Dincon'2007

6° Brazilian Conference on Dynamics, Control and Their Applications
May 21-25, 2007
UNESP – Campus de São José do Rio Preto, SP, Brazil

AN ATTITUDE SIMULATOR TO SUPPORT SPACE MISSIONS

Valdemir Carrara[1] *and* *Carlos Henrique Gustavo Hassmann*[1]

[1]Departamento de Mecânica Espacial e Controle (DMC), Instituto Nacional de Pesquisas Espaciais, São José dos Campos/SP, Brasil,
val@dem.inpe.br, hassmann@terra.com.br

Abstract: The spacial missions will have a high automatization level, making the pointing precision growing up. The control system trust will be very important. It is possible to see that the tests will have more hardware into de software mesh, so the process starts from a control system complete simulation and, slowly, the on board computer and sensors will be added, using the real system parts, and simulating at a computer only what is necessary.

Keywords: satellite attitude, attitude control, attitude simulation.

1. INTRODUCTION

The next Brazilian space mission will have an important improvement in on board control systems. Past satellites had passive attitude stabilization with on ground attitude determination and control. Next generation of satellites shall have high degree of autonomy concerning attitude determination and control, by moving pointing accuracy up to tenths of grades, based on recent technology of sensors and actuators. As a result of the on board autonomy the reliability of the control system shall be high, requiring a new verification and qualification level in closed loop tests. In order to guarantee these reliability figures, the test environment shall change from software simulation to hardware in-the-loop tests, with some intermediate configuration levels. In other words, the process starts from a complete simulation of the control system (Figure 1), and progressively includes real-time simulation, the onboard computer, the sensor electronics and finally the complete set of sensors with external stimuli. During these phases the onboard program will be revised and updated according to the tests results and software product assurance requirements. The goal behind this scheme is to use real system parts, and to simulate in computer the parts that can't be assembled in the test bed. As the space microgravity can't be simulated in a lab, so the attitude dynamics must be essentially programmed. This work describes the design approach for this attitude simulator. Main requirements to design are: it shall be modulate, portable for different operational systems, usage of C language, and also portable to upgrade to an object oriented language, making it easier to change to different test configurations. At the present time the attitude simulator is coded and tested, including non-rigid satellite dynamics due to solar array motion, attitude and orbit propagation, orbital ephemerid like Sun and Earth directions, Earth magnetic field strength, sensor simulation (magnetometer, analog solar sensor, inertial unit, star sensor and GPS receiver), actuators models (reaction wheels, magnetic coils and thrusters), and solar panel drive (BAPTA). It is planned to include shortly the attitude environmental perturbations and real time procedures.

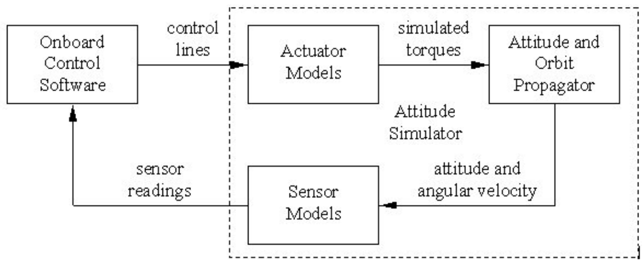

Figure 1 – Interface between the simulation program and onboard control software.

2. MATHEMATICAL MODELS

The mathematical model considers the satellite as composed by several rigid bodies linked by articulated joints (solar panels for instance). The motion of these appendages can be expressed in function of time, because its time behavior is well known and this allows the numerical integration of the attitude in terms of a single body instead of several. The vector differential equation of attitude motion is given by [3]:

$$\begin{aligned} \dot{\omega}_0^0 = {} & (I_o + J_n)^{-1}\left[\sum (N_{cont} + N_{per}) + \right. \\ & \left. -\Omega(\omega_o^o)\left(I_o \omega_o^+ h\right) - \dot{h} - \dot{H}_n\right] \\ \dot{h} = {} & T \end{aligned} \qquad (1)$$

where I_o is the inertia matrix of the satellite body, N_{cont} are the control torque, N_{pert} are the perturbation torques, h is the angular momentum stored in the reaction wheels, T are the internal torques applied at the wheels, ω_o^o is the angular velocity of the satellite and $\Omega(\cdot)$ indicate a matrix operator that makes the cross product of the argument [11]. J_n e H_n are given by

$$
\begin{aligned}
J_n = & \sum_{k=1}^{n} \left(A_{k,o} I_k A_{k,o}^T - m_k \Omega \left(a_{ok}^o - a_{k,o}^o \right) \left(a_{o,k}^o - a_{k,o}^o \right) \right) + \\
& + \frac{1}{m_t} \left[\sum_{k=1}^{n} \left(m_k \Omega \left(a_{ok}^o - a_{k,o}^o \right) \right) \right]^2 \\
\dot{H}_n = & \sum_{k=1}^{n} \left[\Omega \left(\omega_o^o + \omega_k^o \right) A_{k,o} I_k A_{k,o}^T \left(\omega_o^o - \omega_{k,o}^o \right) \left(a_{o,k}^o - a_{k,o}^o \right) \right] + \\
& + \frac{1}{m_t} \left[\sum_{k=1}^{n} \left(m_k \Omega \left(a_{ok}^o - a_{k,o}^o \right) \right) \right]^2
\end{aligned} \tag{2}
$$

and β_k, μ_k e r_{cm} came from

$$
\begin{aligned}
\beta_k = & \ \Omega(\omega_o^o)\Omega(\omega_o^o) o k^o + \\
& -\Omega(\omega_o^o + \omega_k^o)\Omega(\omega_o^o + \omega_k^o) a_{ko}^o + \\
& +\Omega(a_{ko}^o)(\Omega(\omega_o^o)\omega_k^o + \dot{\omega}_k^o) \\
\mu_k = & \ \frac{m_k}{m_o + \sum_{j=1}^{n} m_j} = \frac{m_k}{m_t} \\
r_{cm}^o = & \sum_{k=1}^{n} \mu_k (a_{ok}^o - a_{ko}^o)
\end{aligned} \tag{3}
$$

In these equations I_k is the inertia matrix of the appendage k, m_k is its mass, $A_{k,o}$ is the rotation matrix between this appendage and the satellite body, and $\omega_k(t)$ is the angular velocity of its joint. Superscript in those variables means the coordinate system where it is expressed: o for the satellite body and k for appendage k. Each appendage k presents ten scalar parameters (θ_0, d_0, a_0, t_0, θ_1, d_1, a_1, t_1, θ_2 e d_2) of the Denavit-Hartenberg ([1, 4]) parameters that are used in kinematics of robotic arms, in order to perform coordinate transformations, as shown in Figure 2. The advantage using this representation is a set of specific rules established for the Denavit-Hartenberg parameters that allows defining each parameter based only in geometry. If $R_i(\theta)$ represents the rotation matrix around axis i of an angle θ, them the $A_{k,o}$ matrix results in

$$
A_{k,o} = R_z(\theta_o) R_x(t_o) R_z(\theta_k(t)) R_z(\theta_1) R_x(t_1) R_z(\theta_2) \tag{4}
$$

where $\theta_k(t)$ is the rotation angle of appendage k. The vectors a_{ok} and a_{ko} represents the origin position of the coordinate system j fixed in the joint axis in satellite body frame and appendage k frame, respectively, as seem in Figure 2. When expressed in satellite body coordinates, these vectors are given by

$$
\begin{aligned}
a_{ok}^o &= R_z(\theta_o) v_o \\
a_{ko}^o &= -R_z(\theta_o) R_x(t_o) R_z(\theta_k(t)) R_z(\theta_1) \left[v_1 + R_x(t_1) R_z(\theta_2) v_2 \right]
\end{aligned} \tag{5}
$$

where $v_0 = (a_0, 0, d_0)$, $v_1 = (a_1, 0, d_1)$ and $v_2 = (0, 0, d_2)$. Finally, the joint angular velocity ω_k and acceleration $\dot{\omega}_k$ can be obtained from

$$
\begin{aligned}
\omega_o^o &= \dot{\theta}_k(t) R_z(\theta_o) R_x(t_o) z \\
\dot{\omega}_o^o &= \ddot{\theta}_k(t) R_z(\theta_o) R_x(t_o) z
\end{aligned} \tag{6}
$$

in which z is the unit vector $(0, 0, 1)$. It can be noted that

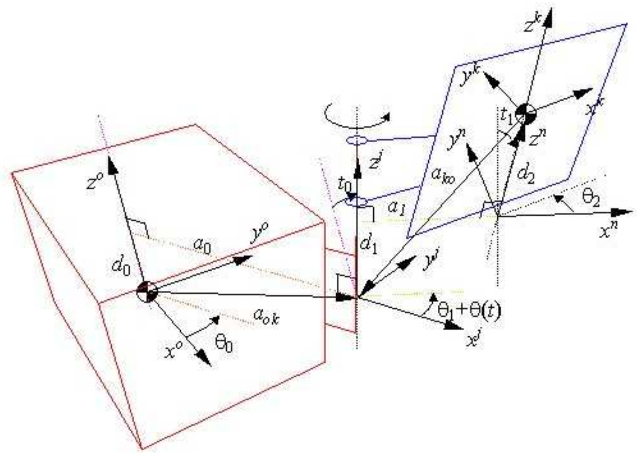

Figure 2 – Main body coordinate system, of the join and appendix k.

the joint torques, based on the dynamics of the articulated bodies, was replaced in the formulation presented here by the angular acceleration of the joints, $\ddot{\theta}_k$. In fact this is not totally true, since the angular acceleration depends on the appendage and satellite inertias, among other parameters. But is also true that some driver mechanisms like BAPTA (Bearing and Power Transfer Assembly) rely on step motors, whose behavior is more likely related to position and velocity control than to torque. Besides, this model needs to integrate only the angular acceleration of each appendage, so as to obtain both position and velocity. It takes into account the changing of the total satellite inertia (including appendages), and also the center of mass motion due to asymmetrical mass distribution caused by appendage rotations.

3. SIMULATOR STRUCTURE

The attitude simulator was designed to work as a state machine. In other words, it was constructed in such a way that some functions allow defining the simulation environment and satellite configuration before doing attitude propagation. The simulator was developed in C language (Ansi C) because the controller shall also be programmed in this language. Functions were grouped in 8 modules:

1. Attitude Propagation
2. Coordinate conversion
3. Control functions
4. Orbital ephemerid data
5. Sensors simulation (magnetometer, GPS, solar sensor, inertial unit, star sensor).

6. Actuator Simulation (reaction wheels, magnetic coils, thrusters and solar panels rotating system - BAPTA).

7. Environmental Perturbations

8. Real time operations

Additionally, it was created some structures to manipulate vectors and matrices, and also some math operators were extended in order to perform matrices and vector algebra (like products, inner product, cross product, sum, etc.). These structures are:

- `matrix3` - structure with 9 elements (square matrix with order 3).
- `vector3` - structure with 3 elements (vector with order 3).
- `vector6` - structure with 6 elements (matrix with 2 vectors with order 3).
- `quaternion` - structure with 4 elements,

and their respective operators. A product of a matrix `M` and a vector `P` is coded as `M*P`. It was implemented functions to invert and to transpose matrices too.

3.1. Attitude simulator functions

The attitude simulator is composed of functions to perform several tasks, including torque generation, low level actuator modelling, numerical propagator adjustment and satellite mass and inertia configuration (prototypes for all functions and physical constants are easily found in the `att_pro.h` header file). These functions were grouped in some topics:

- External control torques or environmental perturbations (yet to be modelled)
- External torques in coils
- External torques in propellers
- Internal torques in reaction wheels
- Satellite properties
- Attitude integrator parameters
- Initial conditions for attitude
- Attitude propagator
- Ephemerid calculus

3.2. Coordinates conversion

The satellite attitude can be defined by Euler angles that consist in three coordinates rotations over the Cartesian axis or by the Eulert's angle and vector, which describes the rotation axis and rotation angle, or by the quaternion and also by attitude matrix. There exists 12 combinations for Euler angles, but only 2 are of interest in satellite attitude. They are the rotation sequence over the x-y-z and z-x-z axes. These rotations are known as 1-2-3 and 3-1-3 [11]. So there are 5 different forms for attitude representation that solve basically any type of problem involving the attitude visualization, control or initial condition. Coordinate transformation from one to other attitude parameters are presented at the Figure 3. Arrows indicate the implemented transformation functions, whose name is formed by the transformation itself. For instance, to transform from a Euler x-y-z rotation to a quaternion representation the function is `exyzquat(vector3 euler)`. These functions return pointers to a structure compatible to the expected attitude parameter. In this case `exyzquat` is a quaternion type structure. Even considering there is no function like `exyzezxz`, this transformation is still possible by means of function cascading: `rmxezxz(vector3) exyzrmx(vector3 euler)`.

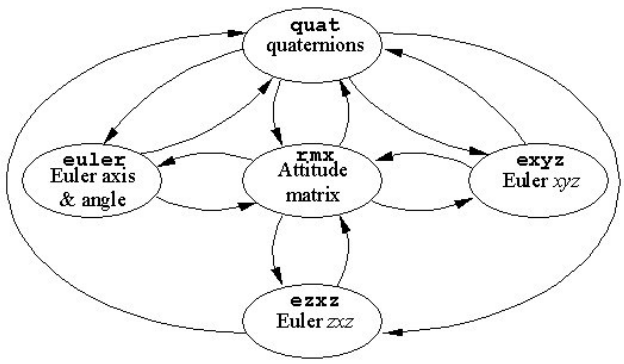

Figure 3 – Different attitude representations and the possible conversions

3.3. Orbital Ephemerid

Although the orbital motion of the satellite can be disregard for short time attitude simulation, it strongly affects the attitude for long time propagation (one orbit or so). These effects are named orbit ephemeris and are related to relative position of Earth, Sun and Earth's magnetic field to the satellite attitude. Normally it is necessary to add the ephemerid computation in the simulator environment because the attitude sensors need those data. The implemented functions perform the following tasks:

- Date and hour transformations
- Ephemerid conversions [9]
- Sun position [7, 9, 10]
- Earth magnetic field (IGRF10)[12, 13]
- SGP8 [8] and analytical [2] orbit propagation.

3.4. Equipment Simulation

Some hypotheses were considered in the modelling of attitude sensors and actuators. In first place it was necessary to admit that a computer simulation will never be so good as the real system or equipment. This means that a computer system or a mathematical model can be very complex, but still have an error compared to the real system. By the other hand, a strong simplification of the model could create an unrealistic simulation. The equilibrium will be the point that allows a good simulation, but not too complex that exceeds real time constraints. Unfortunately, there is no other way except trial and feeling to do that. The approach adopted here was to implement a model that can represent many (or even all) sensors of the same type, though not depending of a given sensor supplier data. The models considered here are: magnetometer, analog sun sensors, star sensors, GPS receivers and inertial unit.

- Magnetometer

The magnetometer was modelled as a 3 axes fluxgate with alignment (matrix e_{mag}) and scale (K_{mag}) errors. The mathematical model is:

$$_{mag} = K_{mag}\left[e_{mag}(B_T + B_{mag}) + \sigma_{mag}\omega_{mag}\right] \qquad (7)$$

where B_T is the Earth's magnetic vector (in attitude frame coordinates), B_{mag} is the magnetometer bias, due mainly to the presence of magnetic materials in the satellite but also due to electrical inductance, σ_{mag} is the magnetometer standard deviation and ω_{mag} is a Gaussian white noise. Figure 4 presents a result of a simulation where it was considered a magnetometer reading in the x axis.

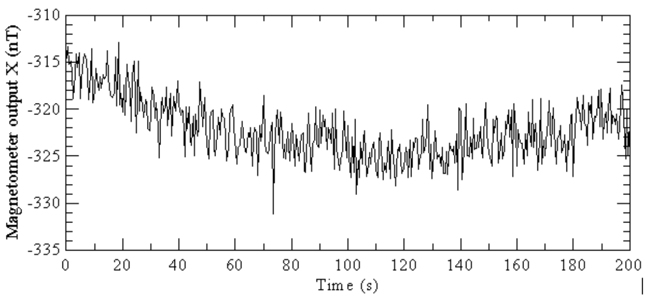

Figure 4 – Magnetometer Simulation in the x direction.

- Analog Sun sensor

These sensors were modelled as a single solar cell whose output voltage s_{css} is proportional to the cosine of the angle between the Sun's position r_{sun} relative to the satellite frame and the cell normal vector N_{css}, that is, s_{css} is given by

$$K_{css}\left[s_{sd}\frac{1}{D_{AU}^2}(N_{css}r_{sun})(1 + \sigma_y\omega_y) + \sigma_a\omega_a\right], \qquad (8)$$

if $N_{css}r_{sun} > 0$, and by

$$K_{css}\sigma_a\omega_a, \qquad (9)$$

if $N_{css}r_{sun} \leq 0$.

In this equation K_{css} is the sensor's gain (in Volts), s_{sd} is the Earth's shadow factor (1 if the satellite is in the Sun's view, 0 if is in the Earth's shadow), D_{AU} is the Earth's Sun distance (in Astronomical Units), to correct for the Sun's power difference between perihelion and aphelion, σ_a and σ_r are the absolute and relative standard deviation of measurements. Finally, ω_r and ω_a are the absolute and relative Gaussian white noise. The simulator allows positioning a solar cell in any already configured appendage. In this case the sensor readings will depend on the appendage rotation angle. Figure 5 depicts a simulation with 8 analog sun sensors oriented in an octahedral configuration. In this simulation the spacecraft is rotating with a period of 100 seconds, and it enters in the Earth's shadow after 185 seconds, approximately.

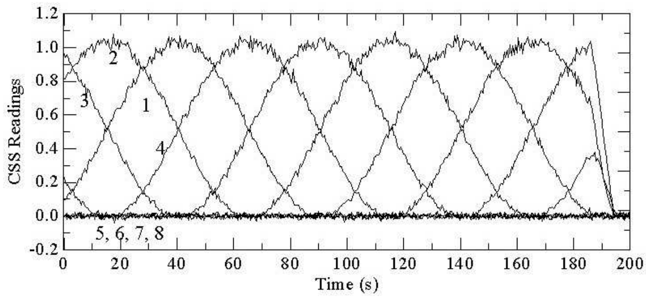

Figure 5 – Simulation of an 8 analog sun sensor readings. Satellite angular velocity is one rotation at each 100 seconds.

- Star sensor (Figure 6)

A Star Sensor equipment gives directly the attitude by measuring the satellite orientation with respect to the stars. By comparing an image acquired by a CCD camera with a star catalog previously stored in the sensor memory, it can provide attitude determination in quaternion with a high degree of accuracy. The model implemented in this package gives the quaternion as a function of the simulated attitude quaternion q, the quaternion that represents the sensor attitude relative to the satellite frame, q_{ss}, and the covariance matrix σ_{star} of the sensor, in sensor reference coordinates:

$$q_{star} = qq_{ss}\begin{pmatrix} 0.5\sigma_{star}\omega_{star} \\ 1 \end{pmatrix} \qquad (10)$$

where ω_{star} is a 3 component vector of a white Gaussian noise. Figure 6 illustrates the behavior of typical Star sensor readings, in the x direction, in terms of Euler angles of a x-y-z rotation instead of quaternion.

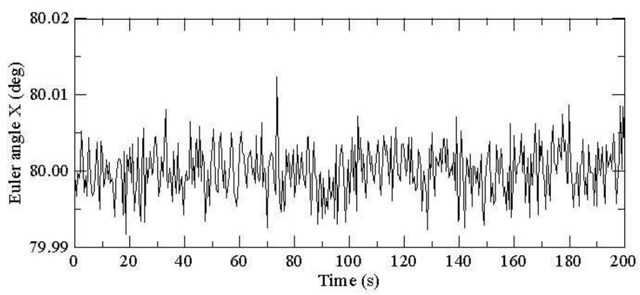

Figure 6 – Star sensor simulation - x axis angle of an Euler x-y-z rotation.

- GPS receiver

The mathematical model of the GPS receiver is based on orbit propagation. It is supposed that the errors presented by this sensor depend on the measuring direction, and probably the error in satellite velocity direction is the highest. So the covariance matrix shall be related to the orbital coordinate system (velocity direction, zenith and normal to orbit plane). The GPS receiver output model is then given by

$$\begin{aligned} p_{GPS} &= K_{GPSP}(P_{in} + R_{orb-in}\sigma_P\omega_{GPSP}) \\ v_{GPS} &= K_{GPSV}(V_{in} + R_{orb-in}\sigma_V\omega_{GPSV}) \end{aligned} \quad (11)$$

where P_{in} and V_{in} are the propagated inertial position and velocity of the satellite, R_{orb-in} is a rotation matrix from orbital frame to inertial coordinates, and ω_{GPSP} and ω_{GPSV} are white Gaussian noise vectors. Figure 7 presents a transformed GPS simulation where the position was converted in Keplerian elements. The orbit semi major axis shows a decreasing behavior due to osculated elements from SGP8 orbit model propagation.

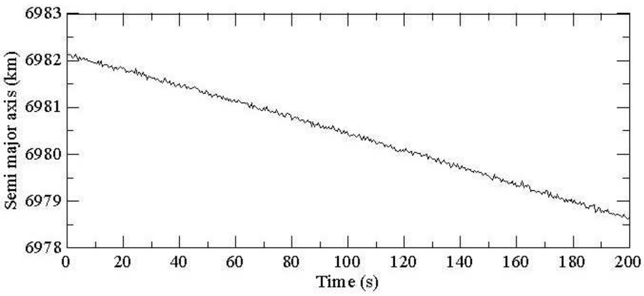

Figure 7 – GPS reading simulation - orbital semi major axis in km.

- Inertial unit

Inertial units are satellite equipments to provide direct angular velocity measuring. They come in different technologies, like mechanical, fiber optics and laser gyros. An inertial unit is composed of one or more gyroscopes that sense velocity in one or two directions. Therefore, the inertial unit combines the individual gyro measuring in a single, reliable and orthogonal angular velocity information. The model employed here does not take into account the number or orientation of the internal gyros, but, instead, it supposes that the inertial unit output is affected by the same errors individual gyros are. So the model is

$$w_{IU} = A_{IU}(I + K_{IU})w + b_{IU} + \sigma_e\omega_e + \eta_v(k), \quad (12)$$

where w_{IU} is the measured angular velocity vector, in equipment frame, A_{IU} is the orientation matrix of the measuring axes relative to satellite frame, w is the simulated angular velocity and K_{IU} is the scale factor error (I is a identity matrix). The other parameters are: b_{IU} is the equipment bias or drift rate (a fixed error) and σ_e represents the covariance matrix of the Gaussian white noise vector ω_e, or random drift. Fiber Optics Gyros (FOG) and also mechanical gyros presents a random walk, a particularly important error. It is a time correlated error $n_v(k)$, where k is the measuring time, and is modelled as a vector

$$n_v(k) = n_v(k-1) + \sigma_v\sqrt{\delta t}\omega_v \quad (13)$$

where δt is the sampling time interval (time between $k-1$ and k), and σ_v is the random walk covariance matrix or the random walk Gaussian white noise ω_v. Figure 8 below shows a simulation of a inertial unit output in the z axis, whose parameters where: scale factor error of 10^{-7} (3 axes), drift rate of 5×10^{-9} radians per second in 3 axes, random drift of 10^{-8} rad/s, and random walk of 4×10^{-10} rad/$s^{3/2}$.

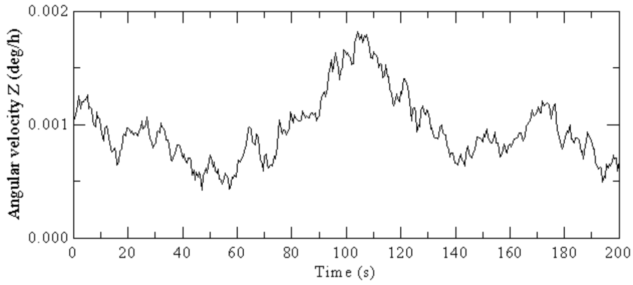

Figure 8 – Simulation of the inertial unit output in the z direction.

4. RESULTS

The simulator shown here helps to easily simulate the attitude and control with high realism and good accuracy any satellite. The satellite can be configured not only in terms of inertia and appendages, but also the sensors and actuators it may have. The results presented in Figures 8 and 9 refers to a fictitious satellite with a diagonal inertia matrix (10, 15, 20 kgm^2 in axes x, y and z, respectively). It was supposed that this satellite has 6 thrusters that give torques in 3 directions. Both position and thruster force vector can be configured in simulator. The arrangement provides thruster of ± 0.8 Nm around each satellite axis. The attitude was supposed known (without sensors). A bang-bang or on-off controller was used as formulated in Wertz (1978) [11], with the objective of eliminate the velocity and the attitude errors (position and rate control):

$$u(t) = -K_p sgn(\theta + K_d\omega), \quad (14)$$

if $|\omega| < 0.1$, or

$$u(t) = -K_p sgn(K_d\omega) \quad (15)$$

otherwise. $K_p = 1$ and $K_d = 6$ are the proportional and derivative gains, and θ, ω, are the Euler angles and angular velocity vector, respectively. The high frequency oscillation seen in Figures 9 left and 10 left are due to control commutation from velocity only to position and rate control. It was also include the effect of a thruster minimum impulse bit (MIB), as seen in the straight lines of Figure 10. No action is taken whenever the attitude remains between those lines - otherwise the generated torque would cause the thruster to fire in positive and negative direction endlessly.

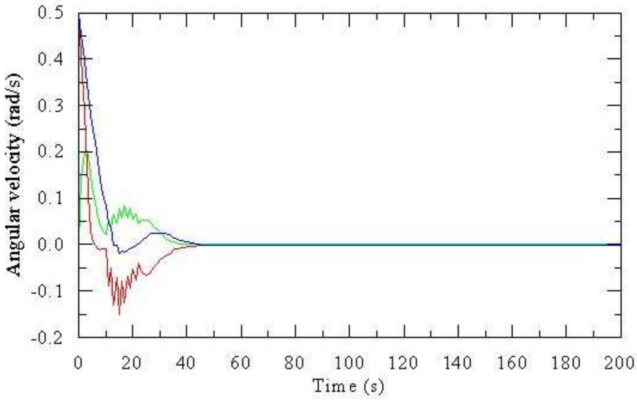

Figure 9 – Simulated angular velocity of a rigid body satellite controlled with thrusters in a dead band on-off control.

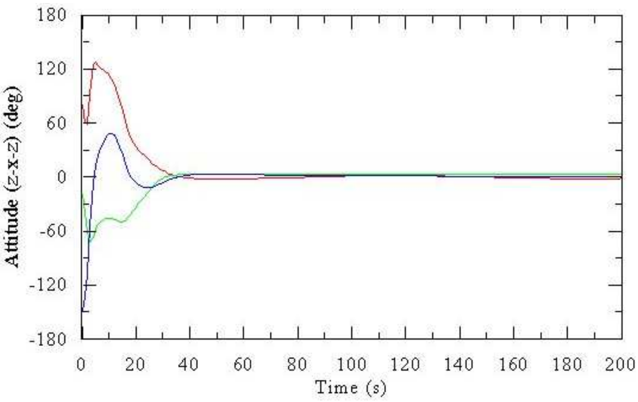

Figure 10 – Simulated attitude of a rigid body satellite controlled with thrusters in a dead band on-off control.

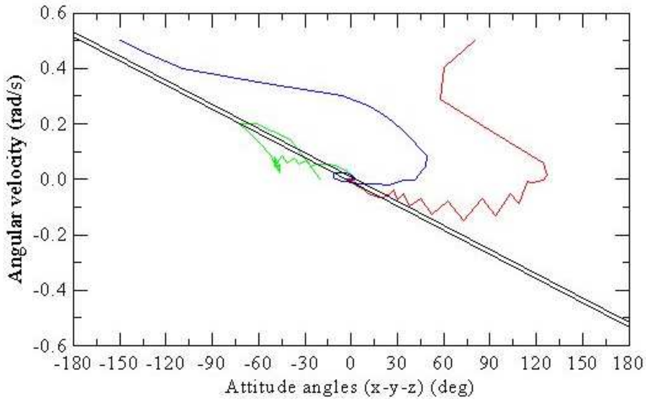

Figure 11 – Phase diagram showing the trajectory of the satellite axes till stabilization. Also shown in diagram is the dead band (straight lines). On right is the activation profile of the thrusters.

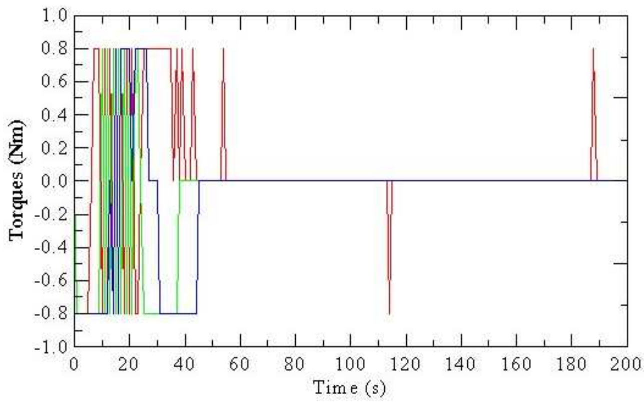

Figure 12 – Activation profile of the thrusters.

5. CONCLUSIONS

The mathematical model of sensors and actuators, and also the simulator presented briefly here are an important step towards capacitating the control team to design, to test and to qualify on-board attitude control. Although the Data Collection Satellites (SCD) launched in the 90's and still fully operational had an attitude control system, they were passively spin stabilized, and the control was performed on-ground, and consists of a series of commands to switch directly and reversely a on-board magnetic air-core coil. The new generation of Brazilian satellites was specified so to have huge pointing and stability requirements, only achievable by means of on-board attitude control. The on-board software is the key point, due to high performance and reliability requirements for the missions. It is also important to Brazil to develop all the technologies necessary to design, code, test and qualify the on-board control software. This technology is kept today by few nations and, not to mention, it is very expensive to buy or to develop, and has many applications outside space area.

The approach adopted during specification of the simulation software allows building, when necessary, a friendly interface with testing operators in a windowed operating system. It should be mentioned that this simulator is not fully tested up to now. It is still being coded, and next step is the inclusion of functions to assure real-time simulations. It is also specified that the simulator and controller shall run in separated computers, with asynchronous communication lines.

ACKNOWLEDGMENTS

The second author thanks CNPq.

REFERENCES

[1] H. Asada, J.-J. E. Slotine, "Robot Analysis and Control", John Wiley and Sons, New York, 1986.

[2] D. Brower, G. M. Clemence, "Methods of celestial mechanics", New York, NY, Academic, 1961.

[3] V. Carrara, "Redes neurais aplicadas ao controle de atitude de satélites com geometria variável", S. J. Campos, INPE, junho, 1997 (INPE-6384-TDI/603).

[4] J.J. Craig, "Introduction to Robotics: Mechanics and Control", (2nd Edition), Addison-Wesley, 1989.

[7] T.C. Flandern, K. F. Pulkkinen, "Low precision formulae for planetary positions", The Astrophysical Journal

Supplement Series, Vol. 41, no. 3, pp. 319-411, Nov., 1979.

[8] F.R. Hoots, R. L. Roehrich, "Models for propagation of NORAD element sets", Aerospace Defense Command, United States Air Force, Spacecraft Report No. 3, Dec., 1980.

[9] H.K. Kuga, V. Carrara, and V. M. Medeiros, "Rotinas auxiliares de mecânica celeste e geração de órbita", São José dos Campos, INPE, julho, 1981 (INPE-2189-RPE/392).

[10] "The Astronomical Almanac 1987 - Supplement to the Astronomical Almanac" - pp. C24, 1987.

[11] J.R. Wertz, "Spacecraft attitude determination and control", London: D. Reidel, 1978 (Astrophysics and Space Science Library).

[12] http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html

[13] http://nssdc.gsfc.nasa.gov/space/model/magnetos/igrf.html