FDI for the IMU component of an AOCS

Maurício Nacib Pontuschka – DCC, PUCSP, Brazil
Ijar Milagre da Fonseca – DEM/INPE and UNINOVE, SP, Brazil
Álvaro A. Neto - CTA, ITA, Brazil

Abstract

Survival of both, manned and unmanned space missions is a critical and decisive issue. The very high costs in research & development, scientific components, and the most important of all, human lives are magnifying the burden on dependability requirements related to software classified as critical or safety-critical software. This work represents the initial phase of a major project being developed in the Space Mechanics and Control Division at the Brazilian National Institute for Space Research (DMC/INPE) involving the complete FDIR subsystem for the AOCS of an hypothetical satellite. For each component it will be identified a set of criteria for fault detection and isolation. Once the criteria set is defined the contingency actions will be defined and implemented in software, aiming the fault recovery. An IMU (Inertial Measurement Unity) comprising four gyroscopes in a tetrahedral configuration is one of the assumed components for the AOCS and this work starts with a FDI for that IMU. The infrastructure necessary to implement the work is the DMC's LABSIM (Laboratório de Simulação Física – Physical Simulation Laboratory). The types of failures considered in this article are, the step abrupt change, ramp/drift/slow, stuck, cyclic, erratic, spike, and finally the stuck for variance alteration noise. It is developed an appropriate algorithm for the automatic detection of each type of fault being analyzed. As a result of this approach it is obtained a mapping of events fault indicators to the IMU. This mapping is very important in the characterization of the occurrence, definition of criteria, and device types and associated fault identification for AOCS.

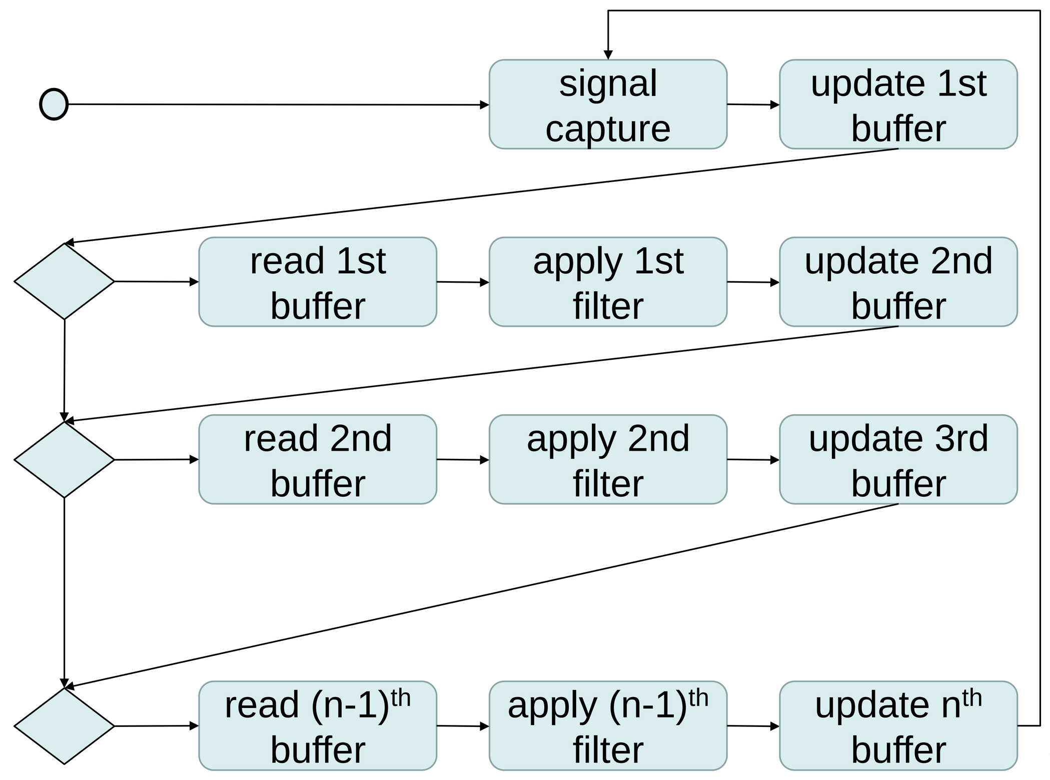
Introduction

The FDIR acronym refers to three main functions related to software (SW) and hardware (HW): Fault Detection, referring to the ability to discover faults or the process of determining that a fault has occurred; Fault Isolation being the function of fault localization within the system by providing information pinpointing it; Fault Recovery referring to the process of limiting the fault propagation and enabling the service to be restored to an acceptable state.
Regarding the On-Board Software (OBSW) FDIR. Some conventional FDIR techniques (For example, SW Watchdogs, Memory Scrubbing, Parameter Monitoring, among others) are typically present in most satellites.
.
More sophisticated software FDIR techniques have recently been developed. One example is the Model-Based FDIR with artificial intelligence (AI) support also called Smart-FDIR. In addition, Advanced FDIR AFDIR) techniques have been made available as reusable components. The AFDIR includes Kalman filtering, Weighted Sum-Squared Residual testing, Generalized Likelihood testing, among others.
In general, any complete FDIR arrangement must be distributed on both HW and software SW levels in any on-board Computer architecture. The FDIR concept at the HW level can be seen for instance in a processor watchdog timer or in an individual watchdog that receives beacon signals from different units. The memory Error Detection and Correction (EDAC) function is also present on the HW level as a classical FDIR component. The HW level is referred as HW FDIR and at SW level as SW FDIR. Considering FDIR analysis and design the following approach can be used:

- Define failure scenarios, detection strategies and levels of autonomy
- Partition FDIR functions into on-board hardware and software or TM/TC actions
- Define FDIR architecture as centralized (single node), distributed FDIR (multiple nodes with majority voting) or a combination of both
- Create an FDIR analysis tool in UML to derive the FDIR actions
- Create an FDIR model of the system incorporating the FDIR algorithm using UML
- Evaluate and optimize the system architecture performance when exposed to different failure scenarios
- Demonstrate the FDIR algorithm running appropriate computer simulations. The MARC (Modular architecture for Robust Computation) demonstrator, handling failures is an option for demonstration.
- Assess the performance and reliability by using appropriate communication protocol (MIL-STD-1553B for example)to handle critical commands/telemetry

This study focuses on the detection and isolation failure (DIF) that can occur in inertial units. These inertial units were originally applied in control of critical systems operations as military combat aircrafts and spacecrafts (Kubat, 1978). Projects of this kind have the characteristic of operating in dynamic environments and demand a huge computational effort to maintain acceptable levels of stability and operability. The stability during the flight requires the use of inertial sensors for acceleration and rotation. The purpose of the associated control devices is to establish reliability, performance, integrity to provide the lowest error rates as possible assuring the mission success.

The algorithms that were created to identify those errors in most cases are based in signal analysis acting as filters that scan the signal searching specific patterns that indicates the presence of known errors. In these cases the computational construction has a loop in which the main structure reads each byte of an input sequence. The problem is how to make several analyses in the same time by an single processor unit. Those algorithms have to be combined in a way that the system could achieve the best level of performance possible.



The strategy is to use a single loop structure and a set of buffers to provide all the analysis in a sequence inside the loop. The sensor provides the raw signal to the DIF that initiates the analysis. The first buffer reads the first byte of the raw signal and it will be enabled to the next level of analysis. In this level the first filter will be used with the information contained in the first buffer. Each filter can have a different quantity of information necessary to begin the analysis, so if the information is not sufficient to initiate the analysis, the flow of execution will go to the next loop where a new value is read from the sensor. When the quantity is sufficient to the first filter, it will process the information contained in the first buffer and provide a new signal to the next buffer. The next buffer will be processed by the next filter in the same way of the first one. In this way it is possible to provide in a same loop all the filter combination necessary to achieve performance in the analysis of the complete set of filters.

Conclusions

In this work it has been considered that all of these filters could apply signal modifications in each buffer in order to process the algorithm and make the complete analysis in the same time by identifying if one of the known errors has occurred. In the case of finding some kind of match with the patterns that were used in the analysis, an exception has to be raised in order to alert that a match has been identified. Is difficult in this time to say that this match is really about an real error, but those matches are fundamental information about on how the system is working and if this behavior is that expected. We know that is practically impossible to predict all kind of errors, but it is sure possible mandatory to incorporate all known errors in a FDIR system. The complete set of those errors and how to identify, isolate and recover each one of them will be used to increase the knowledge base that can be updated each time new information is discovered. It is not a propose of this work to discuss about on how that information could be updated on satellites that are already in operation, but at least new satellites could have the information of earlier missions and could have a safer device to achieve the mission success.

References

Efimov D., Zolghadri A., Raïssi T., **Actuator Fault Detection and Compensation Under Feedback Control**, Elsevier, 2011.
Da Fonseca I.M., Arantes G., Carvalho H., **On The Equars Attitude Dynamics and Control**, 55th International Astronautical Congress - Vancouver, Canada, 2004
Krogmann U., **Failure Management in Spatio-Temporal Redundant, Integrated Navigation and Flight Reference-Systems**, IEEE, Ueberlinger/Germany, 1990.
Leite A., Lopes R., Kuga H., **Model-Based Fault Detection on a Reaction Wheel**, Brazilian Conference on Dynamics, Control and Their Applications, May 22-26, Guaratinguetá, SP, Brazil, 2006.
Oliveira, E., **Detecção e Isolaçãode Falhas em Unidades de Medidas Inerciais com Redundância Mínima de Giros de Fibra Óptica**, Doctored Thesis, INPE, Brazil, 2011.
Omana M., **Robust Fault Detection and Isolation Using a Parity Equation Implementation of Directional Residuals**, Mastes Thesis, Universidad Industrial de Santander, 2001.
Salkham, A., **Fault Detection, Isolation and Recovery (FDIR) in On-Board Software**, Master Thesis, Chalmers University of Technology, Göteborg, 2005.
Stengel R**., Intelligent Failure-Tolerant Control**, IEEE Control Systems, 1991.
Silva, A., **Parameters Identification and Failure Detection Applied to Space Robotic Manipulators**, Doctored Thesis, INPE, 2001.
Teixeira, A., Souza M., Oliva A., **Eigenstructure Techniques for Fault Detection and Isolation in Aerospace and Automotive Systems**, Society of Automotive Engineer, Inc, 2004.
Teixeira, A., Souza M., Oliva A., **Multiples Faults Detection and Isolation in Sensors of Dynamic Systems**, SAE Brazil 2005 Congress Nov 22-24, São Paulo-SP, 2005.