Spacecraft Dynamics Modeling and Simulation Using Matlab-Simulink

Conference Paper · January 2010

CITATIONS
14

READS
11,231

5 authors, including:

Santanu Sarma
University of California, Irvine
76 PUBLICATIONS **604** CITATIONS

SEE PROFILE

Spacecraft Dynamics Modeling and Simulation Using Matlab-Simulink

Santanu Sarma*, A. K. Kulkarni, A. Venkateswaralu, P. Natarajan, and N.K. Malik
ISRO SATELLITE CENTRE, Vimanapura PO, Bangalore-17, India.
Tel: +91805082343, Fax: +91805082306
*Email: santanu@isac.gov.in

Abstract— Digital simulation finds increasing use and is inevitable for better analysis and design of complicated system like spacecraft. This paper presents the practical application of Matlab-Simulink and its features for spacecraft dynamics modeling and simulation. Matlab-Simulink supports building custom libraries using flexible software components that are very user-friendly to use, modify, and build. These features are used to describe the design of a Matlab-Simulink based library for spacecraft dynamics modeling and simulation. Mathematical descriptions of some component of this library are briefly explained and its Simulink model is provided. Typical closed loop simulations with the designed components specifically with thrusters as actuators in Station Keeping mode of a typical geostationary satellite are illustrated. Also possible improvement in simulation performance is discussed.

1. INTRODUCTION

A spacecraft is complicated controlled system that demands digital simulation for better analysis and design. Simulation using a high level language like FORTRAN or C results in very large code whose understanding, handling, and modification becomes increasingly difficult and time consuming. A top-down approach using block diagrams as in Matlab-Simulink [1,2,5] is a suitable solution, providing better and faster development, understanding, visualization, and flexibility. Therefore, a new Matlab-Simulink Library has been designed and developed for simulation of the various classes of spacecraft in various modes and its subsequent improvements are pursued. The library consists of the custom models of various blocks used to describe the spacecraft control system. Matlab's custom component building features using S-functions [5] and its inbuilt components has been used to develop the models of various parts of the spacecraft. In this paper, a brief description of each components and a study of a typical closed loop simulation for a mode of operation are provided. Improvement in simulation performance is also discussed.

2. TYPICAL CONTROL LOOP CONFIGURATION USING THRUSTERS AS AN ACTUATOR

In the past few decades considerable attention has been paid in the design of control schemes of flexible spacecraft. Structural mode interaction with reaction jet control system has been reported [3] as a primary concern for design of future large spacecraft, including the space station, and solution to such problem has been addressed in [4,10,11]. A typical block diagram of the control scheme for such large flexible structures is given in Fig. 1 where PWPFM and TSL represent Pulse Width Pulse Frequency Modulator and Thrusters Selection Logic respectively. Most of the methods [3,4] described requires considerable simulation to validate the concept and flexibility to quickly make changes for studying 'what if' conditions. Matlab-Simulink provides an environment to perform these and provides better visualization and graphics. Combinations of its inbuilt component along with its custom component building features are used to make the necessary components of a new library.

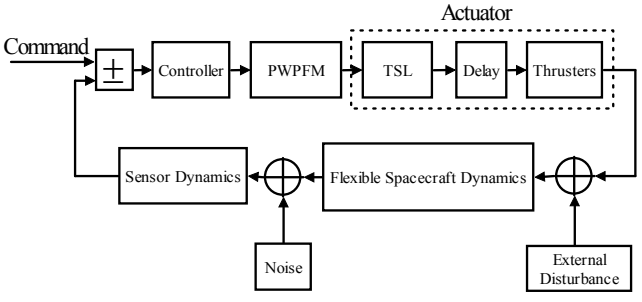


Figure 1: Flexible Spacecraft Attitude Control System.

3. CUSTOM COMPONENT BUILDING IN MATLAB-SIMULINK

Matlab-Simulink provides a power full tool called S-functions to create custom Simulink blocks. An S-function is a computer language description of a Simulink block. S-functions can be written in MATLAB, C, C++, Ada, or FORTRAN. C, C++, Ada, and FORTRAN S-functions are compiled as MEX-files using the mex utility [6]. S-function can be used for a variety of applications, including adding new general purpose blocks to Simulink, adding blocks that represent hardware device drivers, incorporating existing C code into a simulation, describing a system as a mathematical set of equations etc. An advantage of using S-functions is that a general purpose block can be built and used many times in a model, varying parameters with each instance of the block. Simulink also provides a feature called Masking that enables to customize the dialog box and icon for a subsystem. With masking, we can simplify the use of model by replacing many dialog boxes in a subsystem with a single one. Instead of requiring the user of the model to open each block and enter parameter values, those parameter values can be entered on the mask

dialog box and passed to the blocks in the masked subsystem. It provides a more descriptive and helpful user interface by defining a dialog box with user specified block description, parameter field labels, and help text. Masking define commands that compute variables whose values depend on block parameters, create a block icon that depicts the subsystem's purpose, prevent unintended modification of subsystems by hiding their contents behind a customized interface and create dynamic dialogs. Also as the model increases in size and complexity, we can simplify it by grouping blocks into subsystems. Use of subsystems provides the advantage of reducing the number of blocks displayed in the model window keeping functionally related blocks together that enables to establish a hierarchical block diagram, where a subsystem block is on one layer and the blocks that make up the subsystem are on another. The above described featured are used to form the components for the spacecraft dynamics modeling and simulation.

A. *Flexible Spacecraft Dynamics*

The dynamic model of flexible spacecraft structure usually takes into account the structural and sloshing dynamics and the variation in model parameter with different conditions. Here a fairly good approximation of typical communication spacecraft with two flexible solar panels without the sloshing dynamics is presented for clarity. The equations describing such a structure are given by the rigid body and flexible body dynamics as [7]:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \otimes \mathbf{h} + \mathbf{B}\ddot{\mathbf{q}} = \mathbf{T}_c + \mathbf{T}_d = \mathbf{T} \qquad (1)$$

$$\ddot{\mathbf{q}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\mathbf{q}} + \boldsymbol{\omega}_n^2\mathbf{q} + \mathbf{B}^T\dot{\boldsymbol{\omega}} = \mathbf{0} \qquad (2)$$

where,

$$\mathbf{I} = \text{Inertia Matrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix},$$

$$\boldsymbol{\omega} = \text{Body Rates} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T,$$

$$\mathbf{h} = \mathbf{I}\boldsymbol{\omega} = \text{Angular Momentum of rigid body},$$

$$\mathbf{T}_c = \text{Commanded Torque} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T,$$

$$\mathbf{T}_d = \text{Disturbance Torque} = \begin{bmatrix} T_{dx} & T_{dy} & T_{dz} \end{bmatrix}^T,$$

$$\mathbf{B} = \text{Flexible modes coupling matrix},$$

$$= \begin{bmatrix} B_{1x} & \cdots & B_{nx} \\ B_{1y} & \cdots & B_{ny} \\ B_{1z} & \cdots & B_{nz} \end{bmatrix},\ n = \text{no. of modes},$$

$$\mathbf{q} = \text{Flexible Body Coordinates},$$

$$= \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T,$$

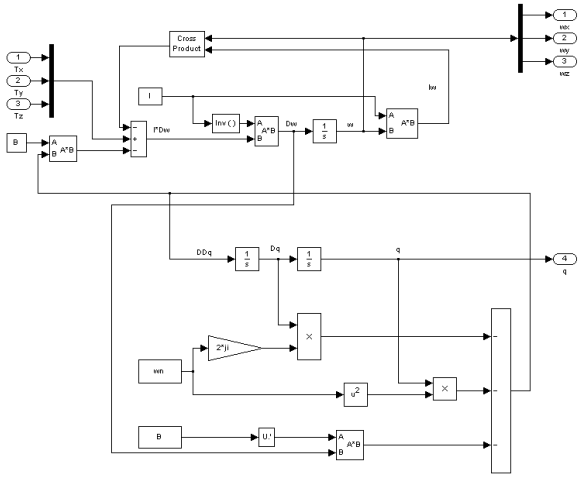
$$\boldsymbol{\xi} = \text{Flexible Mode Damping Ratio},$$

$$= \begin{bmatrix} \xi_1 & \cdots & \xi_n \end{bmatrix},$$

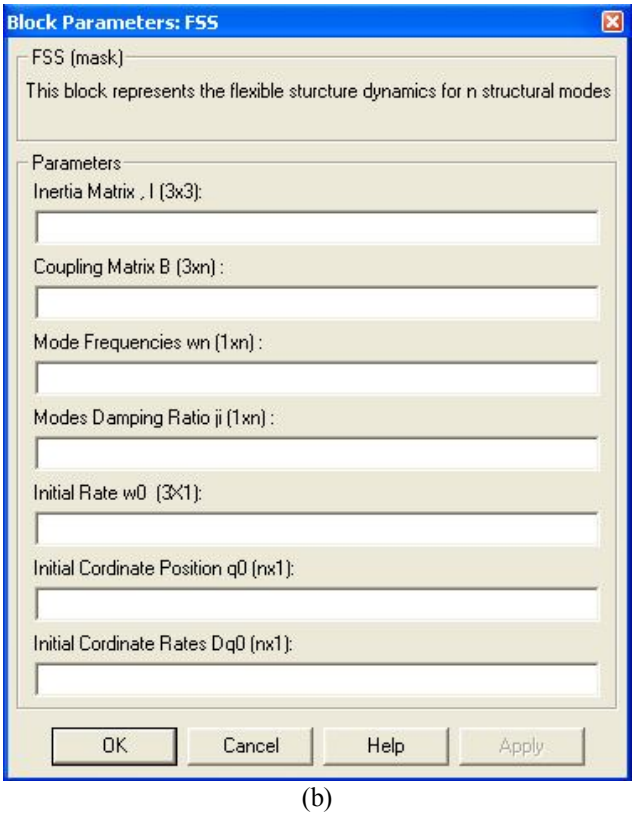
$$\boldsymbol{\omega}_n = \text{Flexible modes frequency},$$

$$= \begin{bmatrix} \omega_1 & \cdots & \omega_n \end{bmatrix}.$$

In (1), $\otimes$ represents cross product and derivatives are represented by dots. The Yaw, Roll, and Pitch axis of the spacecraft is represented by x, y, z respectively. The Simulink model representing (1) and (2) are given in Fig. 2 (a). The masking of this subsystem with suitable variable prompts required for simulation is shown in Fig. 2(b). A logo of a spacecraft to the subsystem can be incorporated to give a associated meaning to the block diagram.



(a)



(b)

Figure 2: (a) Flexible Spacecraft Dynamics Model (b) Dialog Box with Masking.

B. *Sensor Dynamics Model*

Based on the requirement in various modes of operation, different types of sensors are used in a spacecraft. The most common sensors used for attitude control of spacecraft are the gyros, earth sensor, sun sensor and star sensor. During the various modes of operation of the spacecraft the controller processes the data from various sensors like gyro, earth sensor etc. and produces the necessary actuation signal for attitude and orbit correction. In the following subsection, brief mathematical models of the gyroscope and earth sensor are discussed along with their Simulink model.

Dynamic Model of Dynamically Tuned Gyro (DTG)

Gyro based attitude reference systems find increasing importance in satellite attitude control and stabilization due to higher accuracy in measurements of both spacecraft rates and attitude. The latest improvements in various fields of technology have introduced many new types of gyros as well as considerable improvement in performance. Different configuration for attitude reference system has been proposed using conventional single-axis (single-degree-of-freedom (SDOF)) gyros and more popular two-axis dynamically tuned gyros (DTGs) and the newer types of gyros. A DTG offer significant advantages over the single-axis floating gyro particularly for space applications as it provides higher reliability, faster response, two- axis measurement, lesser volume, weight, power consumption and cost, and operation over a wide temperature range. A DTG basically measures angular rates along two perpendicular directions [8, 9] as shown in Fig. (3). The spin axis of the DTG is orthogonal to these measurement axes. The measurements consists of various errors like bias drift, random drift, torquer scale factor, nonlinearity, asymmetry, misalignment etc. which when taken into account gives an improved dynamic model of a gyroscope [8]. Although the model will vary according to the various factors considered like type of gyroscope and its configuration, however only a simplified model of a DTG is discussed here that can fairly describe its characteristics in a dynamic environment. The simplified model of a strapped down gyroscope can be considered as a transfer function relating the input and output rates followed by an integrator giving the attitude angles as in Fig. 4(a). The block diagram describing the dynamics including above mentioned error and even the hardware features is shown in Fig. 4(b). Fig. 4(c) shows a masked dialog box for the block diagram shown in 4(b). The model for other types of gyros are also made in a similar manner and included in the library.

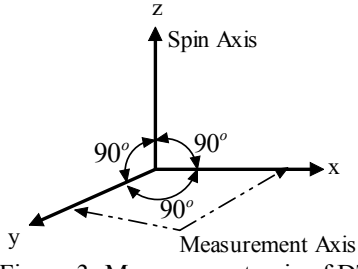
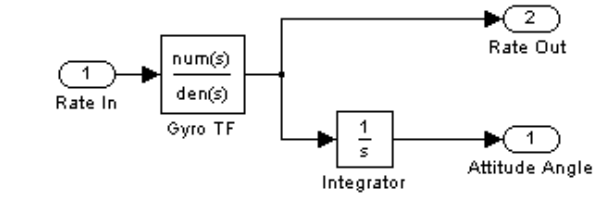
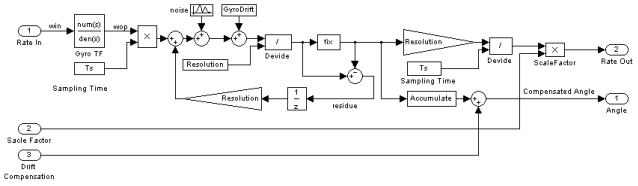
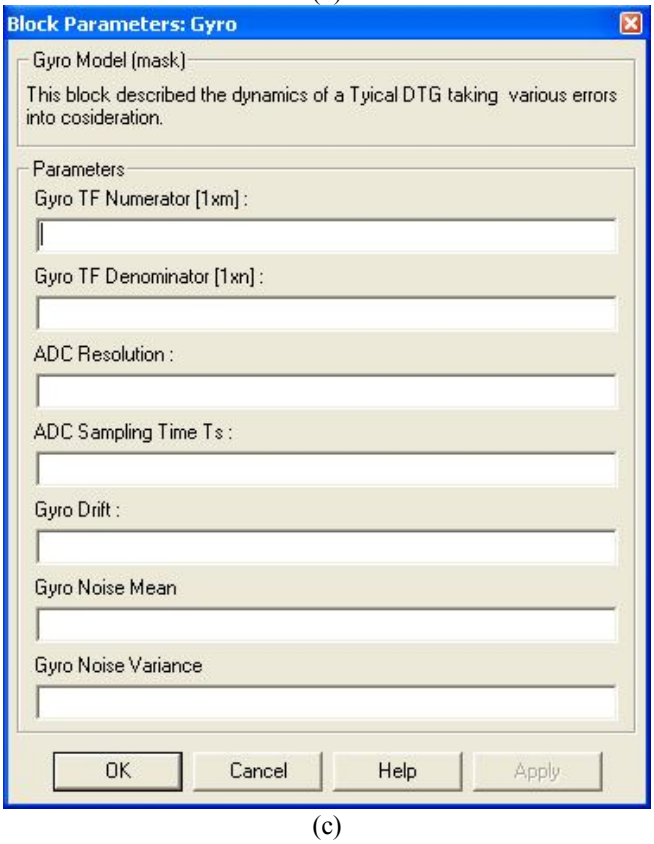


Figure 3: Measurement axis of DTG.



(a)


(b)



(c)

Figure 4: (a) Simplified Gyro Model (b) Elaborate model of DTG (c) Dialog box of the elaborate DTG model.

Dynamic Model of Earth Sensor

The goal of an earth sensor is to determine the spacecraft orientation relative to earth. The principle of operation is either based on the earth's *albed*o which is the fraction of reflected radiation from the globe due to all incident energy that falls on the surface or infrared radiations from earth that is used to define the boundary of the horizon of the earth and subsequently use it for attitude sensing. Infrared base earth sensors using 14.0-16.3 μm (CO_2) band provide more accurate and effective attitude sensing as the energy within that spectrum emitted from all parts of the earths surface is more homogeneous and sharply defines the boundary of the horizon than using albedo. Earth sensors

operate in one of two principle modes. The first is based on dynamic crossing of the earth's horizon (therefore also called *horizon sensors*) and exact determination of the crossing points. The second mode is based on static determination of the location of the earths contour inside the instruments field of view. Earth sensor measures both pitch, θ and roll, ϕ errors of a three-axis stabilized spacecraft. The computation of these angles generally takes into account the surface model of the earth geometrical shape for more accurate result.

The dynamic simulation model of a earth sensor that is used in a typical closed loop simulation of a attitude control system can be modeled using the earth vector E_v obtained from the projection of the direction cosine matrix (DCM) along the yaw axis. The earth sensor head is mounted toward the earth along the yaw axis as shown in the Fig. 5.

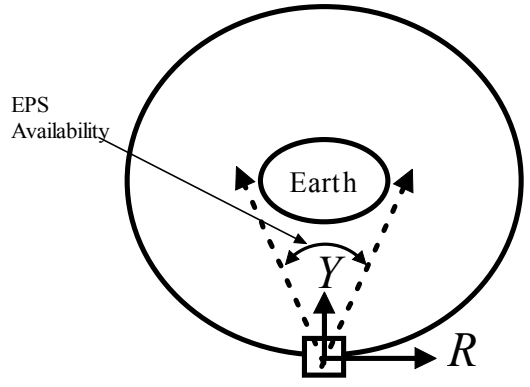


Figure 5: Earth Sensors direction of view and the Earth Presence Signal (EPS) availability.

The DCM is obtained using a transformation of the Quaternion vectors obtained form the solution of (3)

$$\dot{\mathbf{Q}} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{\Omega Q} \tag{3}$$

where

$$\mathbf{Q} = \text{Quaternion Vector},$$
$$= \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 \end{bmatrix}^T;\ Q_4 \text{ the scaler.}$$
$$\mathbf{\Omega} = \begin{bmatrix} 0 & \omega_x & -\omega_y & \omega_z \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix},$$

$\omega_x, \omega_y, \omega_z$ are the body rates of the spacecraft. The Quaternion to DCM transformation is given by

$$DCM = \begin{bmatrix} Q_1^2 - Q_2^2 - Q_3^2 + Q_4^2 & 2(Q_1Q_2 + Q_3Q_4) & 2(Q_1Q_3 - Q_2Q_4) \\ 2(Q_1Q_2 - Q_3Q_4) & -Q_1^2 + Q_2^2 - Q_3^2 + Q_4^2 & 2(Q_2Q_3 + Q_1Q_4) \\ 2(Q_1Q_3 + Q_2Q_4) & 2(Q_2Q_3 - Q_1Q_4) & -Q_1^2 - Q_2^2 + Q_3^2 + Q_4^2 \end{bmatrix}. \tag{4}$$

The Earth Vector E_v as shown in Fig. 6, is obtained from the components of the DCM matrix along the Yaw axis (unit vector along yaw axis is $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$).

$$E_v = \begin{bmatrix} E_{vx} \\ E_{vy} \\ E_{vz} \end{bmatrix} = \begin{bmatrix} DCM \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

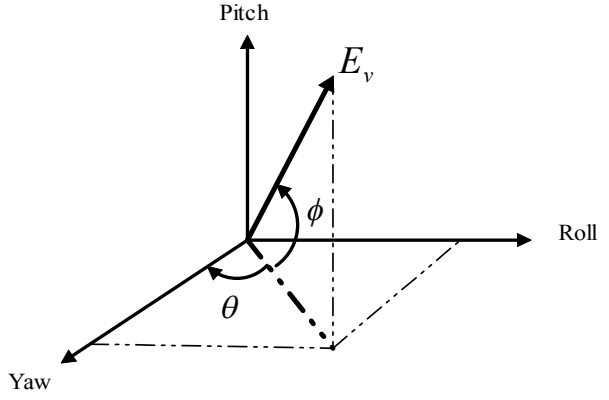


Figure 6: The Earth Vector and the Pitch and Roll Angles.

It is used to compute the Pitch and Roll angle given by

$$\theta = \text{ESPitch} = \tan^{-1}\left[-\frac{E_{vy}}{E_{vx}}\right],$$
$$\phi = \text{ESRoll} = \tan^{-1}\left[\frac{E_{vz}}{\sqrt{E_{vx}^2 + E_{vy}^2}}\right]. \tag{6}$$

One feature of these earth sensors is that the output gets saturated after a threshold value and no signal is available beyond a certain angle indicating the absence of the EPS signal in both the pith and roll axis as shown in Fig. 7. The threshold value and the angle beyond which EPS is absent are generally different for pitch and the roll axis.

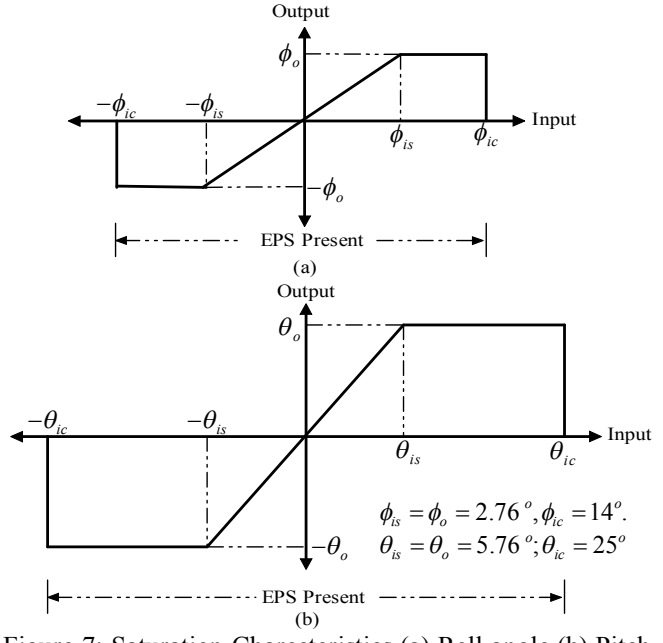


Figure 7: Saturation Charecteristics (a) Roll angle (b) Pitch angle of earth sensor.

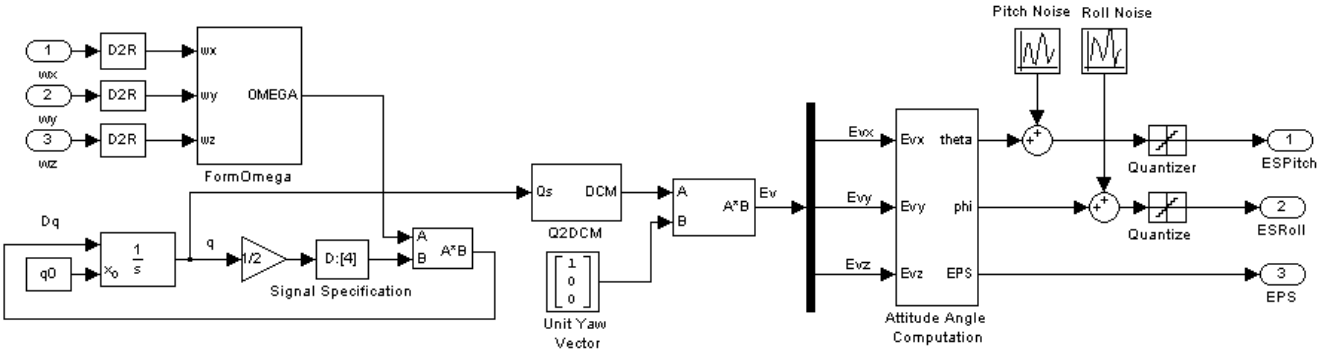


Figure 8: Simulink block diagram of earth sensor model.

The output obtained after saturation is added with noise and quantized to the desired resolution. The Simulink block diagram representation of the earth sensor model is shown in Fig. 8.

A masked dialog box as in the previous section is also created for the earth sensor block. The Attitude angle computation block in Fig. 8 uses an M-file S- function to computes the Pitch and the Roll angles along with the EPS signal and saturates the output as in Fig. 7.

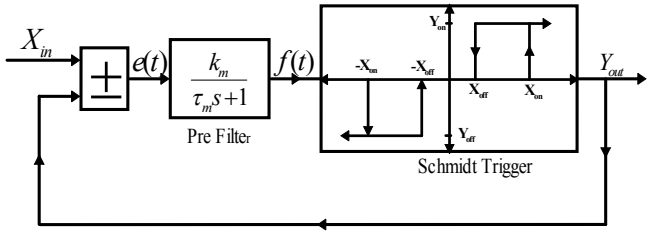
In a similar fashion the model for the Coarse Analog Sun Sensor (CASS) and Digital Sun Sensor (DSS) is also incorporated in the library under the sensors category and provisions are made for future additions and up gradation of the library.

C. *Controller Structure*

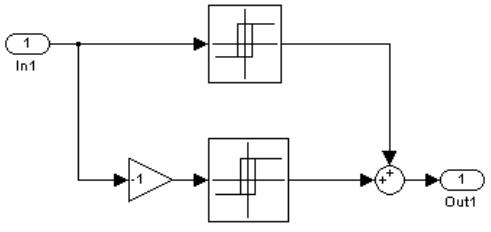
The controller takes the input from various sensors based on the configuration and compares it with the reference to obtain the error signal and computes the necessary actuation signal to be applied to the actuator. Though various forms of control algorithms can be used that is selected based on different specifications and requirements, the present discussions assumes simple structures like PI, PID or PD controller which can be easily modeled using Simulink and hence not elaborated further. Depending on the mode of operation of the spacecraft various sets of gain are selected appropriate for the situation.

D. *The Pulse Width Pulse Frequency Modulator (PWPFM) for On-Off control of Spacecraft*

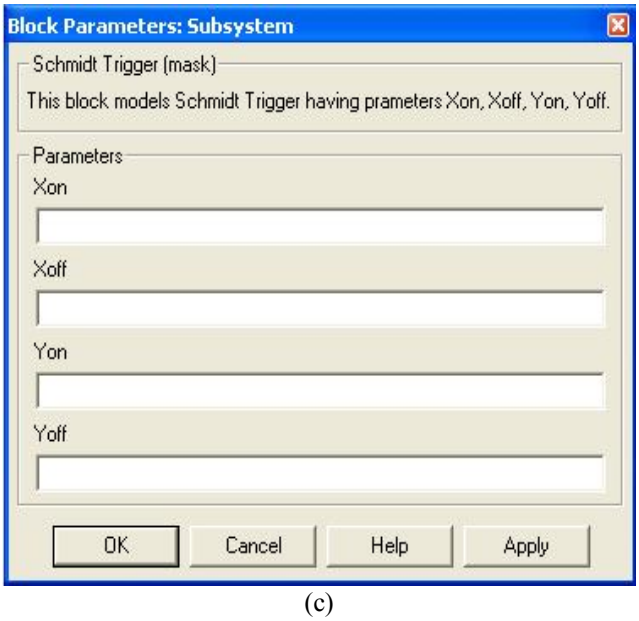
Most of the modern spacecraft attitude control systems employ on-off thrusters, which are controlled using two major approaches viz. bang-bang, and pulse modulation. Bang-bang control is simple in formulation, but results in excessive thruster actions causing more consumption of valuable fuel. Also its discontinuous control actions often interact with the flexible mode of the spacecraft. Therefore it is not commonly used and hence it modeling is not discussed here. On the other hand, pulse modulators are commonly employed due to their advantages of reduced propellant consumption and near linear duty cycle. In general, pulse modulators produce a pulse command sequence to the thruster valves by adjusting pulse width and/or pulse frequency. Pulse modulators such as pseudorate modulator, integral-pulse frequency modulator, and pulse-width pulse frequency modulator (PWPFM) are



(a)



(b)



(c)

Figure 9: (a) Block diagram of PWPFM (b) Simulink model of Schmidt Trigger (c) The Schmidt Trigger dialog box.

also found in literature. Among these, the PWPFM holds several superior advantages such as close to linear operation, high accuracy, etc. and has been used in many satellites like INTELSAT, INSAT and ARABSAT. The block diagram of this is shown in Fig. 9.

As shown in Fig. 9, the PWPF modulator is composed of a Schmidt Trigger, a prefilter, and a feedback loop. A Schmidt Trigger is simply an on-off relay with a deadband and hysteresis. When positive input to the Schmidt Trigger is greater than X_{on}, the trigger output is Y_{on}. Consequently, when the input falls bellow X_{off}, the Schmidt Trigger output is zero. This response is also reflected for the negative inputs. The error signal $e(t)$ is the difference between the Schmidt Trigger output Y_{out} and the input to the PWPFM X_{in}. The error is fed to the prefilter whose output $f(t)$ feeds the Schmidt Trigger. The parameters that governs the operation of the PWPFM are the coefficients of the prefilter k_m and τ_m, the Schmidt Trigger parameters X_{on}, X_{off}, Y_{on} and Y_{off}.

E. *Thruster Actuator Assembly*

The task of a thruster propulsion system is to provide forces and torques acting on the body of the spacecraft, thus enabling changes in its translatory and angular velocities. Spacecraft thruster propulsion system can be divided into three categories viz. cold gas, chemical (solid and liquid) and electrical. A thruster develops its thrust by expelling propellant (such as gas molecules or ions) at a high exhaust velocity relative to the satellite body. When thrusters are used as actuators, it is important to consideration the thrust level, the required numbers and their arrangement or configurations in the body of the spacecraft. Generally six or more thrusters are used to complete a reaction control system taking into consideration the failure and fault tolerant aspects like redundancy. The level of torque that a reaction thruster can apply about a satellite axis depends not only on the thrust level but also on the torque-arm length about the axis. It means that correct thruster use depends primarily on its location on the satellite and also its inclination to the satellite body axis. As different torque levels are needed about three principle body axis, so the locations of the thruster are carefully studied before a final physical setup is adopted for the propulsion system. The location and direction of the thruster is also influenced by the location of the optical sensors and solar panels to avoid damage by thruster plume. Once the location and the canting angle of each thruster is decided along with the thrust level, the torque components applied by a thruster about each body axis, which is a function of the thrusters location and direction denoted by the elevation and azimuth angle are computed. If the thrust vector for the i^{th} thruster is $\mathbf{F}_i$, then the torque about the centre of mass (cm) of the spacecraft will be $\mathbf{T}_i = \mathbf{r}_i \otimes \mathbf{F}_i$, where $\mathbf{r}_i$ is the vector distance of that thruster w.r.t. body axis frame from the centre of mass. When more than one thruster is fired, the torques in each axis due to the individual contribution from each thruster is added to obtain the total torque along each axis of the spacecraft. Usually in practical cases of operation, more than one set of thrusters are used with different thrust level: one set for normal acquisition and another (larger thrust level) may be used for Station Keeping operation in geostationary satellites. Based on the outputs of the PWPFM in the Yaw, Roll, and Pitch axis a set of thrusters are selected and fired for attitude corrections. This operation is usually performed by a Thruster Selection Logic (TSL) table that list all the 27 possible combination of outputs of the Yaw, Roll and Pitch PWPFM and the corresponding thrusters is selected for firing, which is the total torque experienced along each axes tabulated in Table 1. A similar table is used to describe the TSL during station keeping operation to impart incremental velocity to the spacecraft. In this Tables '1' represents the ON condition of the thrusters whereas '0' the OFF state. The thrusters that need to be fired are decided by the torque requirement along various axis and control methodology.

Table 1: Thruster Selection Logic

SL No	PWPFM Outputs			Tx	Ty	Tz	Set 1 (Normal)				
	Y	R	P				R1	R2	R3	:	Rm
1	1	1	1	Tx1	Ty1	Tz1	1	0	0	..	1
2	1	1	0	Tx2	Ty2	Tz2	1	1	0	..	0
3	1	1	-1	Tx3	Ty3	Tz3	.	.		.	.
. .	. .	. .	. .	. .	. .	. .	.	.		.	.
27	-1	-1	-1	Tx27	Ty27	Tz27	.	.		.	.

1= Thruster On; 0= Thruster Off.

The above TSL logic can suitably be implemented using an M-file S-function containing if than else statements. This S-function may have three inputs and outputs. The inputs are the PWPFM Yaw, Roll and Pitch outputs pulses and the output of the S-function is the total torque along the three body frame axis. The call back method that this S-function uses is depicted in Fig. 10. A similar S-function can be used to describe the torque experience by the spacecraft during station keeping operation for the second set of thruster with higher thrust level. Whenever a change of

mode of operation from normal to station keeping or vise-versa is required a suitable switching between these two TSLs can be obtained as shown in Fig. 11.The structure of the M-file S-function used to realize the TSL logic is given bellow:

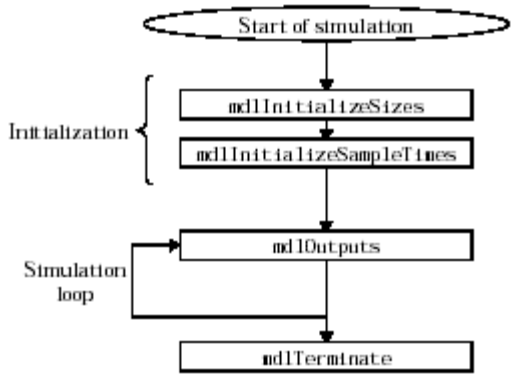


Figure 10: The callback method of TSL S-function.

```
% Structure of the M-file S-Function
%==================================
function [sys,x0,str,ts]=pwpfm2torque(t,x,u,flag)
switch flag,
case 0
   [sys,x0,str,ts]=mdlInitializeSizes;
case 3
      sys=mdlOutputs(t,x,u);
case {1, 2  4, 9}
   sys=[];
otherwise
   error(['Unhandled flag = ', num2str(flag)]);
end
%==================================
function  [sys, x0,str,ts]=mdlInitializeSizes
sizes=simsizes;
sizes.NumContStates=0;
sizes.NumDiscStates=0;
sizes.NumOutputs=3;
sizes.NumInputs=3;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
sys=simsizes(sizes);
x0=[];
str=[];;
ts=[-1 0];
%==================================
function   sys=mdlOutputs(t,x,u)
Yaw=u(1);
Roll=u(2);
Pitch=u(3);
if Yaw==1&Roll==1&Pitch==1          %1
   sys=[Tx1    Ty1    Tz1 ];
elseif Yaw==1&Roll==1&Pitch==1      %2
   sys=[Tx2    Ty2    Tz21 ];
……                                  .....
……                                  .....
elseif Yaw==-1&Roll==-1&Pitch==-1  %27
   sys=[Tx27  Ty27    Tz27];
end
% End of S-Function
%==================================
```

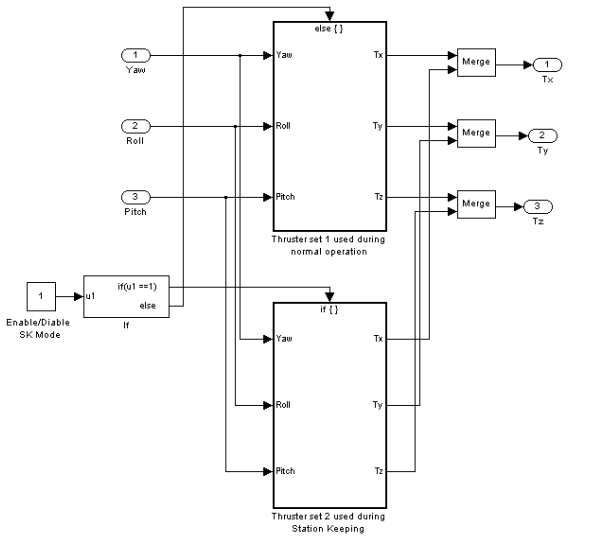


Figure 11: Switching between TSL from Normal to Station Keeping mode of operation and vise-versa.

4. THE SPACECRAFT CONTROL LIBRARY

The described components are put together in a systematic manner in a library named Spacecraft Control Library. It is divided into groups like spacecraft dynamic models, Sensor models, actuator models, controllers, disturbance model etc as shown in Fig. 12. The sensors models contain the models of the various sensors used in a spacecraft whereas the disturbance model will contain the model of the various disturbances experience by the spacecraft and other subsystems.

Figure 12: The Spacecraft Control Library.

5. CLOSED LOOP SIMULATION IN STATION KEEPING MODE

In this section a closed loop simulation using the designed library components is carried out for a geostationary

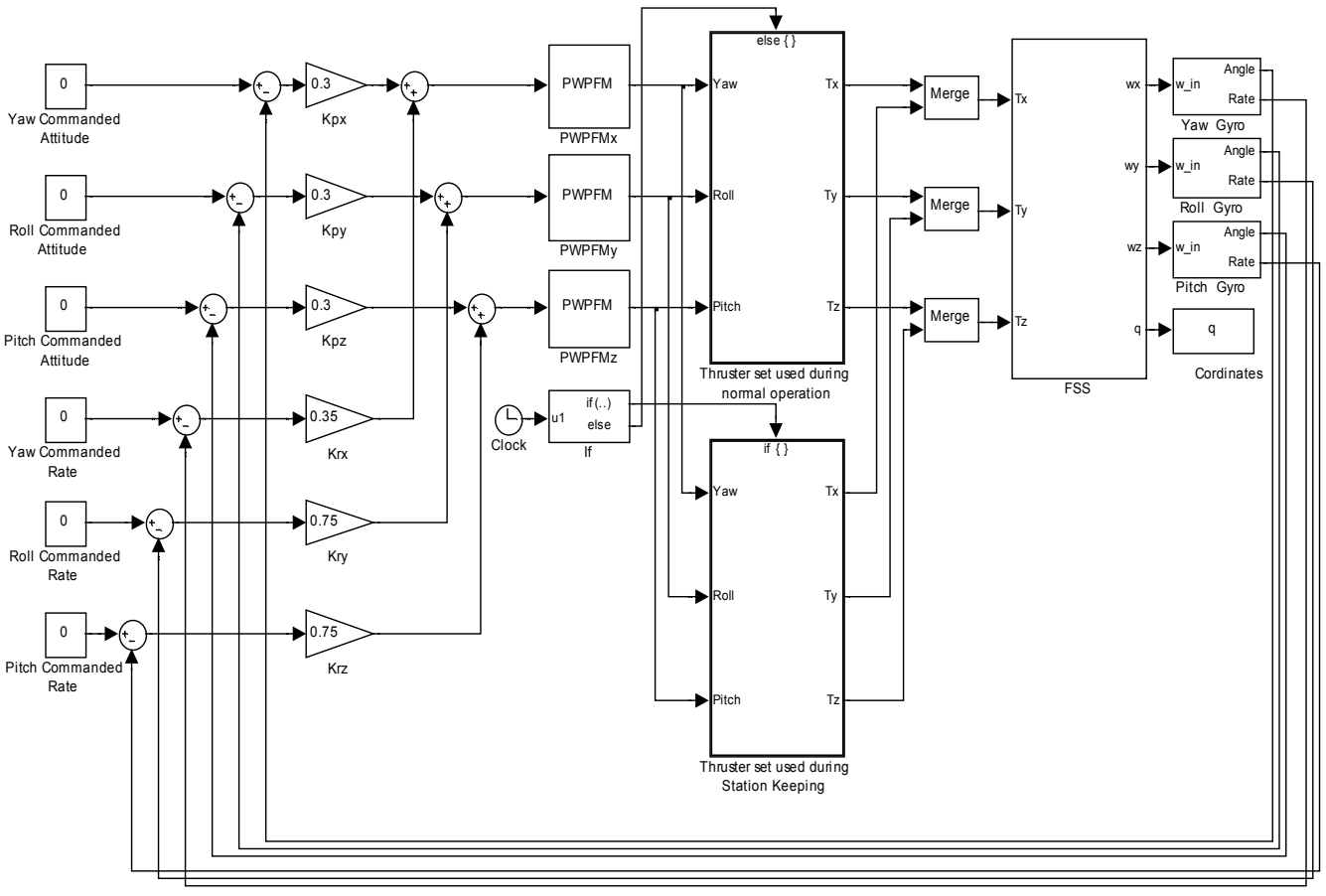


Figure 13: Closed loop simulation in Station Keeping mode and tranfer to normal mode.

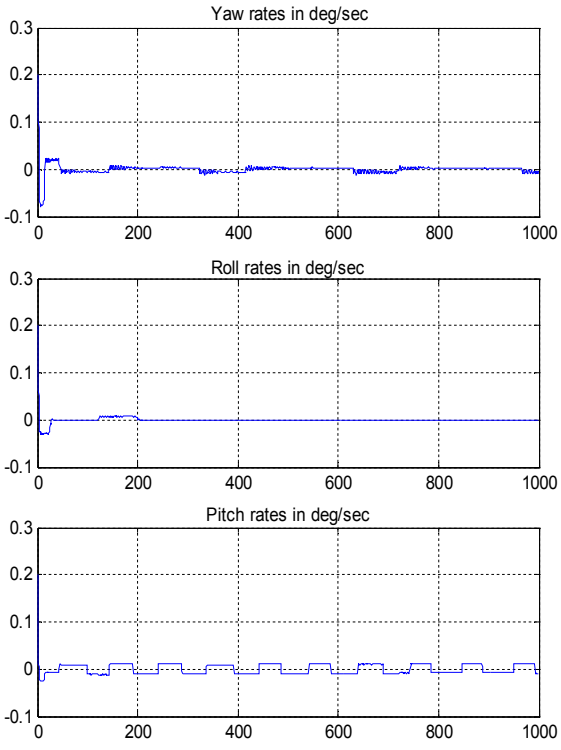
satellite with $n = 10$ number of flexible modes (five each of two panels) with the following parameters:

$$\mathbf{I} = \begin{bmatrix} 1145.400 & -29.000 & -17.90 \\ -29.000 & 1494.300 & -2.000 \\ -17.9000 & -2.000 & 809.7000 \end{bmatrix},$$

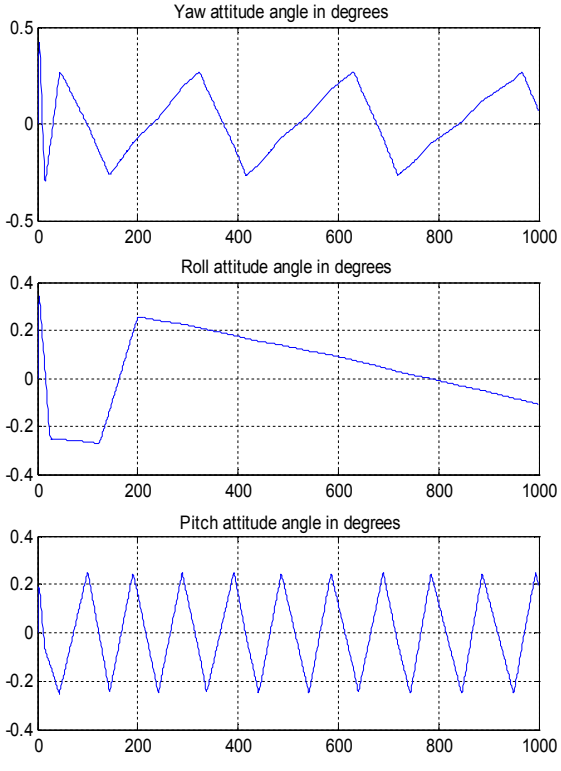
$$\mathbf{B} = \begin{bmatrix} 17.1349 & -0.0335 & 1.4460 \\ -0.0430 & 0.1198 & -3.7999 \\ -0.0019 & -18.1974 & -0.0603 \\ 4.3221 & 0.1456 & 0.7813 \\ 0.0605 & -0.0345 & -1.8578 \\ -17.0955 & 0.0053 & 1.4476 \\ -0.0432 & -0.1486 & 3.8005 \\ -0.0007 & 18.1607 & -0.0090 \\ 4.2920 & -0.0219 & -0.7811 \\ -0.0598 & 0.0081 & -1.8576 \end{bmatrix}^T, \boldsymbol{\omega}_n = \begin{bmatrix} 3.7699 \\ 11.8752 \\ 13.9487 \\ 21.9911 \\ 39.2699 \\ 3.7071 \\ 11.8124 \\ 13.5717 \\ 21.9911 \\ 39.2699 \end{bmatrix},$$

$$\boldsymbol{\omega}_0 = \begin{bmatrix} 0.2000 \\ 0.2000 \\ 0.2000 \end{bmatrix} \text{ deg/sec}, \mathbf{q_0} = [\mathbf{0}], \dot{\mathbf{q}}_\mathbf{0} = [\mathbf{0}], \xi_{1..n} = 0.002,$$

where $\boldsymbol{\omega}_0$, $\mathbf{q_0}$ and $\dot{\mathbf{q}}_\mathbf{0}$ are the initial condition of the body rates, the flexible coordinates and the coordinate rates respectively. The closed loop control scheme with a PD controller followed by a PWPF modulator and TSL is shown in Fig. 13. The proportional and derivative gains used in the simulation are $K_{px} = K_{py} = K_{pz} = 0.3,$ and $K_{rx} = 0.35, K_{ry} = K_{rz} = 0.75,$ respectively. The PWPFM parameters are $k_m = 12.25,$ $\tau_{\mathrm{m}} = 0.128,$ $X_{on} = 1, X_{off} = -0.15, Y_{on} = 1, Y_{off} = -1$ in all three axes. The flexible spacecraft rates and attitude angle are sensed by either the gyros or combination of gyro, earth and /or sun sensor. In the above simulation only a gyro is used for both the rates and the attitude angle needed by the controller. The rates and attitude angle observed by the gyro in all the three axes with the normal set of thrusters are shown in Fig. 14. When a switching is performed between the normal and station keeping thruster sets between 100 to 500 seconds, the rate and attitude angles sensed by the gyro is shown in Fig. 15. The response of the flexible coordinates for both the panels is shown in Fig. 16. The output of the PWPFM and the corresponding torque applier by the thrusters on the spacecraft are shown in Fig. 17. It can be observed that the firing of the station keeping thrusters produces more oscillations in the flexible coordinates than the normal thrusters. Sometimes improper design may lead to build up of oscillations, which are harmful for the spacecraft. For details of control design methodology refer [10, 11].

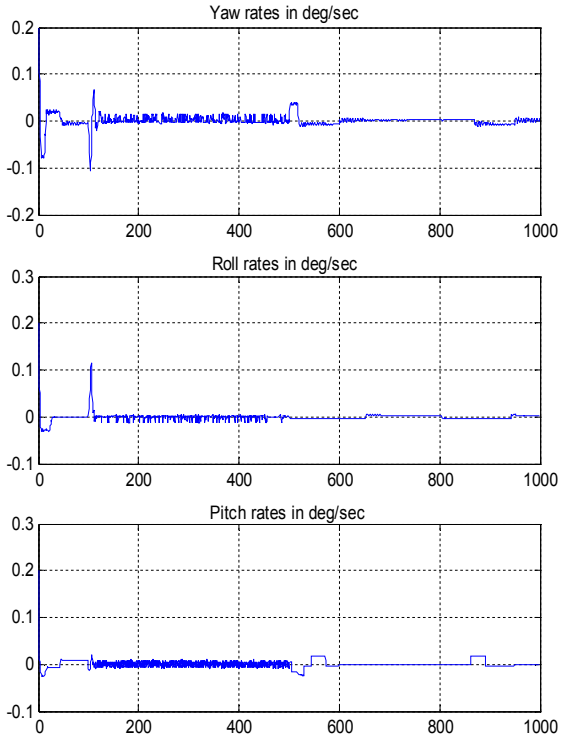


(a)

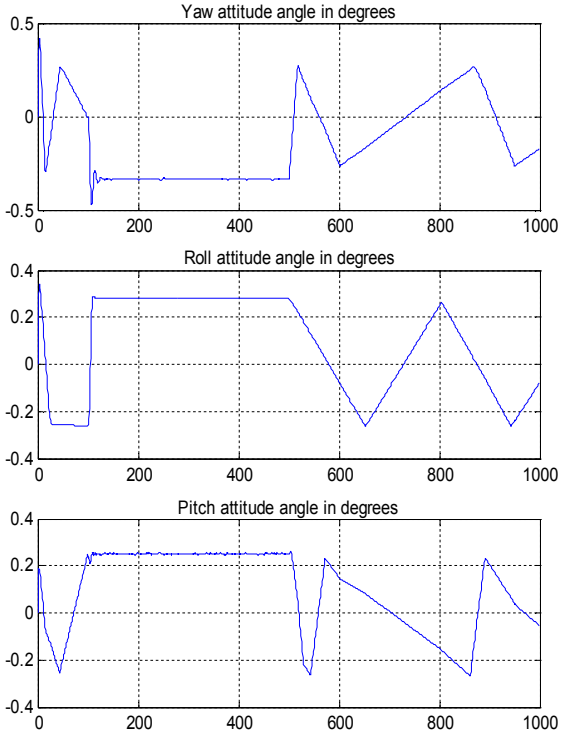


(b)

Figure 14: (a) Rates (b) Attitude angles in Yaw, Roll, and Pitch axis.

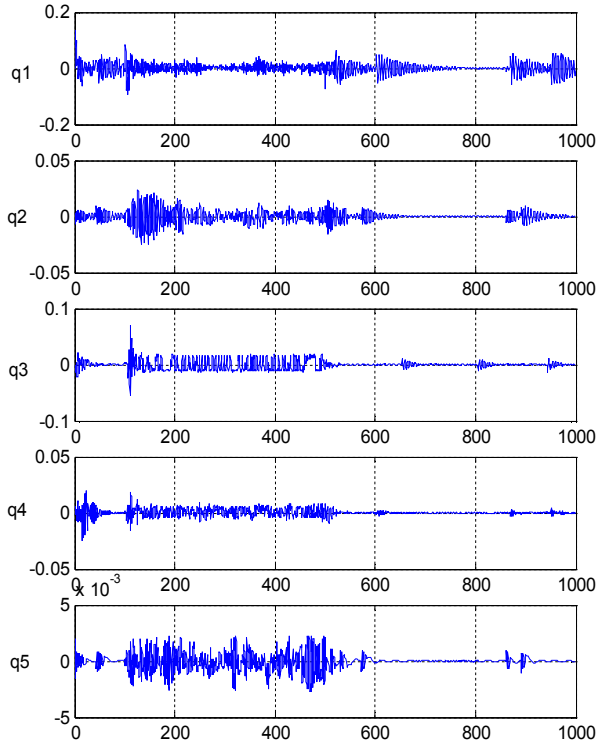


(a)

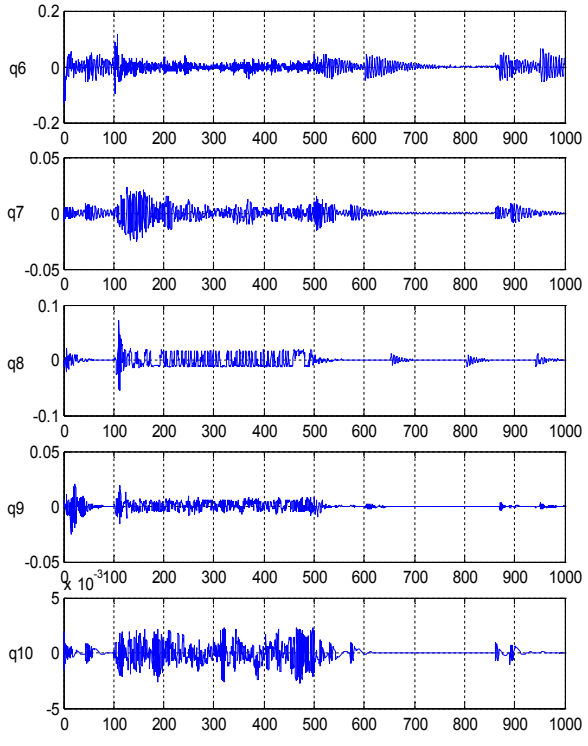


(b)

Figure 15: (a) Rates (b) Attitude angles in Yaw, Roll, and Pitch axis when switching is performed bteween normal and SK mode thrusters.

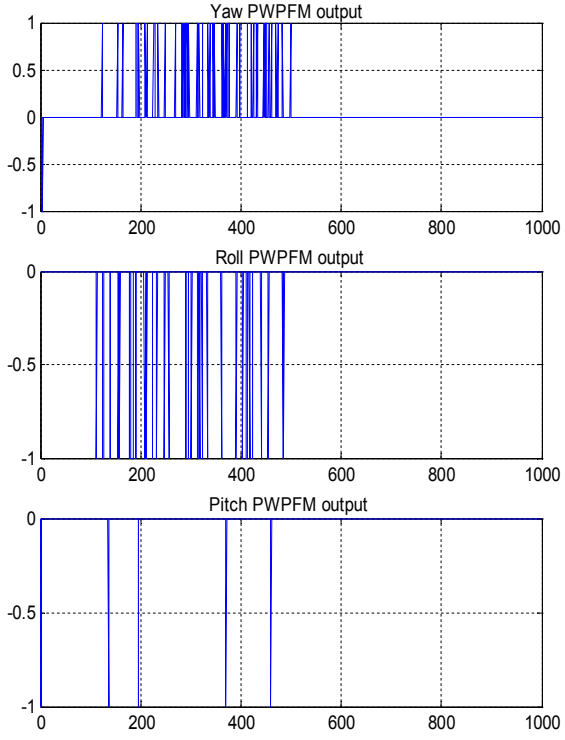


(a)

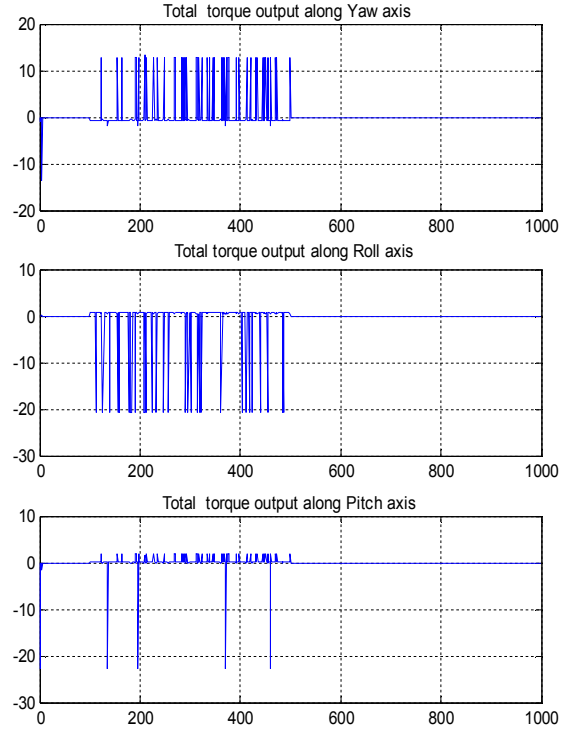


(b)

Figure 16: Flexible coordinates response (a) Solar Panel 1 (b) Solar Panel 2.



(a)



(b)

Figure 17: (a) PWPFM outputs (b) Total Torque produce due to thruster firing in Yaw, Roll and Pitch axis.

6. IMPROVEMENTS IN SIMULATION PERFORMANCE

As described in the previous section that the closed loop simulation involves quiet a lot of computation and may take some time for simulation using Matlab-Simulink, few measures can be taken to reduce the simulation time. The simulation performance can be improved if any algebraic loop [2] in the model is avoided. Although Matlab has the power to solve some of the algebraic loop problems, but it considerably slows down the simulation. Also, excessive scopes for display if avoided and replaced with the data logging function "to workspace" along with the intelligent use of the decimation property can also speed up the simulation. All the M-files s-function used to define custom components takes larger time as it is interpreted language, and as far as possible can be compiled to its MEX format to speed up the simulation. Above all Matlab has a powerful feature called rapid simulation target incorporated in its Real-Time Workshop [12] that converts the Matlab model into C/C++ and produce an executable that can run much faster than the Matlab-Simulink model. This is an interesting area of study and may be a separate topic of discussion.

7. CONCLUSIONS

Digital simulation of complex systems like spacecraft is important as it provides better insight, understanding and tremendously helps in analysis and design. This paper described a design of a Matlab-Simulink based custom library for spacecraft control system modeling and simulation. Mathematical descriptions of some component are briefly explained and its Simulink model is provided. Typical closed loop simulations of geostationary spacecraft in station keeping mode and is transition to normal, using the custom library components specifically with thrusters as actuator is illustrated. Possible improvement in simulation performance is provided that can be incorporated while making the custom components. The possibility of easy up gradation of the library with new addition or modifications in components can easily be done. In case of large complicated systems involving huge computations, the rapid simulation target of Real-Time Workshop can be used and its details are pursued further.

REFERENCES

[1]. *Using Matlab Version 6*, The MathWorks, Inc. November 2000.

[2]. *Using Simulink Version 4*, The MathWorks, Inc. November 2000.

[3]. T.C. Anthony, B. Wie and S. Carroll "Pulse-Modulated Control System for a Flexible Spacecrfat" *Journal of Guidance* Vol.13, No.6, pp.1014-1022, Nov-Dec 1990.

[4]. G.Song, N.V. Buck and B.N. Agrawal "Spacecraft Vibration Reduction Using Pulse-Width Pulse-Frequency Modulation Input Shaper," *Journal of Guidance, Control, and Dynamics,* Vol.22, No.3, pp.433-440, May-June 1999.

[5]. *Writing S-Functions Version 4*, The MathWorks, Inc. November 2000.

[6]. *External Interface/API Version 6,* The MathWorks, Inc. November 2000.

[7]. Marcel J. Sidi, *Spacecraft Dynamics and Control: A practical engineering approach,* Cambridge University Press, 1997.

[8]. R.J.C. Graig, "Theory of operation of elastically supported tuned gyros," *IEEE Transactions in Aerospace and Electronic Systems*, Vol.8 pp.280-288, May 1972.

[9]. E.D. Howe and P.H. Savet, "The dynamically tuned free rotor gyro," *Control Engineering*, II, pp.67-72, June 1964.

[10]. C.L.Kirk and J.L. Jumkins, *Dynamics of Flexible Structures in Space*, Springer Verlag, Berlin, 1990.

[11]. Bong Wie, *Space Vehicle Dynamics and Control*, AIAA Educational Series, USA, 1998.

[12]. *Real-Time Workshop User Guide*, *Version 4*, The MathWorks, Inc. September 2000.