REAL-TIME ATTITUDE AND ORBIT CONTROL SYSTEM OF A SMALL LEO SATELLITE WITH PARALLEL-PROCESSING APPROACH IN A GROUND STATION

Dr. Roger W. Johnson

Associate Professor, MMAE Department, University of Central Florida, Orlando, Florida 32816

407-823-2155, rjohnson@pegasus.cc.ucf.edu

Sanjay Jayaram

Graduate Student, MMAE Department, University of Central Florida, Orlando, Florida 32816

407-823-6696, sj33050@pegasus.cc.ucf.edu

Lei Sun

Graduate Student, MMAE Department, University of Central Florida, Orlando, Florida 32816

407-823-6696, ls00722@pegasus.cc.ucf.edu

Dr. Chan Ho Ham

Assistant Professor, FSI/MMAE Department, University of Central Florida, Orlando, Florida 32816

407-658-5598, cham@pegasus.cc.ucf.edu

1.0 Abstract

A Real-time Attitude and Orbit Control System (AOCS) using a parallel processing approach will be implemented using a Satellite Ground Control Station (SGCS). With this approach, the satellite will collect platform (attitude reference) and payload measurements (at 1Hz) over several orbits and down-link to the SGCS (at 5 KHz) where the real-time filtering, fault detection, diagnosis and corrective (if needed) control is generated. The control commands are up-linked to the satellite before losing telemetry lock. A substantial reduction in on-board computation can provide a cheaper and a higher reliable AOCS. In the proposed scheme, an advanced real-time estimation and control algorithm in the SGCS for monitoring the critical parameters of the system will be described for a gravity gradient stabilized Low Earth Orbit (LEO) satellite. The parallel (or distributed) processing technique effectively reduces the computation time and increases the subsystem performance dramatically where computation time becomes a critical factor due to the low earth orbit operation. The proposed scheme is based on (1) a Kalman Filter (KF) to perform the real-time estimation of the critical parameters, (2) a parallel processing technique to reduce the computation time, and (3) their implementation with an advanced control algorithm to reconfigure the complete AOCS from the SGCS (if necessary).

2.0 Introduction

The ultimate goal in achieving Autonomous Health Monitoring (AHM) for remote vehicle operation is the identification/demonstration of critical technology innovations that will be applied to the Autonomous Health Monitoring and Data Validation System (AHMDVS). The AHMDVS will provide the real-time signal detection, fault mode diagnostic identification, control and system parameter correction (if necessary) for the Spacecraft to autonomously operate in the space environment for long periods of time (years) without frequent telecommunications from a ground station. This new innovation, using a High Fidelity, Dynamic Model-based Simulation (HFDMS) approach will be advanced to implement a real-time monitoring and control methodology for the AHMDVS.

The unique element of this process control technique is the use of high fidelity, computer generated dynamic models to replicate the behavior of the actual systems. It will provide a dynamic simulation capability that becomes the reference truth model, from which comparisons are made with the actual raw/conditioned data from measurement elements (i.e. Sensors and Actuators). The insertion of this new concept of AHMDVS into spacecraft monitoring and control systems will provide a *real-time* intelligent, command and control system that has the

capability to monitor and observe transient behavior, along with the dynamic parameters of the systems being tested. Current capabilities cannot measure the dynamic behavior of the system in real-time. Abnormal dynamic parameters are indicators of an out-of-tolerance performance of the system; they can be a predictor of impending failures in those systems. This feature adds a new dimension to existing control mechanizations that will greatly enhance the visibility of the "system state" which, in turn, increases the reliability of the test and evaluation process and autonomous operations over those currently in use. This processing technique also promises the real-time detection of abnormal data flow conditions and the automatic identification of the specific area (component/subsystem) causing the fault condition. This attribute speeds up diagnostic analysis to near real-time and provide enough time to stabilize the system by parameter correction.

As a first step to this goal, a Attitude and Orbit Control System (AOCS) will be analyzed using small low cost satellites in conjunction with a Satellite Ground Control Station (SGCS) currently under development by the Florida Space Institute (FSI). FSI is a consortium of Universities in Central Florida under the leadership of the University of Central Florida (UCF). It is therefore important to design the AOCS using emerging and effective technologies that provide not only better performance with cost reduction but also higher reliability by reducing complexity of satellite. For small low-cost satellites, on-board large memory and high-speed processor are not usually available and cheap and reliable attitude components such as a magnetometer are also desirable. On the other hand, high-speed computation is often required to enhance the accuracy of the attitude knowledge and to derive yaw attitude when a magnetometer is used. So, "Faster, Better, Cheaper" concept can be achieved either by having greater autonomy in satellite or by transferring heavy computational burden to ground station.

As shown in previous results, the Kalman filter algorithm is computationally intensive, therefore a parallel processing technique will be vital to meet the timing performance requirement. Most existing parallel implementations are based on systolic array and hence require special purpose hardware with many processors linked by an internal network, designed for high speed digital signal processing; however, this approach will be software-alternative solution. In this approach, the equation of KF is split into two parallel groups, state and covariance, connected by Kalman gain and the measurement matrix. Therefore, the computation task for KF can be split into two sets: state and error covariance equations. This is a straightforward parallelization method using two processors, one doing state update and the other doing error covariance update. However, the computations of the covariance equation are approximately three times more computationally demanding than state equations. So, an algorithm of task mapping for balanced parallel computation is developed to reduce the processing time significantly for the covariance equations. For the proper partitioning method, we distribute the computation of multiple tasks into various sub-matrices and let them execute in parallel. An ideal task distribution and mapping technique should agree with the criteria of maintaining balance between the multiple tasks and in each iteration, the inter-task communication should as low as possible. For implementation of the proposed scheme, the VME bus multi-processor computing system will be used. It will be shown that the proposed scheme with the VME bus is very effective to provide an attitude and orbit control system (AOCS) for a small LEO satellite.

3.0 Background

The Florida Space Institute (FSI) is developing a Satellite Ground Control Station (SGCS) that will be linked to virtually every university and college in the US via Internet II. FSI has also established a Small Satellite Fabrication and Test Laboratory at the Cape Canaveral Air Force Station (CCAFS) for building and integrating nano-satellites. It has also identified launch opportunities for small piggyback satellites among the NASA/KSC and industrial launch vehicle operators.

FSI plans to support University/Industry space experimentors by receiving their payloads, rapping a satellite around it, finding a launch opportunity

and supporting the command, control and data retrieval of the payload through Internet II. The SGCS will be student operated and under the supervision of FSI. SGCS will act as the "communication node" for the payload experimentors; through Internet II, down-linked conditioned data will be routed to the experimentor and command and control input from the experimentors will be up-linked to their payload.

4.0 Objectives

The objectives in describing the scope of this paper is to identify and demonstrate critical innovations that can be applied for the autonomous process with increased reliability and parameter correction to support autonomous operations. Using a High Fidelity, Dynamic Model-based (HFDM) simulation, a real-time health monitoring and data validation system will be described for detecting an "abnormal signal flow" in the sensors and actuator sub-systems by understanding and utilizing the science and platform measurement data collected. As the result, this methodology can achieve rapid identification of non-expected data flow and/or "fault isolation capability". The associated control (correction factor) for this fault will be up-linked to correct the anomaly (if detected).

5.0 Description of AOCS Methodology

The Attitude and Orbit Control System (AOCS) mechanization assumes that the small satellite has a lightweight magnetometer attitude control platform with remote sensing instrumentation (payload). This sensor obtains earth's surface measurements that are correlated to sensor pointing data. This requirement makes it necessary to determine accurate sensor pointing data for each measurement. Therefore the AOCS needs positional and attitude data to not only control (attitude stability) but also for the augmentation of payload sensor measurements.

Assuming three sensor measurements for each attitude "state", the total number of "state" measurements is 18 Bytes (8 bits per byte). If platform data is collected at 1 Hz (1.8 B/sec), one

orbit (5400 sec) of platform data will total 10 KB. But one ground station may be out of sight over four or five orbits, therefore, storage of platform data may be as high as 50 KB.

An active pass of a LEO satellite to a fixed location on the ground lasts approximately 90 sec. We assume the platform data can be down-linked at 5 KHz, therefore the 50 KB would be transmitted in 10 sec at a rate of 5 KB/sec. At the ground station, the raw data is filtered using a Kalman Filter (KF) with selective sampling down to 3.6 msec/18 B of raw data.

The KF or Extended Kalman Filter (EKF) will perform a time accelerated estimation of critical parameters using a nominal (reference) model and minimizing the error (Root Mean Square) between the "best estimate" and the model reference. A dynamic thresholding detection system will detect abnormal transient or steady-state error profiles. Comparisons to a fault-mode-file (pre-generated) will pick up the controller correction sequence to up-link the satellite before it is out of sight. Attitude platform errors detected will also be available to update the payload remote sensor position information thereby validating the down-linked scientific data back to the users. The next section briefly tells about the HFDM methodology. Some illustrative plots for the dynamic thresholding detection system is shown in the results section.

5.1 HFDM Methodology

The new innovation using a High Fidelity, Model-based Simulation (HFDMS) approach will be used to implement a real time monitoring, detection/diagnostic methodology. The unique element of this process control technique is the use of high fidelity, computer generated dynamic models to replicate the behavior of dynamic vehicle parameters by monitoring the system states obtained from telemetry data. It will provide a dynamic simulation capability that becomes the reference truth model, from which comparisons are made with the actual raw/unconditioned data from the sensors on-board the satellites.

6.0 Dynamic Models

For any spacecraft vehicle, there are two types of parameters that are involved in the controller design for a navigation system design, one is a 6 DOF position vector where the vehicle is assumed to be a point mass, and we consider the position of the point at any given time. The other type is the attitude vector, in this case, the shape of the vehicle is considered and the angles between the vehicle body and the reference frame are the control variables. Non-linearity exists in both cases. However, a linear model is used for the attitude controller design. The following paragraph is a presentation of the linear attitude model and the non-linear navigation model. System response with random noise is also discussed as a preparation for Kalman filter application.

6.1 Spacecraft Attitude Dynamic Model

Here we consider a system of coordinates that maintain their orientation relative to the Earth as the spacecraft moves in orbit. These coordinates are known as *roll, pitch* and *yaw* or RPY. The spacecraft considered is a gravity gradient stabilization one in a circular orbit. The linearized

$$I_x\ddot{\boldsymbol{f}} + 4w_0^2(I_y - I_z)\boldsymbol{f} - (I_x - I_y + I_z)w_0\dot{\boldsymbol{y}} - T_{dx} = \boldsymbol{t}_{cx}$$

$$I_y\ddot{\boldsymbol{q}} + 3w_0^2(I_x - I_z)\boldsymbol{q} - T_{dy} = \boldsymbol{t}_{cy} \qquad (1)$$

$$I_z\ddot{\boldsymbol{y}} + w_0^2(I_y - I_x)\boldsymbol{y} + (I_x - I_y + I_z)w_0\dot{\boldsymbol{f}} - T_{dz} = \boldsymbol{t}_{cz}$$

equations of motion are given by [5]:

where, the parameters defined are

I_x : Moment of Inertia in Roll axis
I_y : Moment of Inertia in Pitch axis
I_z : Moment of Inertia in Yaw axis
$\boldsymbol{w_o}$: Orbital rate
T_{dx}, T_{dy}, T_{dz} : Disturbance torque in roll, pitch and yaw directions.
$\boldsymbol{t}_{cx}, \boldsymbol{t}_{cy}, \boldsymbol{t}_{cz}$: Control torque in roll, pitch and yaw axes.
$\boldsymbol{f}, \boldsymbol{q}, \boldsymbol{y}$: Roll, Pitch and Yaw attitude angles

In state space representation, the dynamic equations of motion are shown below:

$$\dot{x}_1 = x_4$$
$$\dot{x}_2 = x_5$$
$$\dot{x}_3 = x_6$$
$$\dot{x}_4 = -\frac{4w_0^2}{I_x}(I_y - I_z)x_1 + (1 - \frac{I_y}{I_x} + \frac{I_z}{I_x})w_0 x_6 + \frac{u_x}{I_x} \qquad (2)$$
$$\dot{x}_5 = -\frac{3w_0^2}{I_y}(I_x - I_z)x_2 + \frac{u_y}{I_y}$$
$$\dot{x}_6 = -\frac{w_0^2}{I_z}(I_y - I_x)x_3 + (\frac{I_x}{I_z} - \frac{I_y}{I_z} + 1)w_0 x_4 + \frac{u_z}{I_z}$$

where the state vector is represented as,

$$\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{q} \\ \boldsymbol{y} \\ \dot{\boldsymbol{f}} \\ \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{y}} \end{Bmatrix} \qquad (3)$$

6.2 Orbital Parameter Dynamic Model

The satellite orbital measurements (from telemetry range and range rate data) are used so that a Kalman filter can be applied to compute the "state estimation" of the orbital elements. Certain assumptions are made to simplify the problem so that the fundamentals are highlighted. These assumptions are as follows:

- The earth is spherical.
- The earth is non-rotating

The first assumption's effect is the earth's oblateness. This effect is relatively small in free flight trajectories. The second assumption merely fixes a target on the earth in inertial space so that the computation doesn't have to be modified to compensate the movement of the target during the flight of the vehicle.

The mathematical model representing the motion of the vehicle's center of mass is described by the following dynamic equations.

$$\dot{r} = \frac{w}{r}$$

$$\dot{v} = -\frac{m}{vr^2} + (\mathbf{1}_v \cdot \mathbf{p}) + (\mathbf{1}_v \cdot \mathbf{u})$$

$$\dot{w} = v^2 - \frac{m}{r} + (\mathbf{r} \cdot \mathbf{p}) + (\mathbf{r} \cdot \mathbf{u}) \qquad (4)$$

$$\dot{s} = \frac{1}{r}[v^2 - (\frac{w}{r})^2 - (\frac{H}{r})^2]^{1/2}$$

$$\dot{\mathbf{H}} = \mathbf{r} \times (\mathbf{p} + \mathbf{u})$$

$$\dot{L} = \frac{\cot(i)\sin(L) \cdot r \cdot \Delta v}{r^2}$$

where the last equation gives the information of latitude at any time in the orbit.

The parameters defined are

r	: Position at any time in the orbit
v	: Velocity at any time in the orbit
w	: Component of velocity
m	: Product of Gravitational constant and mass
s	: Range angle
H	: Angular momentum
L	: Latitude at any time in the orbit
i	: Angle of incidence.
p	: Perturbation vector
u	: Control vector
1_v	: Unit vector in velocity direction.

7.0 Real-Time & Parallel Processing Configuration

As has been noted from previous results, the Kalman filter algorithms are computationally intensive, therefore, parallel processing will be used to meet the timing performance requirement. Most existing parallel implementations are based on either systolic array or transputer [7][8]. Both of them require special purpose hardware with many processors linked by internal network, designed for high speed DSP. For implementation of a parallel program on a bus based computing system or distributed computing system, one should carefully select the algorithm following a certain guideline, namely, the most important

factor is to maintain the balance after partitioning the computing tasks.

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{1,3\times3} & \mathbf{0} \end{bmatrix}$$

8.0 KF partitioning strategies

The equation of KF can be split into two parallel groups, state and covariance, connected by Kalman gain $\mathbf{K}_k$ and the measurement matrix $\mathbf{H}_k$. Therefore, the computation task for KF can be split into two sets: state and error covariance equations. This is a quite straightforward parallelization method, by taking two processors, one doing state update and the other doing error covariance update. However, the computations of the covariance equation are approximately three times more computationally demanding than state equations. Therefore, one should look for a parallel implementation of an algorithm that significantly reduces the processing time for the covariance equations.

In our KF applications, the covariance matrix (6×6) and the measurement matrix (3×6) can be partitioned into (3×3) sub-matrices which allow the partitioning of the Kalman gain matrix into (3×3) sub-matrices, each of which can be calculated independently. Let's define P to represent the predicted covariance P_k^- and M to represent the filtered covariance P_k, and drop all the time indices, then P, M and Kalman gain K are

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{21}^T \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{21}^T \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_2 \\ \mathbf{K}_1 \end{bmatrix}$$

partitioned as

As $\mathbf{P}$ and $\mathbf{M}$ are symmetric, the sub-matrices $\mathbf{P}_{11}$, $\mathbf{P}_{22}$, $\mathbf{M}_{11}$ and $\mathbf{M}_{22}$ are also symmetric. Since we are taking 3 variables as measurements, the measurement matrix can be partitioned as

Then in the Kalman gain equation, the matrix product becomes

$$\mathbf{P}_k^- \mathbf{H}_k^T = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{21}^T \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{H}_1^T \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}\mathbf{H}_1^T \\ \mathbf{P}_{21}\mathbf{H}_1^T \end{bmatrix}$$

$$\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11}\mathbf{H}_1^T \\ \mathbf{P}_{21}\mathbf{H}_1^T \end{bmatrix} = \mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T \qquad (5)$$

Substituting these two expressions into Kalman gain equation, we have

$$\mathbf{K}_k = \begin{bmatrix} \mathbf{P}_{11}\mathbf{H}_1^T \\ \mathbf{P}_{21}\mathbf{H}_1^T \end{bmatrix}(\mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{R}_k)^{-1} = \begin{bmatrix} \mathbf{P}_{11}\mathbf{H}_1^T(\mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{R}_k)^{-1} \\ \mathbf{P}_{21}\mathbf{H}_1^T(\mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{R}_k)^{-1} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{P}_{11}(\mathbf{P}_{11}+\mathbf{H}_1^{-1}\mathbf{R}_k\mathbf{H}_1^{T})^{-1}\mathbf{H}_1^{-1} \\ \mathbf{P}_{21}(\mathbf{P}_{11}+\mathbf{H}_1^{-1}\mathbf{R}_k\mathbf{H}_1^{T})^{-1}\mathbf{H}_1^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix} \qquad (6)$$

Note that we have broken the computation of the Kalman gain into two parts, each part computes the (3×3) sub-matrices of $\mathbf{K}_k$, and each part can be executed independently. Similarly, we can expand the covariance update equation into

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{21}^T \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix}\begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{21}^T \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \qquad (7)$$
$$= \begin{bmatrix} \mathbf{P}_{11}-\mathbf{K}_1\mathbf{H}_1\mathbf{P}_{11} & \mathbf{P}_{21}^T-\mathbf{K}_1\mathbf{H}_1\mathbf{P}_{21}^T \\ \mathbf{P}_{21}-\mathbf{K}_2\mathbf{H}_1\mathbf{P}_{11} & \mathbf{P}_{22}-\mathbf{K}_2\mathbf{H}_1\mathbf{P}_{21}^T \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{21}^T \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}$$

To prove the symmetry of the above equation, note that $\mathbf{P}_{11}$, $\mathbf{R}_k$ are symmetric. The time-update covariance equation can also be split as:

$$\begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{21}^T \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} = \begin{bmatrix} \ddot{o}_{11} & \ddot{o}_{12} \\ \ddot{o}_{21} & \ddot{o}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{21}^T \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}\begin{bmatrix} \ddot{o}_{11}^T & \ddot{o}_{21}^T \\ \ddot{o}_{12}^T & \ddot{o}_{22}^T \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} \qquad (8)$$

It follows that

$$\mathbf{P}_{11} = \ddot{o}_{11}\mathbf{M}_1\ddot{o}_{11}^T + \ddot{o}_{12}\mathbf{M}_2\ddot{o}_{11}^T + \ddot{o}_{11}\mathbf{M}_2^T\ddot{o}_{12}^T + \ddot{o}_{12}\mathbf{M}_{22}\ddot{o}_{12}^T + \mathbf{Q}_1$$
$$\mathbf{P}_{21} = \ddot{o}_{21}\mathbf{M}_1\ddot{o}_{11}^T + \ddot{o}_{22}\mathbf{M}_2\ddot{o}_{11}^T + \ddot{o}_{21}\mathbf{M}_2^T\ddot{o}_{12}^T + \ddot{o}_{22}\mathbf{M}_{22}\ddot{o}_{12}^T + \mathbf{Q}_{21} \qquad (9)$$
$$\mathbf{P}_{22} = \ddot{o}_{21}\mathbf{M}_1\ddot{o}_{21}^T + \ddot{o}_{22}\mathbf{M}_2\ddot{o}_{21}^T + \ddot{o}_{21}\mathbf{M}_2^T\ddot{o}_{22}^T + \ddot{o}_{22}\mathbf{M}_{22}\ddot{o}_{22}^T + \mathbf{Q}_{22}$$

8.1 KF Task Mapping Strategies

Based on the above partitioning method, we can map the computation of the sub-matrices into multiple tasks and let them execute in parallel. An ideal task mapping technique should agree with the following criteria

(1) To maintain balance between the multiple tasks
(2) In every iteration, the inter-task communication should be as low as possible.

Due to the symmetry of matrices $\mathbf{P}$ and $\mathbf{M}$, both of them have 3 sub-matrices to be computed. Hence, 3 processors (tasks) can be applied to carry out this computation. However, in computing the $\mathbf{P}$ matrix all of the 3 sub-matrices of $\mathbf{M}$ are needed, which increase the inter-task communication time. The partitioning of the Kalman gain gives us a hint that we can take 2 processors, one to compute $\mathbf{K}_1$ and the other to compute $\mathbf{K}_2$. Based on this idea and the study of partitioned equation, we can distribute the computation as follows:

Task1: Compute $\mathbf{K}_1, \mathbf{M}_{11}$ and $\mathbf{P}_{11}$ and compute the state estimation and prediction
Task2: Compute $\mathbf{K}_2, \mathbf{M}_{21}, \mathbf{M}_{22}$ and $\mathbf{P}_{21}, \mathbf{P}_{22}$

9.0 Hardware Implementation

All of the applications discussed previously have been implemented in ANSI C language, on a VME bus multi-processor computing system. The computations include KF simulation on a single processor and KF simulation on multiple processors and SKF simulation on a single processor. The next section describes the hardware used in this research.

In this research, a *VME bus multiprocessor computing system* was used for parallel processing. VME bus is a flexible open-ended bus system that makes use of the Eurocard standard. VME bus was intended to be a flexible environment supporting a variety of computing tasks, and has become a very popular standard in the computer industry. The bus was developed from a computing point of view, to allow processors' access to a fully memory mapped scheme. Every device on the bus can be viewed as an address, or block of addresses. Under VME, addresses and data are not multiplexed. A block transfer however, is also possible for DMA-style applications. The bus allows multiple masters, and contains a powerful interrupt scheme. A resource manager is required to handle the interrupts. The VME bus transfers are asynchronous, and allow the data transfer speed up to approximately 20 Mbytes per second. A typical transfer consists of an arbitration cycle (to gain bus control), an

address cycle (to select the register) and the actual data cycle. Read, write, modify and block transfers are supported. Three MVME167 single board computers are plugged into the VME bus system. Each of them contains a MC68040 processor and an Ethernet interface that supports TCP/IP, thus making it suitable to transfer the data through Internet. Figure 1 illustrates the outline of this multiprocessor computing system.

In the configuration shown, processor 0 is the "Master Processor" and its onboard memory (dual port) is configured as "shared memory". Other processor can read or write this shared memory region through the bus, at the same time. Each processor has local memory and is able to access its own memory without interacting with other processors or the shared memory.
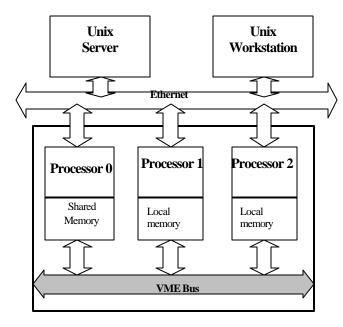
Figure (1): Outline of Multi-processor Computer System

10.0 Simulation Results

The spacecraft we consider is a gravity gradient stabilized spacecraft with a circular orbit. The attitude physical parameters and initial conditions are given in Table 1.

Table 1: Physical parameters and initial conditions

Moment of Inertia	$I_x = 120$ Kg-m^2	$I_y = 120$ Kg-m^2	$I_z = 1.2$ Kg-m^2
Initial Angle	$\theta = 0.1$ rad	$\phi = 0.1$ rad	$\psi = 0.1$ rad

Since actual sensor data was not available, an artificial data stream was synthesized by integrating the state equations (1) and (2) to obtain the state variable value at each sampling period and superimposing a random number with a known standard deviation. Some of the example plots (Figures (2), (3) and (4)) illustrate the simulation results for spacecraft attitude (linear model) estimation, the Sigma (measurement noise) is assumed to be 0.5.
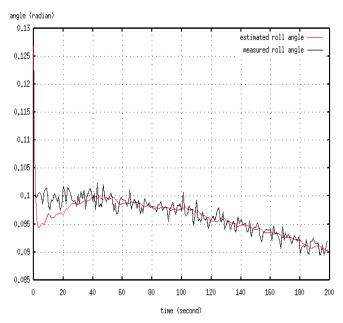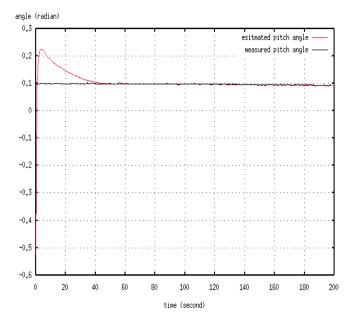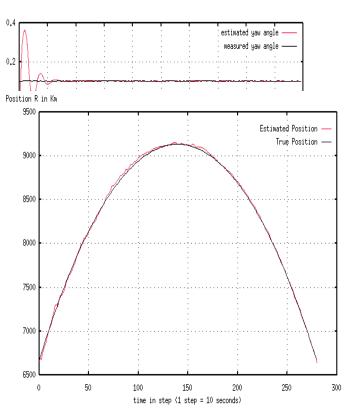
Figure (2): Estimated & Measured Roll Angle vs Time

Figure (5): Estimated Position and Reference (True) Position vs. Time

Figure (3): Estimated &Measured Pitch Angle vs Time

Figure (4): Estimated & Measured Yaw Angle vs. Time

For Orbital parameter dynamic model, the initial conditions for simulation are shown in Table 2. Figures (5), (6), (7) and (8) illustrate the results for orbital parameter (non-linear model) estimation.

Table 2: Initial Conditions for Orbital Dynamics model

Symbol	Name	Value
r	Position	6678.1363 Km
v	Velocity	7.32 Km/s
w	Velocity component	24441.98 Km/s
σ	Range angle	2.75 radian
H	Angular Momentum	42334.75 Km^2/s
i	Inclination angle	0.4974 radian
ω_o	Orbital period	0.00114 radian
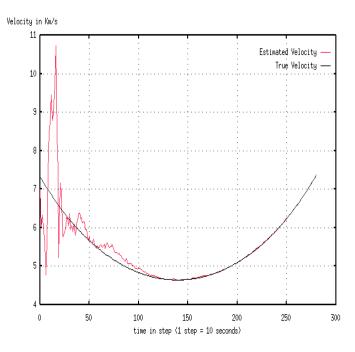
Figure (6): Estimated Velocity and Reference (True) Velocity vs. Time

Rangel Angle in Radian
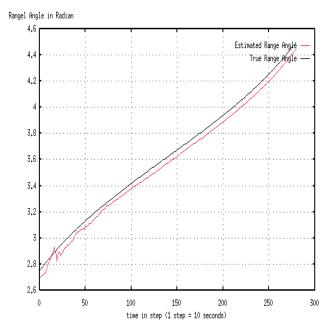
Figure (7): Estimated Velocity Component and Reference (True) Velocity Component vs. Time
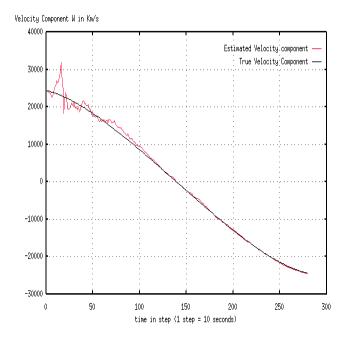
Velocity Component W in Km/s

Figure (8): Estimated Range Angle and Reference (True) Range Angle vs. Time

Some illustrative plots to show the dynamic thresholding detection system. Figure (9) shows the results of simulation of the nominal system with no error detection, and the dynamic threshold nicely brackets the Mean Square Error (MSE) computation during the transient periods. Figure (10) is included to show what happens if the dynamic threshold value is not filtered to match the MSE. This figure demonstrates the time lag due to filtering the MSE, can cause a false detect if it is not compensated for in computation of the dynamic threshold.
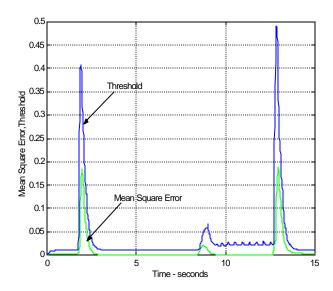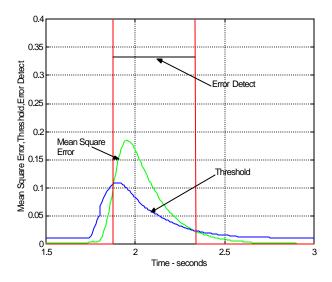

Figure (9): Nominal system – No abnormalities.

Figure (10): Abnormality detected when Mean Square Error (MSE) signal is outside of threshold.

11.0 Control Implementation

The fault-mode files are generated by using the high-fidelity models and perturbing the parameters of the differential equations by a small value, thus generating a unique error response profile for each "state" due to that perturbation. Therefore, when the fault detection system identifies an abnormality in a "state" profile, the error profile is compared to the fault-mode file. When a match is made in a fault-mode, the associated controller for that fault-mode is used to correct the parameter.

12.0 Conclusions

A linear standard Kalman filter is used to demonstrate the best estimate for the space vehicle attitude system and to estimate the vehicle position information. The corresponding real-time simulation is performed based on the available hardware and software. Test results show that based on the dynamic models used and implemented, the standard Kalman filter with implementation of parallel processing will give good estimation of the system state variables and the simulated measurement noise is eliminated successfully. The parallelization method discussed here is based on the number of measurements used in the state equations, which are applicable to our particular application. The more general parallelization method would be to partition the state estimation and error covariance estimation equations coupled to the Kalman Gain. Experiments considering the proposed parallelization method were accomplished on a VME bus under VxWorks operating system (VxMP). The timing performance for sequential version of KF and parallel version of KF is found out to be 6.35 milliseconds and 6 milliseconds respectively. The time value is the average value after repeatedly executing the program. For more computationally intensive estimation of the state variables, the results show a dramatic decrease in the timing performance for parallel version of the Kalman filter than the sequential version. Hence over a longer period of time, the parallel processing Kalman filter proves to be a better tool for reducing time during computations.

At the SGCS, the process of real-time filtering, fault detection and diagnosis of the measurements are performed. If a fault is detected and diagnosed, a correction is required, the corrective control is generated and up-linked to the satellite before we lose the telemetry signal. Hence it is very crucial that the computation time is reduced at the real-time filtering, fault detection and diagnosis phase, and based on the results obtained, parallel processing Kalman filtering is the viable tool to meet the requirements. One of the salient results in this investigation is that a sustained data rate of 167 Hz for the Kalman filter case can be maintained using a distributed implementation. Future investigation of parallel processing application of the various forms of Kalman filter will address (1) optimizing the de-coupling method among selective multi-processors (2) move to faster hardware for implementations.

13.0 References

[1] Robert Grover Brown and Patrick Y.C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, John Wiley & Sons, Inc, 1997.

[2] Arthur Gelb, Applied Optimal Estimation, The M.I.T Press, Massachusetts Institute of Technology, Cambridge, Massachusetts and London, England, 1974.

[3] Martin Morf and Thomas Kailath, Square-Root Algorithms for Least-Squares Estimation, Transaction on Automatic Control, Vol AC-20, No 4, August 1975.

[4] Mi Lu and Xiangzhen Qiao, Parallel Computation of The Modified Extended Kalman Filter, International Journal of Computer Math. Vol. 45, PP69-87.

[5] J.R. Wertz, Spacecraft Attitude Determination and Control, D. Reidel Publishing Co., Boston, 1980.

[6] David A. Vallado, Fundamentals of Astrodynamics and Applications, The McGraw-

Hill Companies, Inc. College Custom Series, Copyright 1997.

[7] Col Mahamed K. EI-Mahy, An investigation into Kalman filter target tracking algorithms and their real time parallel transputer implementation, Transaction of Measurement and Control, Vol 16, No 1, 1994.

[8] Mi Lu and Xiangzhen Qiao, Parallel Square-Root Algorithm for Modified Extended Kalman Filter, IEEE Transaction on Aerospace And Electronic Systems, Vol 28, No 1. January 1992.

[9] VxWorks 5.4 Programmer's Guide, Edition 1, WindRiver Systems, Inc. 1999.

[10] Tornado 2.0 User's Guide, Edition 1. WindRiver Systems, Inc. 1999.

[11] Gray J.S., Inter-process Communication in Unix, Prentice Hall, Upper Saddle River, NJ, 1997.

[12] Roger W. Johnson, A New Methodology for Launch/Spacecraft Vehicle Ground Health Management, Monitoring and Test and Evaluation (White Paper), Institute for Simulation and Training, University of Central Florida, April 29, 1994.

[13] Roger W. Johnson, A New Real-Time Detection Methodology for Automated Ground Health Monitoring (AGHM) Phase I, Report, University of Central Florida, Sep 30, 1994.

[14] Dr. Roger Johnson, Sanjay Jayaram, " Applications of Extended Kalman Filter at a Satellite Ground Control Station", Second IASTED International Conference on Control and Applications, July 25-29 1999, pp 124-127.

[15] R. Johnson, S. Jayaram, S. Lei, "Distributed Processing Kalman Filter for Automated Vehicle Parameter Estimation – A Case Study", to be presented at IASTED – ASM2000 Conference, Jul 2000, Banff, Canada.