Health-Management Driven Control Reconfiguration Approach for Flight Vehicles

Asif Khalak and Kai Goebel

Scientific Monitoring, Inc.
8777 E. Via de Ventura
Scottsdale AZ 85258
asif@scientificmonitoring.com

NASA Ames Research Center
Moffett Field CA 94035
goebel@email.arc.nasa.gov

Abstract

A prognostic system makes it possible to anticipate loss of functionality before it occurs with sufficient lead time to take actions that mitigate the impact of this loss. We focus on the forms of mitigation within the flight vehicle that influence the operational dynamics but do not directly amend the mission plan. Thus, we focus upon the reconfiguration of the feedback control strategy for the flight system.

The high degree of complexity in the design and dynamics of modern aircraft is typically handled using a hierarchical control scheme in which there are several levels of control at increasing levels of responsibility: the component level, the subsystem level, and the system level. Our reconfiguration strategy involves mitigating problems that are detected at the component level at both the level in which the fault is detected and higher levels as well. There are, thus, two subproblems to the reconfiguration: (a) an adaptive control problem at the lower level to extend component life and derive new component performance limits, and (b) a supervisory control problem at the higher level to adapt the system controller to maximize system capability while respecting the performance limitations. Since our reconfiguration occurs in the context of a dynamic system, we need to respect the stability implications of the reconfiguration. To address this, we apply bandwidth analyses at the component level and the systems level in a robust performance context. A conservative criterion for stability is to impose rate limits for reconfiguration that insure that undesired, and possibly unmodeled, modes of behavior are not driven by reconfiguration activities. For specific hardware, extensions beyond this conservative approach may be warranted (e.g. to catch faulty behavior) and validated on a case-by-case basis, essentially by extending the component modeling to include a model of behavior under certain types of reconfiguration.

Introduction

The drive to reduce costs without sacrificing reliability or performance has led to impressive gains in both inherent design robustness and in the more intelligent and aggressive use of information in the health management process. For systems such as the Space Shuttle Orbiter System, months are required to prepare for re-launch, involving extensive retooling and rebuilding of the systems. Launch systems of the future will benefit from having continuous health tracking, and potentially need only a fraction of the current turnaround time since many of the systems will not need to be extensively requalified, instead using the available health information. In aviation, the last two decades have seen significant increases in both the volume and efficiency of civilian air traffic and greater cost pressure in military aviation, requiring greater and greater levels of availability. Current safety, reliability and availability concerns are handled through a combination of spare capacity and periodic inspections. In the push to meet greater demands on air traffic, an even more proactive approach will be useful to anticipate and mitigate problems in flight.

Health management technology has advanced significantly in the last decade through a combination of advances in sensor technology, device modeling, reasoning algorithms, and real-time computational capability. The envisioned needs, however, push the boundaries of what has been achieved in previous applications terms of speed and need for precision. As an approach to address these needs, we describe a methodology for adaptation of the critical and fast-response flight controls system using the prognostic health information. Such a system can exploits the current state-of-the-art in terms of richer health information made possible by the development of advanced sensors and signal processing algorithms.

The current discussion represents a further step towards incorporating health status observers into adapting the flight control of a vehicle. The general idea of controller reconfiguration as a result of health degradation has been studied before (Patton, 1997; Zhang et.al., 2000), and can take the form of something simple (like a shutoff switch to eliminate a noisy sensor), or more complex planning schemes.

Thusfar, overarching schemes for reconfiguration have had substantial barriers to adoption, partly because they require a recasting of the whole controls problem into a reconfigurable framework, which is typically not equipped to handle the reality of modern, complex systems. Two aspects which distinguish this approach are (a) the focus on the use of the health status information in a quantitative manner for integration with prognostic estimators, and (b) the focus on integration into hierarchical control architectures, as found in modern aviation platforms. We feel that these two system aspects are essential characteristics of a realistic system for adapting the dynamic behavior of aircraft to account for incipient health information.

Hierarchical Architectures for Aircraft Systems

Modern aircraft represent one of the most complex machines in use. Due to the safety concerns, these platforms must undergo rigorous design engineering, testing and validation. There can be no single point of failure, due to flightworthiness regulations (FAR Part 23). This and other constraints lead to highly optimized, but constrained designs, and thus are extremely complex. As a means of organizing the systems on the aircraft, a hierarchical scheme is generally employed, which separates responsibilities between the system, subsystem, and component levels.

Figure 1 shows a general architectural concept for integrating reconfiguration into such a hierarchical control scheme. There is a separation between three aspects of the system: (a) the physical system, (b) the reasoning elements, and (c) the flight critical control elements. We distinguish between (b), the health reasoning and reconfiguration elements, and (c), the flight critical control elements for two reasons. First, the synthesis of the flight controller itself is a highly refined topic that is constrained by the safety-criticality of the controller responsibilities. Since we are focused on reconfiguration due to health monitoring, we prefer to discuss how the inputs (i.e. the gain schedule) to the flight controller should be adapted, rather than building a flight controller from scratch. Second, the time-criticality and safety-criticality of the flight control software makes it a highly regulated and constrained aspect of the system. Thus, we separate the health estimation and reconfiguration tasks since they are not as time-critical as the flight software itself.

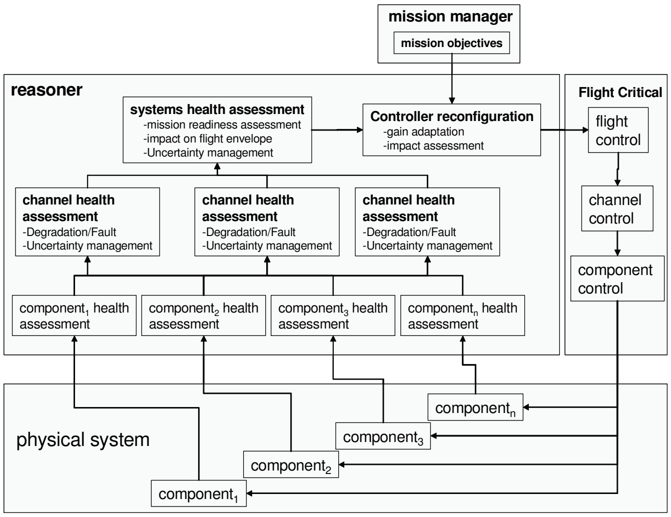

Figure 1 Example of three-layer hierarchical architecture, with component level, channel level, and system levels of control.

At the bottom of the architecture are the components that make up the physical system. These are instrumented with sensors, which are interpreted and calibrated at the lowest level of the reasoner. Within the reasoner, in this architecture, there are three hierarchical levels of reasoning, to align with the structure of the platform itself. Above the component health assessment, there is a channel health assessment which reasons about fault degradation and management of errors/uncertainties in the system. The systems level of health assessment integrates the information from the channels and deduces the overall system capabilities. These considerations enter the reconfiguration block, which can adjust the parametric inputs to various levels of control.

At the head of this entire architecture lies the mission manager, which in many cases may be a pilot or an autopilot.

Reconfiguration Based on Health Assessment

The system architecture in the previous section, is essentially representative of the software architecture necessary for implementation. It is also useful to look at the reconfiguration problem as a dynamic system, and to consider the sources of error in this system that set the constraints on our algorithm design.

A canonical signal flow diagram for a closed-loop dynamic system is shown in Figure 2, with extensions made for a health observer and a controller reconfiguration block. In some ways, this merely recasts the previously shown hierarchical systems architecture by rotating the diagram 90 degrees counter-clockwise, abstracting some blocks and introducing new ones.

This view focuses on the dynamics of the plant (i.e. the physical system) and the errors introduced during operation, which are modeled as disturbances to the output of the plant. The controller attempts to reject these disturbances in the output, besides dealing with the plant dynamics to match the output with a reference signal. A health observer takes information from the output and some internal sensors in the plant and computes an assessment of health.

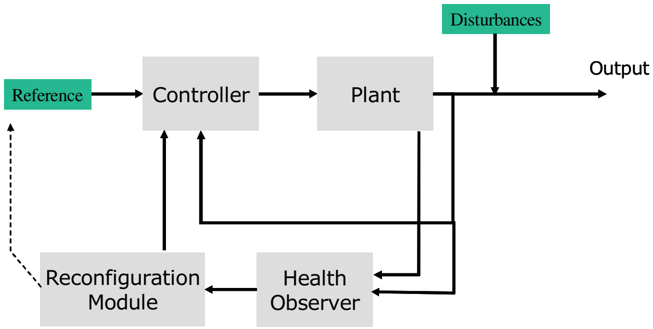

Figure 2 Block diagram of health-assessment based reconfiguration, showing reference and disturbance control signals.

This health assessment is passed to the reconfiguration block. In the case in which there is no active reconfiguration, the minimum action is to pass the health assessment to the user of the platform to assist in manual changes in the reference setting (e.g. helping the pilot make the decision to land the aircraft). We consider the cases in which the reconfiguration block has direct influence on the controller as well. In the next section, we develop some high level guidelines to govern how the reconfiguration block should make changes to the controller.

Analysis of Reconfiguration

The controller has the job of minimizing the output tracking error from the reference (i.e. to reject the inserted disturbances). This controller, is reconfigured, however, to adapt it to changes in the plant health. For example, actuator rate limits which might be acceptable for nominal operation, may be overly damaging in a degraded state.

Thus, the reconfiguration module has a multi-objective optimization problem. That is to adapt the controller (a) to maximize controller performance in rejecting disturbances, and (b) to guarantee safe operation. Moreover, since the system will be operational the whole time, we need to also make sure that the extra dynamics introduced by the reconfiguration module do not adversely affect system performance, and stability in particular. This introduces some constraints on valid and invalid strategies for the reconfiguration module.

We address this question of the reconfiguration stability in a robust control context. Referring to the blocks in Figure 2, we label y as the residual output of the system (as taken from the reference) and we label u as the commanded signal to the plant. In the robustness analysis, we model the uncertainty in the system in terms of two cutoff filters:

- A low pass filter, p, that relates to the bandwidth of performance that we would like to control,
- A high pass filter, r, that relates to the bandwidth of unmodeled dynamics in the system that we do not wish to excite via control action.

If we use simple low pass and high pass filters for p and r, then in the Laplace domain, they have the following forms:

$$p = 1\,/\,(1+as)$$
$$r = s\,/\,(s+b)$$

where a and b are the cutoff time-scales for each filter, respectively.

Based on this model for the system, we can filter the output and command signals, retaining the portions of those signals that have the most pertinent information for our controls synthesis problem:

$$y_f = y * p$$
$$u_f = u * r$$

where the * operator indicates convolution with the filter kernel in the time domain (i.e. application of the filter). Thus, y_f is the portion of the output signal that we wish to control that defines our system performance, and u_f is the portion of the control signal which may excite unmodeled parts of the plant. The criterion for robust performance is defined on the basis of norms on these two signals as follows:

$$\|\, |y_f| + |u_f| \,\|_\infty < 1$$

where the magnitude (in terms of the infinity norm) of the disturbances as depicted in Figure 2 are normalized to 1.

From a reconfiguration perspective, the important term in this criterion is the second term, that with $|u_f|$. The reconfiguration controller must be careful to avoid destabilizing the system by rapid adjustments. We can thus adopt the following principles to conservatively insure robustness of the reconfiguration module.

- Make sure that every change to the controller lies within the set of controllers that is at least as robust as the original controller (i.e. without accounting for the dynamics of the reconfiguration module)
- Make sure that the reconfiguration between these controllers occurs on a timescale that is significantly slower than b, the cutoff timescale of

the *r* filter. That is, within the modeled bandwidth of the system

These criteria determine requirements on the diagnostic/prognostic performance of the health assessment block. That is, we need to capture health issues with sufficient lead time to introduce controller changes gradually, slower than the timescale of the *r* filter associated with the unmodeled dynamics of the system.

Reconfiguration Synthesis

To show an integration of these concepts into a hierarchical framework, we address the example of integrating the reconfiguration into one specific channel controller associated with actuation of the flight control surfaces (e.g. wing and tail flaps, etc.). This particular channel has a tight interaction with the overall flight controller, and thus presents a particularly challenging reconfiguration controls synthesis problem.

A block diagram that shows this integration from a dynamic perspective is shown in Figure 3. Note in relationship to the software/hardware architecture of Figure 1, this diagram shows information flows rather than interfaces. Thus, information that the reconfiguration module sends to the subsystem (i.e. channel) controller may pass through the flight control to be consistent with the hardware/software architecture to simplify the interfaces. Also in this view several modeling components within the controls and reasoning modules are shown. The actuator subsystem is modeled in three places: (1) in a feed-forward form in the flight controller, the ‘actuator model,’ and (2) in a feed-back form in subsystem controller the ‘actuator-based observer, and (3) in a physics-of-failure model in the reasoner.

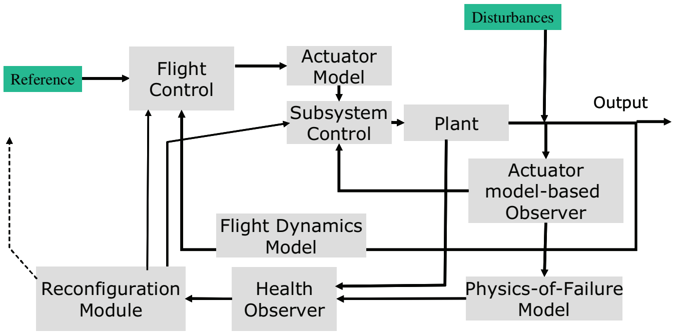

Figure 3 Reconfiguration in the context of a hierarchical architecture with separate model-based controllers for the flight dynamics and the actuator.

Each of these models of the actuator has different inputs and outputs, with the feed-forward actuator mainly concerned with the specifications of the actuator, the feed-back observer concerned with lower-level details that govern dynamic performance errors, and the physics-of-failure model at a slower timescale tracking changes that ultimately affect both the specifications and performance.

Note that this current approach indicates that a change in the actuator may necessitate updates to three separate models, one of which is dedicated for the reconfiguration. Thus, the reconfiguration does add some complexity to the system specification and extra software maintenance to track design changes.

At the supervisory level, reconfiguration of the control scheme proceeds by recomputing parameters (i.e. gains) for the controller in an optimization that uses the performance objective specified in the mission plan and the other objectives, accounting for the actuator degradation. Here, we are using improved knowledge about the actuator as well as specialized knowledge of the mission to adapt the controller to achieve the mission, which is akin to a Model-Predictive approach to reconfiguration.

This general strategy at the supervisory level is limited by the fact that the structure of the controller remains unchanged, and only the parameters are adjusted. At the controller level, specific details are known about the hardware (e.g. the actuators) that warrant more specific reconfiguration strategies involving structural changes to the subsystem controller. While such approaches are reasonable, they must be considered and validated on a base-by-case basis for specific hardware architectures.

Conclusions

In this paper, we have presented an architecture for controller reconfiguration in hierarchical, complex systems, and specifically applied it to flight-control actuators as a part of an overall aircraft system.

The architecture presented has three layers of hierarchy: the systems level, the subsystems (or channel) level, and the component level. The controller application shown utilizes two of these levels, namely the systems and subsystems level.

An analysis of reconfiguration was introduced, on the basis of robust control theory, and it was recognized that insuring that the reconfiguration always switches the control scheme to another stable controller is not sufficient to guarantee the robust performance of the system. Also, one must limit the reconfiguration rate to lie within the controlled bandwidth of the system. This encourages a relatively gradual switching regime.

An example of applying the architecture to a particular aircraft subsystem was conducted, in the form of the flight control actuator. The models for hardware are partitioned in terms of their responsibilities, namely whether they are modeling the idealized performance, the actual closed-loop performance, or the damage evolution of the hardware.

Depending on these, the model will be contained in either the supervisory controller, the subsystem controller, or the health management reasoner.

System architecture is an essential consideration in the application of systems as complex and important as aviation platforms. Practical concerns constrain the manner in which we can modify the hardware and software to achieve availability and reliability goals. Thus, a reconfiguration framework which anticipates the integration into hierarchical systems is a useful first step towards developing a realizable system to exploit the benefits of health management information.

References

FAA FAR Part 23, AIRWORTHINESS STANDARDS: NORMAL, UTILITY, ACROBATIC, AND COMMUTER CATEGORY AIRPLANES.

Patton, R., “Fault Tolerant Control Systems, the 1997 Situation,” IFAC Safeprocess’97 (Hull, United Kingdom), August 1997, pp. 1033–1055.

Zhang H., Ray, A., Phoha, S., “Hybrid life-extending control of mechanical systems: experimental validation of the concept”, Automatica 36 (2000) 23--36

Camacho, E.F., Bordons, C., and Johnson, M., Model Predictive Control. Advanced Textbooks in Control and Signal Processing. Springer Verlag, June 1999.

Doyle, J.C., Francis, B., and Tannenbaum, A., Feedback Control Theory. Macmillan Publishing Company, 1992.