

Prognostics in the Control Loop

Liang Tang*, Gregory J. Kacprzynski*, Kai Goebel[‡],
Johan Reimann*, Marcos E. Orchard[†],
Abhinav Saxena[†], and Bhaskar Saha[‡]

*Impact Technologies, LLC. 200 Canal View Blvd. Rochester, NY 14623

[†]Intelligent Control Systems Lab, Georgia Institute of Technology, Atlanta, GA

[‡]NASA Ames Research Center, Moffett Field, CA.

Abstract

The term Automated Contingency Management (ACM) has been used to describe intelligent systems capable of mission re-planning and control reconfiguration in the presence of a current health state diagnosis. While a diagnostics driven ACM capability designed to optimize multi-objective performance criteria remains a significant technical challenge, it cannot hope to overcome the fact that it will always be a reactive paradigm. This paper, therefore, introduces an automated contingency management paradigm based on both current health state (diagnosis) and future health state estimates (prognosis). Including Prognostics in the control loop poses at least two additional challenges to ACM. First, future state prediction will, in general, have uncertainty that increases as the prediction horizon increases so adaptive prognosis routines that manage uncertainty are critical. Secondly, a warning period afforded by prognosis allows ACM to be split into a real-time “reactive” component and a non-real time “planning” component that considers temporal parameters and the potential impact of being proactive with mitigating action. The proactive ACM paradigm was developed and evaluated in the context of a generic mono-propellant system model in Simulink/Stateflow with diagnostics, prognostics and an optimal reconfigurable control system. Applications of Artificial Intelligence (AI) technologies in prognostics enhanced ACM system are briefly discussed. Preliminary results from the on-going research work are presented and the paper is concluded with remarks on future work.

Introduction

Stimulated by the growing demand for improving the reliability and survivability of safety-critical aerospace systems, a variety of Integrated Vehicle Health Management (IVHM) and fault tolerant control techniques have been developed. Techniques that are capable of detecting the occurrence of faults while still retaining acceptable performance in the presence of faults are being developed for both manned and unmanned air vehicles. The concept of using system health information (diagnosis)

in conjunction with reconfigurable control has been introduced through different techniques at various levels of sophistication ranging from engine controls, to flight controls and mission reconfiguration (Vachtsevanos et al. 2007, Byington et al. 2004, Litt, Parker and Chatterjee 2003).

Automated Contingency Management (ACM) system architecture provides a framework to accommodate the integration of prognosis & health management (PHM) and control reconfiguration techniques (Tang et al. 2005 & 2007). Most fault detection and fault accommodating control techniques found in literature can be categorized as ACM systems at certain levels. While a diagnostics driven ACM can react to and compensate for faults and performance degradation after they are detected, it cannot hope to overcome the fact that it will always be a reactive paradigm. This paper introduces a proactive ACM paradigm based on both current health state (diagnosis) and future health state estimates (prognosis). Including Prognostics in the control loop poses several challenges. First, future state prediction will have uncertainty that increases as the prediction horizon increases so adaptive prognosis routines that manage uncertainty are critical. Secondly, a warning period afforded by prognosis allows ACM to be split into a real-time “reactive” component and a non-real-time “planning” component that considers temporal parameters and the potential impact of being proactive with mitigating action. The purpose of this paper is to introduce an approach for integration of prognostics data with ACM system as prognostics becomes readily available by the introduction of IVHM techniques.

The organization of this paper is as follows: First, an overview of ACM system is briefly introduced followed by an optimization-based ACM design methodology and a finite state machine based ACM modeling paradigm. The following sections address the issue of integrating prognostics in both high level ACM planner and low level control reconfiguration. A pressure-fed monopropellant propulsion system for a small spacecraft is utilized as an initial proof-of-concept implementation for the proposed techniques and preliminary simulation results are

presented. The paper concludes with remarks on the technical challenges and future developments.

Prognostics Enhancement to ACM

Conceptually, ACM refers to a system that is designed to provide the ability to confidently and autonomously adapt to fault and/or contingency conditions while either achieving all or an acceptable subset of the mission objectives (Roemer et al. 2006). An ACM system is differentiated from a fault tolerant control system in that it consists of not only low level control reconfiguration, but also high level (mission) planning and optimization. A typical ACM implementation usually utilizes a hierarchical architecture that covers low level redundancy management, mid level fault accommodation strategies, and high level adaptive mission re-planning modules. Figure 1 shows the high level conceptual schematic of the interaction between the PHM and ACM system. The PHM and situation awareness modules provide fault diagnostics, prognosis and contingency information to the ACM system, which in turn, identifies and executes the optimal contingency mitigation strategies.

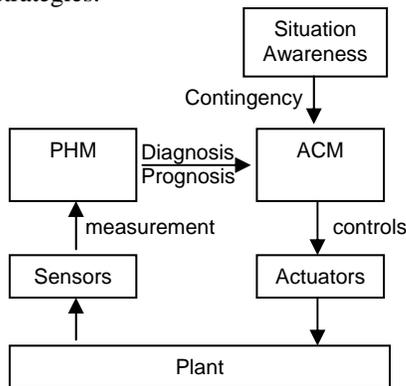


Figure 1 : PHM & ACM Systems

Real world implementations of ACM systems for aircraft, spacecraft or, on a smaller scale, their propulsion systems, are based on a variety of problem-specific solutions. Conceptually, ACM strategies can be implemented within a generic hierarchical architecture shown in Figure 2. This approach relies not only on current system performance/fault information (diagnosis) but also incorporates the projected future condition of the system (prognosis). By incorporating the likely future condition of the system into the ACM routine, it is possible to assess the likelihood of accomplishing a given set of objectives and, if necessary, change the objectives to avoid catastrophic failures.

The ACM routine manages detected faults in three different ways. 1) Complete low-level reconfiguration will be utilized if the fault is located in a subsystem for which there exists a redundant backup system. 2) Complete high-level re-planning will be performed if the fault is located in

a subsystem for which no redundancy exists. 3) Partial low-level reconfiguration and partial high-level re-planning will be performed if indirect redundancy is present at the lower level, that is, the system’s operational capabilities will be partially restored by utilizing other subsystems to compensate for the fault, and the overall objectives will be modified based on the degraded system’s capabilities.

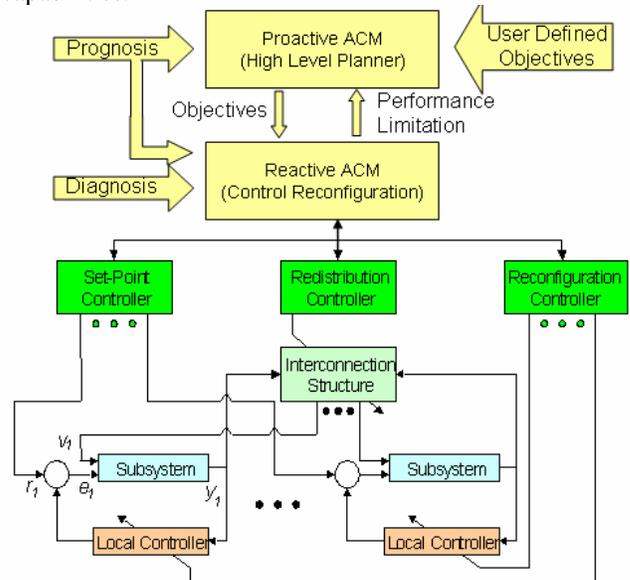


Figure 2 : Prognostics Enhanced ACM Architecture

To effectively perform ACM, diagnostic and prognostic information must be provided. The diagnostic information regarding the system’s current fault condition will be used to assess immediate capabilities of the system, while the prognostics information will be used to establish realistic long term resource management and mission planning. The high level planner enables the ACM routine to not only address immediate fault conditions but also establish a long term operational plan that will optimize the utility of the entire system.

Optimization Based ACM System

Most ACM strategies for large-scale dynamical systems, like aircraft, propulsion systems, etc., rely on heuristic information about a reduced set of severe, frequent and testable fault modes, a reasonable number of active controllers and a mapping between the fault modes and the control reconfiguration routines. The strategy has the ability to adaptively switch from one controller to another, if control limits are reached and by this switching action critical mission objectives can be realized. In this paper, we present a new approach in which an optimization problem is dynamically formulated and solved on-line to solve the optimal contingency strategy constrained by the available performance and resource. A typical cost model is expressed in terms of such elements as critical

component on-time, time to complete the mission, fuel consumption etc. Costs are computed in real-time and may change dynamically.

Analytically, the objective of the ACM system is to optimize the utility of the vehicle with impaired capability to accomplish an assigned mission. The ACM system can be formulated as an optimization problem in two levels,

High Level Planner:

$$J(M) = \max_M U(P_e, P_r, M, M_{com}) \quad (1)$$

Control Reconfiguration (lower) level:

$$J(R) = \max_R P_e(F_m, P_r, R, M) \quad (2)$$

where U is a cost function that quantifies the usefulness of the vehicle to accomplish its mission. U is a function of the available prognostic information (Pr), the system's closed loop performance (Pe), and the mission objectives (M and $Mcom$). Pe is a function of fault mode Fm , future prediction Pr , as well as any re-structuring/ reconfiguration R applied to the system and current mission objective M . Fm is a vector of indicators (0 or 1) that characterizes the fault modes detected on the aircraft; R is a vector of indicators that characterizes all restructuring applied to the system. $Mcom$ describes the mission assigned to the aircraft. M allows the fault-tolerant control architecture, specifically the mission adaptation and resource management components, to modify the parameters of the assigned mission and redistribute the of available resources based on vehicle's current performance, Pe . At the high level, mission adaptation and resource redistribution (M) allows the control architecture to pursue relaxed mission objectives in order to achieve greater vehicle usefulness U . At the lower level, the objective is to optimize vehicle performance Pe while satisfying the mission constraints, through restructuring and reconfiguration, R . Practically, the above optimization problems have to be solved while adhering to various constraints including system dynamics and resource limitations.

To facilitate the formulation of the optimization problem, an ACM guarded system can be represented by a Finite State Machine (FSM) as shown in Figure 3. There can be multiple states in each of the three state-spaces, but the general nature of transitions between different states can be described by five types of transitions as depicted.

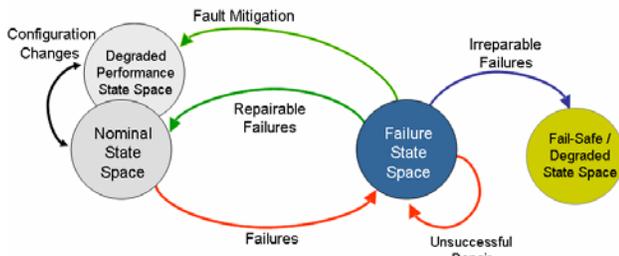


Figure 3 : ACM Modeling

Contingencies may move the system to a failure state, while repairable failures allow the system to eventually come back to the normal state. Irreparable failures may force the system to a failsafe state to avoid further catastrophes and buy some extra time before external help can be sent. 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.

With the modeling paradigm described above, the ACM algorithm can be formulated as a constrained optimization problem as stated: Given the current states of the system, and subject to predefined system constraints, find the optimal action series that will bring the system to the desired states with a minimal cost.

High Level ACM Planner

Prognostic information concerning the changes in the fault condition will in the future enable the ACM algorithm to derive long term strategies and thereby minimize the overall likelihood of a catastrophic failure. The high level ACM planner responsible for mission re-planning and resource/load redistribution can take advantage of the prognosis information to repeatedly optimize the mission objectives and the available resources as the fault develops. To illustrate how prognostics can be integrated into the optimization problem, a simplified mission planning problem is used as an example. Suppose the aircraft (e.g. an UAV) is assigned k tasks: $S = \{s_1, s_2, \dots, s_k\}$. The importance of task s_i is denoted c_i and the expected time and resource (e.g. fuel) required to accomplish the task is t_i and r_i respectively. The mission planning routine will find the sequence of tasks that maximizes the mission success criterion subject to time and resource constraints. For simplicity, it is assumed that the tasks are independent of one another. The optimization problem can therefore be formulated as,

$$\begin{aligned} \max_P C \cdot P \\ \text{s.t. } T \cdot P \leq T_{lim} \\ R \cdot P \leq R_{lim} \end{aligned} \quad (3)$$

Where, $C = [c_1, c_2, \dots, c_k]$, $T = [t_1, t_2, \dots, t_k]$, $R = [r_1, r_2, \dots, r_k]$, T_{lim} and R_{lim} are the maximum time and resource allowed for all tasks; $P = [p_1, p_2, \dots, p_k]^T$, $p_i \in \{0,1\}$ is the decision variable. When $p_i = 1$, task s_i is chosen.

With this simple problem formulation, prognosis information can potentially affect T, R, T_{lim} and R_{lim} . For example, if an engine or a flight actuator is degrading, the degraded performance of the vehicle will affect the expect

time to accomplish the tasks (T) and the time allowed for this group of tasks (T_{lim}) will be reduced to make sure there is enough time for successive missions. Similarly, a leakage in the fuel tank will affect R and R_{lim} . To keep the mission plan updated, the optimization routine must be executed repeatedly to accommodate for changes in the output of the prognostic routine.

To capture the uncertainties inherently present in the prognostics routine, the high level optimization can be formulated as a stochastic programming problem. Stochastic programming provides a framework for modeling optimization problems that involve uncertainty. The consideration of uncertainty in stochastic programming is very important since the future state prediction has considerable uncertainty that increases as the prediction horizon increases. To accomplish this, a multistage look-ahead procedure will be added to the Hierarchical ACM. The underlying idea is as follows: Given the current and predicted state of the system it should be determined if it is possible to improve the overall system performance over a given time horizon T by taking a calculated risk early on. The deterministic optimization approach without consideration of prognostic information may in fact not be optimal over a longer period of time.

Essentially, the considered time horizon is broken down into n pieces, and at each stage the prognostics routine provides the information regarding the status of the system fault, that is, the current probability density function associated with the fault will be propagated through time and sampled at each decision instance. Hence, when determining the best possible course of action, the prognostics routines will be executed in parallel with the ACM routine to improve the expected cost.

Control Reconfiguration with Prognostics Consideration

Receding horizon control (RHC) has been proposed as a method for reconfiguration due to its ability to handle constraints and changing model dynamics systematically. RHC relies on an internal model of the system, which can be identified online in real-time (Tang, Roemer and Kacprzyński, 2007).

Reconfigurable RHC control can be applied to accommodate both actuator failure and structural failures caused by hostile action or hazardous atmospheric weather conditions. These failures can be handled naturally in a RHC framework via changes in the input constraints and internal model. Prognostic information can be integrated into the controls as soft constraints on control variables.

For example, actuator limit and rate constraints can be written as:

$$\begin{aligned} u_i^l &\leq u_i(t) \leq u_i^h \\ \Delta u_i^l &\leq \dot{u}_i(t) \leq \Delta u_i^h \end{aligned} \quad (4)$$

for actuator inputs u_i through u_m . If actuator i becomes jammed at position \tilde{u}_i , the RHC controller can be made to accommodate the changes by simply changing the constraints on input i to

$$\begin{aligned} \tilde{u}_i &\leq u_i(t) \leq \tilde{u}_i \\ 0 &\leq \dot{u}_i(t) \leq 0 \end{aligned} \quad (5)$$

Moreover, if prognosis of actuator i predicts that the actuator may get stuck in the near future, the RHC controller can help the actuator to get stuck in a preferable position (usually the neutral position) by setting the following constraints,

$$\begin{aligned} u_0 - du_0 &\leq u_i(t) \leq u_0 + du_0 \\ \Delta u_0^l &\leq \dot{u}_i(t) \leq \Delta u_0^h \end{aligned} \quad (6)$$

where du_0 is the reduced range for actuator i around the neutral position u_0 , and is a function of the remaining useful life distribution. Adding constraints on \dot{u}_i can also help to mitigate or defer actuator failure by avoiding aggressive control signals.

Typically, the receding horizon optimal control is derived by solving the receding horizon optimization problem,

$$\min J = \int_{t_0}^{t_f} \frac{1}{2} [x^T Q_x x + u^T Q_u u] dt \quad (7)$$

where u is the controls to be solved, x is an augmented state vector that includes the reference model states, identified plant states, and states of a state space model that integrates tracking error dynamics,

$$x^T = [x_r^T \quad x_p^T \quad x_i^T] \quad (8)$$

x_r^T are the states of the reference model defined by (δ is the pilot command),

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r \delta \\ y_r &= C_r x_r \end{aligned} \quad (9)$$

x_p^T are the states of the identified system. x_i^T are the states of a state space model that integrates tracking error dynamics,

$$\begin{aligned} \dot{x}_i &= C_r x_r - C_p x_p \\ y_i &= C_i x_i \end{aligned} \quad (10)$$

Matrix Q_x is defined as,

$$Q_x = \begin{bmatrix} C_r^T Q_T C_r & -C_p^T Q_T C_p & 0 \\ -C_r^T Q_T C_p & C_p^T Q_T C_p & 0 \\ 0 & 0 & C_i^T Q_i C_i \end{bmatrix} \quad (11)$$

where Q_T and Q_I are symmetric positive semidefinite matrices that assign importance to predicted and integrated predicted tracking error, and Q_u is a symmetric positive definite matrix that penalizes control usage.

System constraints are appended to the integrand of the objective function, and extremization of the integral results in matrix and vector differential Riccati equations which are solved at each control update based on the most recently identified aircraft dynamics.

The reference models for the retrofit reconfiguration algorithm are implemented as low order equivalent system transfer functions that output pitch rate, roll rate, and yaw rate for the controller to track. The natural frequencies, damping ratios, and transfer function gains define the response characteristics of the reference dynamics (Monaco, Ward, and Bateman, 2004).

The Application of AI Techniques

There are many potential applications of Artificial Intelligence (AI) technologies in a Prognostics enhanced ACM system. An expert system is an efficient way to model rule-based contingency strategies. It is anticipated that early integration of prognostics based fault mitigation strategies are in the form of expert specified rules. The test scenario #2 in the case study presented in the following section is an example of rule-based contingency strategy, where the ACM system switch the fuel path to the redundant path in the idle mode to avoid propellant tank overpressurization in successive thrust mode. Fuzzy logic can be utilized to set the actuator movement constraints du_0 , Δu_0^l and Δu_0^h in the RHC problem presented in the previous section. Computational intelligence techniques such as genetic algorithms and particle swarm optimization can be applied to the optimization problems in the high level ACM planner.

Case Study

Figure 4 depicts the overall scheme conceptualized for proof-of-concept demonstration using a Monopropellant Propulsion System (MPS, shown in Figure 5) (Saxena et al. 2007). 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 model 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 applied to the system. Various fault injection options have also been included using a fault simulator that can simulate various faults like stuck valves, malfunctioning regulator valve, malfunctioning heater or gas leakage, etc.

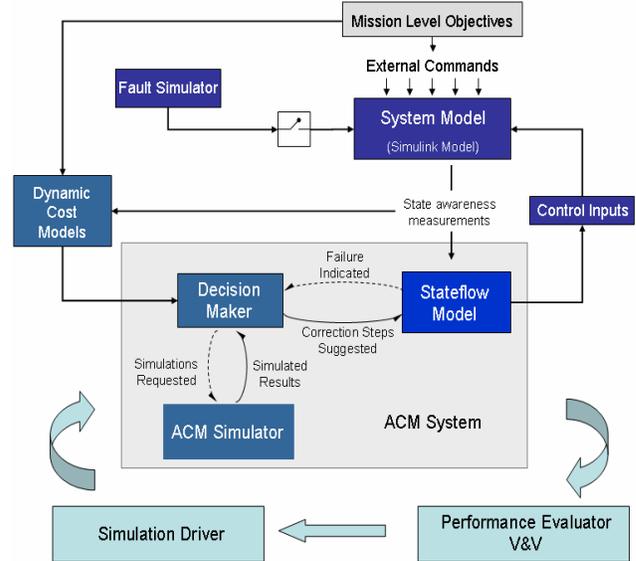


Figure 4: Optimization-based ACM & Test Bench

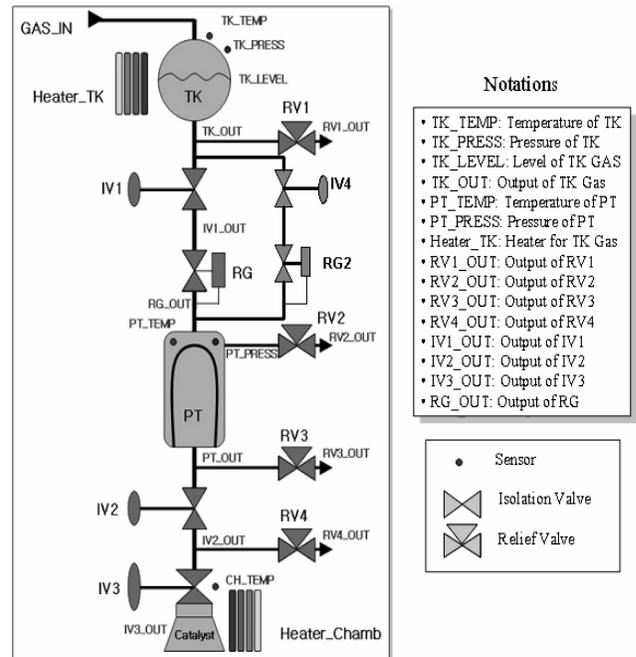


Figure 5: A generic monopropellant propulsion system schematic

A Simulink® model for MPS has been developed as a test bench for developing PHM methodologies with particular emphasis on ACM. This MPS model has been taken from NASA's Fault Tree Handbook (Vesely et al. 2002) 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.

Monopropellant Propulsion System (MPS)

The system uses hydrogen peroxide (H2O2) 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 valve 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 H2O2 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.

The ACM model has been developed using Matlab Stateflow™ toolbox. Figure 6 shows a part of the Stateflow diagram that covers a heater fault (stuck on) and a regulator valve fault (stuck open). Whenever the system makes a transition from the normal mode to a fault mode the costs are computed and the action is taken at the moment the total costs are the minimum.

As a proof-of-concept, a simple cost model was developed. This model takes two factors into account in calculating the total costs, i.e. fuel consumption and time to accomplish the mission. Total Cost = $w_1 * \text{time}(\text{heater_on})^2 + w_2 * \text{cost}(\text{extra time to complete mission})$

Figure 7 shows two scenarios each with fault occurring at an early and a later stage of the mission. As can be seen, if the fault occurs in the early stage of the mission, the heater need not be turned on immediately whereas if the fault occurs towards the end, the heater should be immediately turned on. This example also illustrated the advantage of optimization-based ACM design compared to an ACM

system based only on predetermined rules. Once other cost factors need to be considered, a composite cost function can be formulated and incorporated in the decision making process.

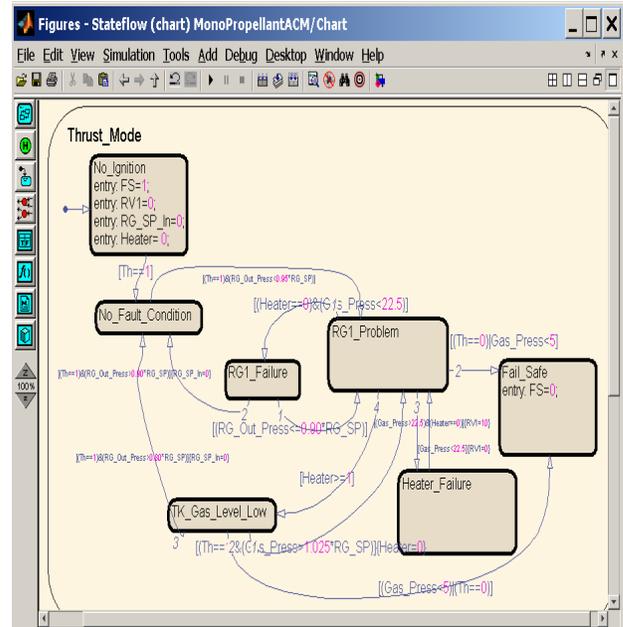


Figure 6: ACM Model in Stateflow

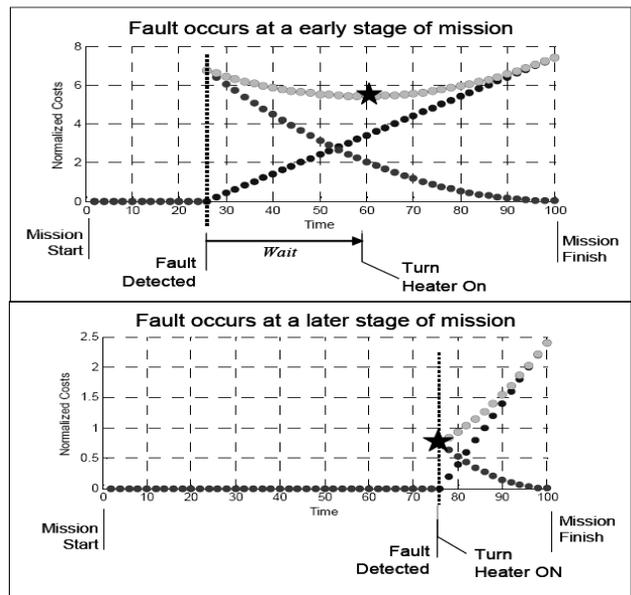


Figure 7: Cost Model

The developed simulation platform has been used to evaluate how prognostics can be effectively utilized in the controls of the monopropellant system. Seal leakage in an isolation valve (IV1) and pressure sensing degradation in a

regulator valve (RG) are simulated. The pressure sensing degradation in RG is caused by corrosion that clogs the sensing port of the regulator valve which leads to a stuck open failure. The proof-of-concept demonstration showed that the proactive ACM system could switch the fuel path to the redundant path in the idle mode to avoid a dangerous high pressure in the propellant tank in the successive thrust mode (scenario 1 and 2). In addition, the simulation results also demonstrated that by considering the prognosis of seal leakage severity level, the mission requirements could be relaxed to secure most important tasks (scenario 3).

Case Scenario #1

This scenario considers a mission profile that includes three thrust periods: the first one (duration 50 cycles) is intended to lower the orbit of the spacecraft, the second period (20 cycles) is used to reach the desired altitude for the mission and to return to the intermediate orbit, and final thrust periods puts the spacecraft back into the original orbit.

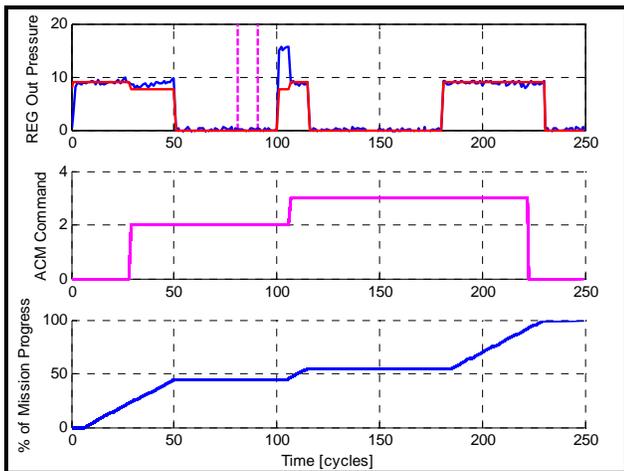


Figure 8 - Simulation results for RG fault without Prognostic module

Figure 8 shows the results of the simulation when the prognosis results are not considered. At the 100th cycle, the pressure at the output of the regulator valve reaches dangerous levels because RG is stuck open due to accumulated corrosion around the sensing port. This fault propagates and leads to over-pressurization in the propellant tank (PT). The ACM system reacts by switching to the redundant regulator valve (second plot in Figure 8), which prevents the condition from getting worse. However, obviously the reactive ACM system is not able to prevent a sudden change of pressure after the idle mode. Note that in the idle mode, the fault can't hopefully be detected because the regulator valve (RG) is not in operation. Prognosis of the corrosion growth is the only clue that should be considered in the ACM system as shown in scenario 2.

Case Scenario #2

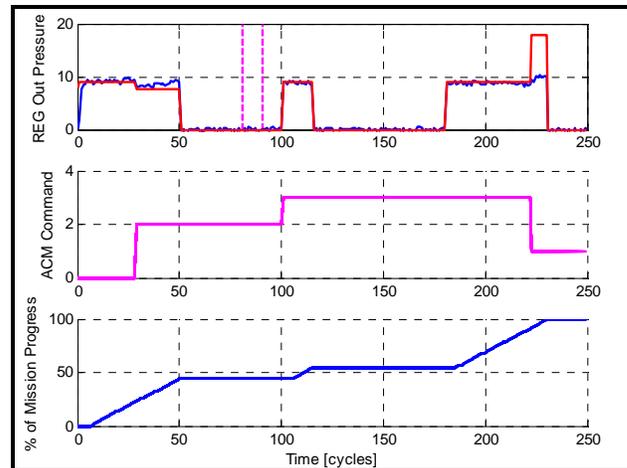


Figure 9 - Simulation results for RG fault with Prognostic module

Figure 9 shows the results of the simulation when the prognosis results are taken into account by the ACM system. Clearly the system does not undergo the same sudden change in pressure observed in Figure 8. It can be observed that the ACM system is allowed to take action (and switch to the redundant regulator valve) much earlier. The magenta-dotted-vertical lines in plot illustrate the confidence interval for the remaining useful life of the regulator valve till it got stuck open.

Case Scenario #3

The scenario simulates a severe leakage condition in IV4 (the system is already using the redundant fuel path). There is also the problem of sensor degradation in the regulator valve (RG2) that is accommodated by adjusting its set point value. Therefore, the mission reconfiguration in the ACM system decides to skip the second thrust mode as it can be seen in Figure 10.

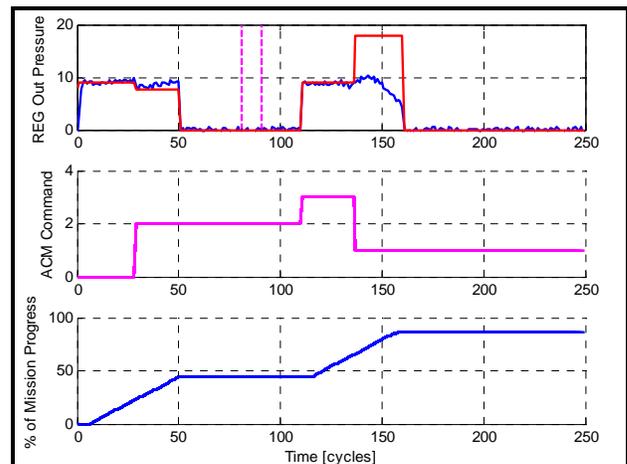


Figure 10 - Mission re-planning simulation results

Conclusion & Future Work

This paper presented an overview of the integration of prognostics in the planning and control loop in the context of an Automated Contingency Management for complex systems like modern aircraft and spacecraft. Applications of artificial intelligence techniques were also discussed. The main contribution is a generic hierarchical ACM architecture and novel optimization-based solutions for both high level ACM planner and control reconfiguration that incorporate prognostic information. A simple proof-of-concept example was presented to provide insight into the utility of the suggested ACM techniques. The proposed techniques will be matured through further development and evaluated on a real application.

Prognostics uncertainty management is an important issue that affects the performance of a Prognostics enhanced ACM system significantly. In practice, accurate prognostics has proven rather difficult to accomplish, thus uncertainty management and reduction techniques have to be implemented before the prognostics can be confidently utilized by the ACM system. In this paper, the stochastic programming approach for high level planner, and the fuzzy logic rule for setting constraints for actuator movement in the RHC approach take prognostics uncertainty into consideration, but many technical issues remain to be solved. Reducing uncertainty in failure prediction is being accomplished in the prognosis by the integration of many techniques including the integration of multi-discipline models, introduction of new sensors for state awareness and damage detection and, more importantly, reasoners to fuse failure models, usage history, environmental conditions, current state, and planned near-term operational use into predicted capability. Accurate and precise prognosis demands good probabilistic models of the fault growth and statistically sufficient samples of failure data to assist in training, validating and fine tuning prognostic algorithms. Many accomplishments have been reported but major challenges still remain to be addressed (Goebel and Eklund, 2007). Impact Technologies is actively involved in this research area and is developing a Probabilistic Modeling and Analysis Toolbox to support on-line prognosis uncertainty management.

Acknowledgement

This work is partially supported by the NASA Ames Research Center in Moffett Field, CA, under STTR contract No. NNA05AC53C. The Impact/GaTech team gratefully acknowledges Dr. Ann Patterson-Hine for her support. We would also like to recognize the contributions of Mr. Michael Koelemay from Impact Technologies, LLC, and Dr. Biqing Wu, Dr. Bin Zhang and Dr. Young Jin Lee from Georgia Institute of Technology.

References

- Byington, C.S., Watson, M., Edwards, D., and Stoelting, P., 2004. A Model-Based Approach to Prognostics and Health Management for Flight Control Actuators, *Proc. IEEE Aerospace Conference*, Big Sky, MN.
- Goebel K., and Eklund, N., 2007, Prognostic Fusion for Uncertainty Reduction, *AIAA Infotech@Aerospace Conference and Exhibit*, 7-10 May, Rohnert Park, California.
- Litt, J.S., Parker, K.J., and Chatterjee, S., 2003, Adaptive Gas Turbine Engine Control for Deterioration Compensation due to Aging, *Technical Report TM 2003-212607*, NASA Glenn, Lewis, OH.
- Monaco J. F., Ward D. G., and Bateman, A. J., 2004, A Retrofit Architecture for Model-Based Adaptive Flight Control, *AIAA 1st Intelligent Systems Technical Conference*, 20 - 22 September, Chicago, Illinois.
- Roemer, M., Tang, L., Kacprzynski G. J., Ge, J., and Vachtsevanos, G., 2006, Simulation-Based Health and Contingency Management, *Proceedings of IEEE Aerospace Conference*, Big Sky, Montana. Mar. 4-11.
- Saxena, A., Orchard, M. E., Zhang, B., Vachtsevanos, G., Tang, L., Lee, Y., and Wardi, Y., 2007, Automated Contingency Management for Propulsion Systems, *European Control Conference*. Kos, Greece 2-5 July.
- Tang, L., Kacprzynski, G. J., Roemer, M. J., Vachtsevanos G., and Patterson-Hine, A., 2005, Automated Contingency Management Design for Advanced Propulsion Systems, *AIAA Infotech@Aerospace Conference and Exhibit*, Arlington, Virginia. 26 - 29 September.
- Tang L., Roemer, M. J., Kacprzynski, and G. J., 2007, Real-Time Health Estimation And Automated Fault Accommodation For Propulsion Systems, GT2007-27862, *Proceedings of ASME Turbo Expo 2007*, Power for Land, Sea and Air, May 14-17, Montreal, Canada.
- Tang, L., Saxena, A., Orchard, M. E., Kacprzynski, G. J., Vachtsevanos, G., and Patterson-Hine, A., 2007, Simulation-based Design and Validation of Automated Contingency Management for Propulsion Systems, *Proceedings of IEEE Aerospace Conference*, Big Sky, Montana. Mar. 3-10.
- Vachtsevanos, G., Lewis, F. L., Roemer, M. J., Hess, A., Wu, B., 2007, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, Wiley.
- Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick III, J., Railsback J., *Fault tree handbook with Aerospace Applications*, version 1.1, August 2002.