Experiment on signal filter combinations for the analysis of information from inertial measurement units in AOCS

Maurício N Pontuschka¹, Ijar M da Fonseca² and Marco A A Melo³

¹Computer Science Department, PUC-SP, Brazil
²Aeronautical Mechanics Division, ITA, Brazil Prof. Emeritus at EMC/INPE, Brazil
³Engineering Department, PUC-SP, Brazil Researcher LPS, USP-SP, Brazil

E-mail: tuska@pucsp.br

Abstract. The FDIR software subsystem may be part of the attitude and orbit control subsystem, AOCS. The AOCS quite often includes inertial navigation sensors being physically implemented by accelerometers and gyroscopes which provide electrical signals to the AOCS on-board software (OBSW) which, in its turn, generates the commands to the control actuators. In general, hardware like sensors and actuators present nonlinearities which sometimes make it difficult to properly interpret the output signals. In the scenario of space applications, filters are used to eliminate noise and to increase the reliability for the correct interpretation of those signals. In this paper we present a collection of filters used in inertial navigation subsystems enabling the fusion of data from sensors. Fundamentally, the filters are composed of the Kalman filter in its derivations. The filters can be used for state estimation of a system as well as for noise filtering. In this work the filters are configured with respect to their different orders of execution, their sampling rate, and their cutting-off frequency. The filter configurations can be changed by software so as to allow a flexible structure that can be adjusted for the best quality of output signal and consequently the best analysis of the satellite behaviour. The main purpose of this paper is to test the algorithm that combines several signal filters considered in this study. To accomplish this goal we developed an experiment encompassing an accelerometer and a wireless communication system so as to provide input signals to be filtered by the filtering algorithm.

1. Introduction

The scope of this paper is to develop an experiment to test and validate filter algorithms from IMU signals. The main objectives of the experiment are: (1) To test a collection of filters required for inertial navigation systems in an ordered configuration according to their executions; (2) To determine the individual parameter of each filter aiming at real time results; (3) To implement the Kalman filter, low pass filter, high-pass filter, moving average filter, and median filter for signal filtering; and (4) To generate real time graphical results associated with the output obtained from the experiment.

The AOCS and its associated FDIR subsystem are used in navigation control of satellites, rockets and others space vehicles [1] [14]. Those subsystems include inertial navigation sensors such as accelerometers and gyroscopes which provide electrical signals to the AOCS OBSW [9] [10]. Such sensors can be strategically combined to work together to obtain a more precise navigation [7]. In general, sensors present nonlinearities which sometimes make it difficult to properly interpret the signals. Thus, filters are used to eliminate noise and to increase the reliability of the interpretation of satellite sensor signals.
Three families of sensors are highlighted in the FDIR implementation: accelerometers, gyroscopes, and magnetometers. Accelerometers detect proper acceleration acting on the system. Gyroscopes detect the system angular velocity. By using gyroscope sensors it is possible to continuously calculate the orientation of the sensor while it is under acceleration. Magnetometers are not inertial sensors themselves, but they are usually part of a set of devices that are part of the AOCS, so they are under the surveillance of the FDIR software subsystem. Magnetometers give the orientation of the system with respect to the magnetic north. Such sensors provide the direction and the magnitude of the magnetic field vector for specific positions in orbit [8]. The signal analysis initiates with the signal reading, and then the readings are digitalized to be processed by the filters. Then the filtered signal can be processed and subjected to analysis.

The experiment, as shown on Figures 1 and 3, is comprised of equipment containing an accelerometer mounted on a rotating base and a wireless communication system. The experiment is assembled aiming at sensor signal filtering. The accelerometer generates the signal and provides it to the microprocessor. The processed signal is then sent to a computer by Bluetooth technology.

2. Experiment hardware
The processing unit is part of the experiment’s communication system and it is comprised of one Arduino Uno microcontroller. The Arduino Uno is a processing device with input and output data that can be used in laboratory environments for embedded systems. It does not contain large processing capacity, but it is widely used in the academic environment.

The ADXL345 accelerometer provides the information from the three components of the force vector [4]. The sensor provides the digital information that represents the total force acting upon the experiment in the laboratory. The ADXL345 accelerometer is built in such a way that 255 represents 1 G.

The device used for communication between the computer and the experiment is a HC06 Bluetooth port module for Arduino[3]. It allows establishing a serial connection between the Arduino microcontroller and the computer over the Bluetooth protocol1.

![Figure 1. Equipment hardware.](image1)

![Figure 2. Accelerometer y-axis direction.](image2)

![Figure 3. Experiment assembly.](image3)

The equipment was designed with an Arduino processor equipped with an ADXL345 accelerometer [4] and a Bluetooth HC06 [13] assembled on a rotating base. The equipment is restricted to the planar motion as shown in Figure 2. The reading is done only in the y-axis.

3. Filter characteristics
This study addresses the following filters: (1) Kalman filter; (2) high pass filter; (3) low pass filter; (4) moving average filter; and (5) median filter.

---

1 This module can operate in several rates, from 4.800bps to 1.382.400bps and the default value is 9.600bps. In this experiment the default rate of 9.600bps was used.
The Kalman filter was developed by Rudolf E. Kálmán [5]. The Kalman filter (KF) is the optimal Bayesian estimator that provides the global minimum mean square error estimate in a Gaussian and linear system. On the other hand, if the KF is used in a non-linear, non-Gaussian system, it also provides the Linear Minimum Mean Square Estimate (LMMSE), which is the best estimate that can be achieved from a linear filter.

According to Haykin [6], Kalman filter application for state-space formulation of linear dynamical systems provides a recursive solution to the linear optimal filtering problem. It applies to stationary as well as nonstationary environments. The recursive solution at each updated estimate of the state is computed from the previous estimate and the new input data. So only the previous estimate requires storage. Furthermore, in removing the need for storing the entire past observed data, the Kalman filter is computationally more efficient than computing the estimate directly from the entire past observed data at each step of the filtering process.

The low pass filter cuts the high frequency allowing only lower frequencies to pass. As shown in Figure 4, the main parameter of this filter is the $v_1$ parameter. It consists of a number between 0 and 1 stating how much signal is allowed to pass as output signal [2].

$$H_d(v) = \sum_{n=-\infty}^{\infty} C_n e^{j\pi n v}$$

$$C_n = \int_{0}^{1} H_d(v) \cos(n\pi v) \, dv = \frac{\sin(n\pi v_1)}{n\pi}$$

$H_d(v)$: is the response frequency of the magnitude of the digital filter or gain of the digital filter.
$v$ : is the variable digital frequency, defined as the ratio between the analog frequency and the Nyquist rate.
$v_1$ : is the digital cutting frequency.
$C_n$ : is the Fourier coefficients of digital filter.

![Figure 4. Low pass filter equation [2].](image)

The high pass filters the low frequencies allowing only higher frequencies to pass. The main parameter of this filter is also the $v_1$ parameter, which consists of a number between 0 and 1 as a signal informing whether to let pass as output signal [2].

$$H_d(v) = \sum_{n=-\infty}^{\infty} C_n e^{j\pi n v}$$

$$C_n = \int_{0}^{1} H_d(v) \cos(n\pi v) \, dv = \frac{-\sin(n\pi v_1)}{n\pi}$$

$H_d(v)$: is the response frequency of the magnitude of the digital filter or gain of the digital filter.
$v$ : is the variable digital frequency, defined as the ratio between the analog frequency and the Nyquist rate.
$v_1$ : is the digital cutting frequency.
$C_n$ : is the Fourier coefficients of digital filter.

![Figure 5. High pass filter equation [2].](image)

The geometric moving average algorithm can be categorized as low-pass filter. It produces a smoothing of curves representing the signal being processed. It is based on the use of an average signal range [11].

$$Yav(n) = \text{Average}[X(n-K), \ldots, X(n), \ldots, X(n+K)]$$

$Yav(n)$ : is the temporal response of the moving average digital filter.
$X(n)$ : is the temporal input of the moving average digital filter.
$X(n \pm K)$ : is the temporal input of the moving average digital filter in the $n \pm K$ instant.

![Figure 6. Average filter equation.](image)
The median filter is classified as low-pass filter, but unlike the moving average filter, it discards major changes in the signal. The algorithm sorts a signal range and then uses the median value. For this reason the range size must always be odd [11].

\[ Y_{med}(n) = \text{Median}[X(n - K), ..., X(n), ..., X(n + K)] \]

\( Y_{med}(n) \) is the temporal response of the median average digital filter.
\( X(n) \) is the temporal input of the moving average digital filter.
\( X(n \pm K) \) is the temporal input of the moving average digital filter in the \( \pm K \) instant.

**Figure 7.** Median filter equation.

The scheduling process is based on a pipeline approach in which the loop of each algorithm is shared by all filter implementations. The scheduler controller is responsible for applying the filters using the correct order and parameters. This order can be configured without the need to recompile the software code.

![Pipeline diagram](image)

**Figure 8.** Basic scheme of the schedule approach.

The scheme illustrated by Figure 8 shows that each process has its own buffer and filter. It also shows that the output of a filter is connected to the next filter input. Each buffer is independent for each process; the data stored in a buffer has to be used only by the associated process [12].

In this experiment two software were developed: (1) Embedded software for the operation of the accelerometer and to transmit the signal to the computer; (2) Analysis software for the capture of transmission made by the experiment. It is also responsible for filtering the captured signal and providing a graphical visualization in real time. The embedded software was developed in C language with libraries for the Arduino environment. In general, the embedded software performs the initialization of the serial baud rate and establishes the connection. Then, the software loops which perform the reading of the "y" accelerometer axis and send the value of the reading to the serial port connected to the Bluetooth device. The analysis software was developed in Java language by using the Eclipse platform. The captured signal is analyzed on the computer where it is possible to perform the filter application. The system allows choosing which filter will be used and also the filter application order. A filter can be applied more than once if necessary. The pipeline approach was used while applying filters, allowing multiple filters to be applied in the same real-time analysis.

**4. Results**

In this work the software that was implemented, in addition to carrying out the filtering process, also shows the original signal (optional) and the processed signal graphically. The analysis can be performed in real time by viewing a representation of the forces that are sensed by the accelerometer.

The analysis software can show the signal that is processed by the combination of filters and, at the same time, compare that signal with the raw signal. Figures 9 to 14 present several tests of real signal readings from the device that was assembled.
As can be seen on the highlighted area, the signal wave (blue line) has been smoothed with respect to the noise that appears on the raw signal. The low pass is a category of filter that comprises an algorithm to filter the noise in a signal wave. As can be seen in Figure 10, the result is nearly the same as that for the Kalman filter.

The high pass filter is a different kind of filter. In Figure 11 we see the high pass filter which filters the signal low frequency. Only the high frequencies can pass through that filter. The median filter illustrated in Figure 12 ignores abrupt changes caused by static electricity or other kinds of noise. In Figure 12 it is possible to see an abrupt change that is completely ignored by the median filter.

2 The Kalman filter parameters used in the experiment were: process noise covariance $Q=0.022$ and measurement noise covariance $R=0.617$.

3 The low pass filter parameters used in the experiment were: $v_1=0.02$ and window size=7 bytes.

4 The high pass filter parameters used in the experiment were: $v_1=0.98$ and window size=7 bytes.

5 The median filter parameters used in the experiment were: window size=7 bytes.
In Figure 13 we used the Kalman filter in a signal associated with an impulsive noise application. It is possible to see that the filter did not ignore the impulsive noise. Instead, it provides a smoothed wave after the noise occurrence until the wave achieves the normal signal again.

In the example shown in Figure 14, two filters were combined. The median filter is applied in first place to remove the impulsive noise, and after that the Kalman filter is applied. In this case, we can see that the impulse was removed. For this case a small latency is observed between the raw and the filtered signal. The latency is associated with the use of two filters.

5. Conclusion
In this work several filter mathematical models were presented to be used in signal processing. An experiment containing an accelerometer was built to generate and process signals so as to test those filters. The experimental approach considered the combination of different filters execution order according to their applications. The results show that the combined approach can be an option for signal analysis.

References
[1] Bartys M, Patton R, Syfert M, Heras S de las, and Quevedo J 2006 Introduction to the DAMADICS actuator FDI benchmark study, (Control Engineering Practice vol 14) no. 6 (Elsevier) pp 577-596
[2] Chassaing R 1998 Digital Signal Processing: Laboratory Experiments Using C and the TMS320C31 DSK (Wiley)
[3] CuteDigi BMX Bluetooth to UART/I2C/USB Module (GEN II) Datasheet
[4] 2009-2013 Digital Accelerometer ADXL345 Data Sheet (Analog Devices Inc.)
[5] Kalman R E 1960 A New Approach to Linear Filtering and prediction problems (Journal of Basic Engineering 82) (ASME) pp 35-45
[6] Haykin S 2001 Kalman Filtering and Neural Networks (Wiley)
[7] Iserman R 2005 Model-based fault-detection and diagnosis – status and applications (Annual Reviews in Control vol. 29) no 1 pp 71-85
[8] King A D 1998 Inertial Navigation – forty years of evolution (Gec Review vol 13) no 3
[9] Olive X 2012 FDI(R) for satellites: how to deal with high availability and robustness in the space domain? (International Journal of Applied Mathematics and Computer Science vol 22) no 1 pp 99-107
[10] Patton R J, Uppal F J, Simani S, and Polle B 2010 Robust FDI applied to thruster faults of a satellite system (Control Engineering Practice vol 18) no 9 pp 1093-1109
[11] Pitas I, Venetsanopoulos A N 1992 Order Statistics in Digital Image Processing (Proceedings of the IEEE vol 80) no 12
[12] Pontuschka M, Fonseca I et al FDIR for the IMU Component of AOCS Systems (Mathematical Problems in Engineering vol 2014)
[13] Scarfone K, Padgette 2008 J, Guide to Bluetooth Security – Recommendations of the National Institute of the Standards and Technology (Computer Security Division Information Technology Laboratory, NIST) (Gaithersburg) pp 20899-8930
[14] Zhang Y and Jiang J 2008 Bibliographical review on reconfigurable fault-tolerant control systems (Annual Reviews in Control vol 32) no 2 pp 229-252