanced Autonomous Systems", *AIAA 2005-7117*, September 2005, Arlington, VA

[11] Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick III, J., Railsback J., "Fault tree handbook with Aerospace Applications" version 1.1, August 2002