Three-Axis Stabilized Earth Orbiting Spacecraft Simulator

A Senior Project

presented to

the Faculty of the Aerospace Engineering Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Alan F. Ma and Nikola N. Dominikovic

October, 2012

Three-Axis Stabilized Earth Orbiting Spacecraft Simulator

Alan F. Ma[1] and Nikola N. Dominikovic[2]
California Polytechnic State University, San Luis Obispo, California, 93407

This report details the method and results of the program created for simulating an Earth orbiting spacecraft with control actuators and orbital perturbations. The control actuators modeled are reaction thrusters, reaction/momentum wheels, and control moment gyros (CMG). The perturbations modeled were gravity gradient, electromagnetic torques, solar radiation pressure, gravity gradients, third-body effects, Earth oblateness and atmospheric drag. This simulation allows for satellite control in all 6 degrees of freedom for any Earth orbiting spacecraft. Assumptions include rigid body dynamics, no sensor noise, constant spacecraft cross-sectional area, constant coefficient of drag and reflectivity, ignoring the effects due to the moon, moment of inertia doesn't change with a change in mass, and reaction thrusters only produce torque. The results from test trials showed reasonable numbers and system behavior.

Nomenclature

A_r	=	cross-sectional area of the spacecraft in the ram direction (m^2)
A_{sun}	=	cross-sectional area of the spacecraft in the sun (m^2)
a	=	semi-major axis (km)
$\vec{a}_{SR}$	=	acceleration vector due to solar radiation pressure (km·s^{-2})
α	=	reciprocal of the semi-major axis (km^{-1})
$\vec{\alpha}$	=	spacecraft angular acceleration (s^{-2})
β	=	reaction wheel angle from z-axis
C	=	Stumpff function
C_d	=	coefficient of drag
C_r	=	coefficient of reflectivity
cm	=	center of mass (m)
cp	=	center of pressure (m)
$\vec{D}$	=	dipole moment of the spacecraft (J·T^{-1})
E	=	eccentric anomaly (rad)
e	=	eccentricity
$\vec{F}_a$	=	drag force (N)
$\vec{F}_{Thrust}$	=	reaction thruster force vector (N)
f	=	Lagrange coefficient
G	=	gravitational constant (m^3·kg^{-1}·s^{-2}
g	=	Lagrange coefficient
h	=	angular momentum (km^2·s^{-1})
I	=	spacecraft moment of inertia (kg·m^2)
I_{CMG}	=	CMG moment of inertia (kg·m^2)
I_r^{wi}	=	reaction wheel moment of inertia of the ith wheel (kg·m^2)
i	=	inclination (rad)
J_0	=	Julian day number at 0 hr UT (days)
j	=	current density (I·A^{-1})
JD	=	Julian Date (days)
K_x	=	proportional control gain in the x-direction

[1] Undergraduate Student, Aerospace Engineering Department, 1 Grand Avenue San Luis Obispo, CA 93407
[2] Undergraduate Student, Aerospace Engineering Department, 1 Grand Avenue San Luis Obispo, CA 93407

K_y = proportional control gain in the y-direction
K_z = proportional control gain in the z-direction
K_{xd} = derivative control gain in the x-direction
K_{yd} = derivative control gain in the x-direction
K_{zd} = derivative control gain in the x-direction
k = boltzman's constant ($m^2 \cdot kg \cdot s^{-2} \cdot K^1$)
k_0 = variable relating the daily $F_{10.7}$ solar activity to a weighted average
k_1 = variable representing the daily effects of the atmospheric density distribution
k_2 = variable accounting for the semiannual effects
k_3 = variable accounting for the changing atmospheric density with a deviation of the daily $F_{10.7}$ from F_{81}
k_4 = variable representing the dependence on geomagnetic activity index
L = mean longitude (deg)
λ = Magnetic latitude
M = mean anomaly (deg)
$\vec{M}$ = Earth magnetic field (T)
M_e = mass of Earth (kg)
m = mass of spacecraft (kg)
Ω = right ascension of the ascending node (deg)
ω = argument of perigee/perihelion (deg)
□ = longitude of perihelion (deg)
$\vec{\omega}$ = spacecraft angular velocity (s^{-1})
$\dot{\vec{\omega}}$ = spacecraft angular acceleration (s^{-2})
$\vec{\omega}_{CMG}$ = CMG angular velocity (s^{-1})
$\dot{\vec{\omega}}_{CMG}$ = CMG angular acceleration (s^{-2})
ω_{wi} = i^{th} reaction wheel angular velocity (s^{-1})
$\dot{\omega}_{wi}$ = i^{th} reaction wheel angular acceleration (s^{-2})
p_{sr} = solar pressure ($kN \cdot m^{-2}$)
Q = any of the six orbital elements
$[\boldsymbol{Q}]_{\bar{x}X}$ = transformation matrix from perifocal to geocentric equatorial coordinate frame
q_c = command quaternion vector
q_e = error quaternion vector
q_s = spacecraft quaternion vector
R_w = reaction wheel coordinate transformation matrix
$\{\boldsymbol{r}\}_X$ = position vector in geocentric equatorial coordinate frame (km)
$\{\boldsymbol{r}\}_{\bar{x}}$ = position vector in perifocal coordinate frame (km)
$\boldsymbol{r}$ = position vector (km)
$\boldsymbol{r}_0$ = initial position vector (km)
$\vec{r}_{sat/sun}$ = position vector from the satellite to the sun (km)
$\vec{r}_{\oplus/3}$ = position vector of the third body relative to the Earth (km)
$\vec{r}_{sat/3}$ = position vector of the third body relative to the satellite (km)
$\ddot{\vec{r}}_{\oplus/sat}$ = acceleration vector of the satellite relative to Earth ($km \cdot s^{-2}$)
ρ = density ($kg \cdot m^{-3}$)
ρ_n = night-time density profile ($kg \cdot m^{-3}$)
S = Stumpff function
T = transformation matrix from ECI to ECEF
T_0 = number of Julian centuries between J2000 and the date in question
$\vec{T}_a$ = drag torque (N·m)
$\vec{T}_g$ = gravity gradient torque (N·m)
$\vec{T}_{ex}$ = disturbance torque (N·m)
$\vec{T}_m$ = magnetic torque (N·m)
$\vec{T}_{Thrust}$ = reaction thruster torque vector (N·m)
T_x = command torque in the x-direction (N·m)
T_y = command torque in the y-direction (N·m)

T_z	=	command torque in the z-direction (N·m)
Δt	=	change in time (s)
θ	=	true anomaly (rad)
θ_G	=	Greenwich sidereal time (rad)
μ	=	gravitational parameter (km^3·s^{-2})
μ_3	=	gravitational parameter of the third body (km^3·s^{-2})
$V_{1,2}$	=	voltage at points in the electron saturation region (V)
V_r	=	velocity in ram direction (m·s^{-1})
V_{the}	=	thermal velocity (m·s^{-1})
v_{r0}	=	initial radial component of velocity (km·s^{-1})
$\boldsymbol{v}$	=	velocity vector (km·s^{-1})
$\boldsymbol{v}_0$	=	initial velocity vector (km·s^{-1})
$\{\boldsymbol{v}\}_X$	=	velocity vector in geocentric equatorial coordinate frame (km·s^{-1})
$\{\boldsymbol{v}\}_{\bar{x}}$	=	velocity vector in perifocal coordinate frame (km·s^{-1})
χ	=	universal anomaly (km$^{1/2}$)

I. Introduction

A MATLAB program, 6 DOF Satellite Simulator, was created to simulate a three-axis stabilized spacecraft orbiting the Earth. This two-body (or more) system simulates the effects of gravity gradient, solar radiation pressure, atmospheric drag, Earth oblateness, third-body effects and electromagnetic torques. Gravity gradient torques are caused when a spacecraft's center of gravity is not aligned with its center of mass with respect to the local vertical. Solar radiation pressure is the pressure due to electromagnetic radiation from the sun. The photons from the sun's beam transfer momentum from itself to the surfaces of a spacecraft that it comes in contact with thus creating pressure. Solar radiation pressure is one of the more difficult perturbations to model. Its significance changes with the altitude of the spacecraft; in low Earth orbit (LEO) it is almost negligible but in geosynchronous orbit (GEO) it is one of the main sources of perturbations. Atmospheric drag is due to the spacecraft colliding with particles in its orbit. It can cause both a torque and a force onto the spacecraft which will cause the spacecraft to both rotate and translate. Atmospheric drag is only prevalent in low Earth orbit, so the assumption was made that all atmospheric drag above an altitude of 1500 km is negligible and will be ignored. Earth oblateness is due to the Earth not being a perfect sphere, thus gravity acts differently on a spacecraft depending on where the spacecraft is located relative to the Earth. Third body perturbation is the gravitational attraction, in this case, of a satellite with, for example, the sun where the Earth is the second body. And electromagnetic torques are due to residual magnetic moments of the spacecraft. These residual moments can range anywhere from .1-20 A·m^2. When a spacecraft's residual moment is not aligned with a local magnetic field, it experiences an electromagnetic torque that attempts to align the magnet moment to the local field. The Earth's magnetic field is complex, asymmetric and varies, however for use in the ADCS design process, it is usually sufficient to model Earth's magnetic field as a dipole and to determine the maximum possible value of the magnetic torque for a spacecrafts altitude.[1]

Control actuators were modeled under these conditions using a set of and/or a combination of reaction thrusters, reaction/momentum wheels, and control moment gyros (CMG). The program, in the form of a graphical user interface (GUI) accepts user defined elements of the spacecraft (i.e. mass moment of inertia, classical orbital elements, control actuators, duration, etc.) to generate a mission profile that includes, but not limited to disturbance forces and moments on the spacecraft, performance of the control actuators, and orbital profile.

The simulation for a six degree of freedom satellite requires a lot of different information for accuracy; from the control method and the control type to the orbital propagator and the disturbance modeling, with each method being dependent on the other. The reason behind the centralization of all these methods is to simplify and more importantly, quicken the design process of an Earth orbiting satellite. The goal of this program, 6 DOF Satellite Simulator, is to achieve exactly that.

II. Orbital Determination

Before calculating the position of a spacecraft relative to the Earth, a time system was established using Julian Date (JD). JD is the continuous count of the number of days since Greenwich noon (12:00 UT) on January 1, 4712. It is also the universally adopted solution for astronomical problems.[2] JD was used due to its simplicity of using continuous count of days as opposed to dealing with months, days, hours, seconds, and minutes. JD was obtained by using the following equations,[3]

$$J_0 = 367y - floor\left\{\frac{7\left[y + floor\left(\frac{m+9}{12}\right)\right]}{4}\right\} + floor\left(\frac{275m}{9}\right) + d + 1{,}721{,}013.5 \tag{1}$$

$$JD = J_0 + \frac{UT}{24} \tag{2}$$

where J_0 is the Julian day number at 0 hr UT in days, *y* is the year, *m* is the month, *d* is the day, *floor* is the MATLAB command that rounds to negative infinity, *UT* is universal time in hours, and *JD* is Julian Date in days. After calculating for JD, planetary ephemeris was obtained with the planetary orbital elements and their centennial rates found in Table 8.1 in Curtis.[3] The following equations were used to calculate the planetary orbital elements,

$$T_0 = \frac{JD - 2{,}451{,}545}{36{,}525} \tag{3}$$

$$Q = Q_0 + \dot{Q}T_0 \tag{4}$$

$$h = \sqrt{\mu a(1 - e^2)} \tag{5}$$

$$\omega = \varpi - \Omega \tag{6}$$

$$M = L - \varpi \tag{7}$$

where T_0 is the number of Julian centuries between J2000 and the date in question, Q is any one of the six planetary orbital elements in Table 8.1 of Curtis, h is the angular momentum in $km^2 \cdot s^{-1}$, μ is the gravitational parameter of the sun of 1.327×10^{11} $km^3 \cdot s^{-2}$, *a* is the semi-major axis in km, *e* is the eccentricity, ω is the argument of perihelion in degrees, □ is the longitude of perihelion in degrees, Ω is the right ascension of the ascending node (RAAN) in degrees, M is the mean anomaly in degrees, and L is the mean longitude in degrees. It should be noted that all angular quantities were adjusted to lie between 0° and 360°. With eccentricity and mean anomaly, the eccentric anomaly and true anomaly were calculated with the following equations from Curtis,[3]

$$E = \begin{cases} M + e/2 & (M < \pi) \\ M - e/2 & (M > \pi) \end{cases} \tag{8}$$

$$f(E_i) = E_i - e\sin E_i - M \tag{9}$$

$$f'(E_i) = 1 - e\cos E_i \tag{10}$$

$$ratio_i = f(E_i)/f'(E_i) \tag{11}$$

$$E_{i+1} = E_i - ratio_i \tag{12}$$

$$\theta = 2\tan^{-1}\frac{\tan\frac{e}{2}}{\sqrt{\frac{1-e}{1+e}}} \tag{13}$$

where E is the eccentric anomaly in radians, M is the mean anomaly in radians, θ is true anomaly in radians. It should be noted that Eqs. (9-12) are in an iterative loop to solve for E, where the loop breaks once $ratio_i$, or the tolerance, of 10^{-6} is reached. Equation (8) was used as an initial estimate for the eccentric anomaly to input into the iterative loop. Once the approximate eccentric anomaly (where the accuracy was chosen by the tolerance) is solved for, it is then used in Eq. (13) to obtain the true anomaly. With all of the planetary orbital elements, the state vectors of the planets were calculated using the following equations,[3]

$$\{\boldsymbol{r}\}_{\bar{x}} = \frac{h^2}{\mu}\frac{1}{1 + e\cos\theta}\begin{Bmatrix}\cos\theta \\ \sin\theta \\ 0\end{Bmatrix} \tag{14}$$

$$\{\boldsymbol{v}\}_{\bar{x}} = \frac{\mu}{h}\begin{Bmatrix} -\sin\theta \\ e+\cos\theta \\ 0 \end{Bmatrix} \tag{15}$$

$$[\boldsymbol{Q}]_{\bar{x}X} = \begin{bmatrix} -\sin\Omega\cos i\sin\omega + \cos\Omega\cos\omega & -\sin\Omega\cos i\cos\omega - \cos\Omega\sin\omega & \sin\Omega\sin i \\ \cos\Omega\cos i\sin\omega + \sin\Omega\cos\omega & \cos\Omega\cos i\cos\omega - \sin\Omega\sin\omega & -\cos\Omega\sin i \\ \sin i\sin\omega & \sin i\cos\omega & \cos i \end{bmatrix} \tag{16}$$

$$\{\boldsymbol{r}\}_X = [\boldsymbol{Q}]_{\bar{x}X}\{\boldsymbol{r}\}_{\bar{x}} \tag{17}$$

$$\{\boldsymbol{v}\}_X = [\boldsymbol{Q}]_{\bar{x}X}\{\boldsymbol{v}\}_{\bar{x}} \tag{18}$$

where $\{\boldsymbol{r}\}_{\bar{x}}$ is the position vector in the perifocal frame in km, $\{\boldsymbol{v}\}_{\bar{x}}$ is the velocity vector in the perifocal frame in km·s^{-1}, $[\boldsymbol{Q}]_{\bar{x}X}$ is the transformation matrix from perifocal to geocentric equatorial frame, i is the inclination in radians, Ω is RAAN in radians, ω is the argument of perigee/perihelion in radians, $\{\boldsymbol{r}\}_X$ is the position vector in the geocentric equatorial frame in km, and $\{\boldsymbol{v}\}_X$ is the velocity vector in the geocentric equatorial frame in km·s^{-1}. It should be noted that to obtain the state vectors in a heliocentric frame for planetary ephemeris, a μ of 1.327×10^{11} km^3·s^{-2} was used and ω is the argument of perihelion. Equations (14-18) were also used to obtain the state vectors for a satellite in orbit around the Earth, where a μ of 398600 km^3·s^{-2} was used and ω is the argument of perigee. To obtain the state vectors of a satellite in orbit after a change in time and ignoring orbital perturbations, universal variables with Stumpff functions were used. First, the universal anomaly, χ, was obtained with the following equations in an iterative loop,[3]

$$r_0 = \sqrt{\boldsymbol{r}_0 \cdot \boldsymbol{r}_0} \tag{19}$$

$$v_0 = \sqrt{\boldsymbol{v}_0 \cdot \boldsymbol{v}_0} \tag{20}$$

$$v_{r0} = \frac{\boldsymbol{r}_0 \cdot \boldsymbol{v}_0}{r_0} \tag{21}$$

$$\alpha = \frac{2}{r_0} + \frac{v_0^2}{\mu} \tag{22}$$

$$\chi_0 = \sqrt{\mu}|\alpha|\Delta t \tag{23}$$

$$z_i = \alpha\chi_i^2 \tag{24}$$

$$S(z) = \begin{cases} \dfrac{\sqrt{z}-\sin\sqrt{z}}{(\sqrt{z})^3} & (z>0) \\ \dfrac{\sinh\sqrt{-z}-\sqrt{-z}}{(\sqrt{-z})^3} & (z<0) \\ \dfrac{1}{6} & (z=0) \end{cases} \tag{25}$$

$$(z) = \begin{cases} \dfrac{1-\cos\sqrt{z}}{z} & (z>0) \\ \dfrac{\cosh\sqrt{-z}-1}{-z} & (z<0) \\ \dfrac{1}{2} & (z=0) \end{cases} \tag{26}$$

$$f(\chi_i) = \frac{r_0 v_{r0}}{\sqrt{\mu}}\chi_i^2 C(z_i) + (1-\alpha r_0)\chi_i^3 S(z_i) + r_0\chi_i - \sqrt{\mu}\Delta t \tag{27}$$

$$f'(\chi_i) = \frac{r_0 v_{r0}}{\sqrt{\mu}}\chi_i[1-\alpha\chi_i^2 S(z_i)] + (1-\alpha r_0)\chi_i^2 C(z_i) + r_0 \tag{28}$$

$$ratio_i = f(\chi_i)/f'(\chi_i) \tag{29}$$

$$\chi_{i+1} = \chi_i - ratio_i \tag{30}$$

where r_0 is the initial position in km, $\mathbf{r}_0$ is the initial position vector in km, v_0 is the initial velocity in km·s^{-1}, $\mathbf{v}_0$ is the initial velocity vector in km·s^{-1}, v_{r0} is the initial radial component of velocity in km·s^{-1}, α is the reciprocal of the semi-major axis in km^{-1}, χ is the universal anomaly in km$^{1/2}$, S and C are Stumpff functions. It should be noted that the above equations are in a loop similar to Eqs. (8-12). Once χ is obtained, it is then used in the following equations to calculate for the new state vectors,

$$f = 1 - \frac{\chi^2}{r_0} C(\alpha\chi^2) \tag{31}$$

$$g = \Delta t - \frac{1}{\sqrt{\mu}} \chi^3 S(\alpha\chi^2) \tag{32}$$

$$\dot{f} = \frac{\sqrt{\mu}}{r r_0} [\alpha\chi^3 S(\alpha\chi^2) - \chi] \tag{33}$$

$$\dot{g} = 1 - \frac{\chi^2}{r} C(\alpha\chi^2) \tag{34}$$

$$\boldsymbol{r} = f\boldsymbol{r}_0 + g\boldsymbol{v}_0 \tag{35}$$

$$\boldsymbol{v} = \dot{f}\boldsymbol{r}_0 + \dot{g}\boldsymbol{v}_0 \tag{36}$$

where f and g are Lagrange coefficients in terms of the universal anomaly, Δt is the change in time in seconds, **r** is the new position vector in km, and **v** is the new velocity vector in km·s^{-1}. It should be noted that Δt needs to be in small intervals relative to the period of the orbit, otherwise there would be a significant inaccuracy in the new state vectors. To propagate the orbit of a satellite accounting for perturbations, Encke's formulation was used. In Encke's method, the orbit is propagated by integrating the the differences between the osculating and the perturbed orbit. This process continues until the tolerance of about 1% is met, where the osculating orbit is re-initialized.[4]

III. Orbital Perturbations

A. Atmospheric Drag

Atmospheric drag is the first perturbation that the spacecraft will undergo The simplified equation to the atmospheric drag force and torque are given by Space Mission Engineering: The New SMAD and are[1]

$$\vec{T}_a = \frac{1}{2}\rho C_d A_r {V_r}^2 \begin{bmatrix} cp_x - cm_x \\ 0 \\ cp_z - cp_z \end{bmatrix} \tag{37}$$

$$\vec{F}_a = \frac{1}{2}\rho C_d A_r {V_r}^2 \tag{38}$$

where T_a and F_a is the atmospheric drag torque and force, respectively, ρ is the atmospheric density in kg·m^{-3}, C_d is the drag coefficient of the spacecraft, A_r is the area of the spacecraft in the ram direction in m^2 and V_r is the velocity in the ram direction in km·s^{-1}. cp_x and cp_z are the center of pressure in the x and z direction respectively and, cm_x and cm_z are the center of mass in the x and z direction respectively. It should be noted that the cp and the cm need to be in terms of the Local-Vertical Local-Horizontal (LVLH) reference frame and not the body frame. A lot of these parameters change as the spacecraft rotates so it is assumed that transient rotations are not vastly different than the desired controlled attitude and is ignored. Also ignored are any small deviations from the desired controlled attitude due to controller type. So the only parameters that would be inputted are those at the controlled attitude.

To model the atmospheric density, the Russian GOST model was used. This model was constructed from observations of the orbital motion of Russian Cosmos satellites. The density, ρ in kg·m^{-3}, was calculated with the following equation,[4]

$$\rho = \rho_n k_0 k_1 k_2 k_3 k_4 \tag{39}$$

where ρ_n is the night-time density profile in kg·m^{-3}, k_0 relates the daily $F_{10.7}$ solar activity to a weighted average, k_1 represents the daily effects of the atmospheric density distribution, k_2 accounts for the semiannual effects, k_3 accounts for the changing atmospheric density with a deviation of the daily $F_{10.7}$ from F_{81}, and k_4 represents the dependence on geomagnetic activity index. The equations and tables used for this model can be found in Appendix B.2 in Fundamentals of Astrodynamics and Applications.[4] This model is only valid for an altitude range of 120-1500 km.

B. Electromagnetic Torques

The second orbital perturbation is the electromagnetic torques. Electromagnetic forces are very small and complicated and are only notable on spacecraft with long conductive tether systems. For this reason electromagnetic forces will be ignored. The simplified equation for electromagnetic torques is given by[1]

$$\vec{T}_m = \vec{D} \times \left(\frac{\vec{M}}{r^3}\lambda\right) \tag{40}$$

where T_m is the electromagnetic torque, D is the dipole moment of the spacecraft, M is the magnetic field of Earth, r is the distance of the spacecraft to the center of Earth, and λ is a unitless function of the magnetic latitude that ranges from 1 at the magnetic equator and 2 at the magnetic poles.

C. Solar Radiation Pressure

To correctly and accurately model SRP, a precise location of the Sun, spacecraft attitude, time-varying cross-sectional area, and time-varying coefficient of reflectivity need to be known. In this case, the cross-sectional area and coefficient of reflectivity are assumed to be constant due to the complexity of accounting for those variables. Eclipses are accounted for in the spacecrafts orbit to determine when SRP is on or off. To calculate SRP, the following equation was used[4]

$$\vec{a}_{SR} = -\frac{p_{sr} C_r A_{sun}}{m} \frac{\vec{r}_{sat/sun}}{|\vec{r}_{sat/sun}|} \tag{41}$$

where $\vec{a}_{SR}$ is the acceleration due to solar radiation pressure in km·s^{-2}, p_{sr} is the solar pressure of 4.57×10^{-9} kN·m^{-2} where the solar flux is 1367 W·m^{-2}, C_r is the coefficient of reflectivity, A_{sun} is the cross-sectional area of the spacecraft in the sun in m^2, m is the mass of the spacecraft in kg, and $\vec{r}_{sat/sun}$ is the position vector from the satellite to the sun in km. To determine if the satellite is in an eclipse, Algorithm 34 from Fundamentals of Astrodynamics and Applications[4] was used. It uses simple geometry by examining the satellite's vertical and horizontal distances from the Earth-Sun line. It was assumed that if the satellite was in either the penumbra or umbra region, it is in a total eclipse.

D. Third Body

Similarly to SRP the effects of third body perturbations on a satellite are, more or less significant with respect to the altitude of a satellite. In this case, third body perturbations due to the Moon are ignored, the mass of the satellite is negligible, and all planets up to Neptune (including the Sun) are considered. To calculate this acceleration, the following equation was used[4]

$$\ddot{\vec{r}}_{\oplus/sat} = -\mu_3 \left(\frac{\vec{r}_{sat/3}}{r_{sat/3}^3} - \frac{\vec{r}_{\oplus/3}}{r_{\oplus/3}^3}\right) \tag{42}$$

where $\ddot{\vec{r}}_{\oplus/sat}$ is the acceleration vector of the satellite relative to Earth in km·s^{-2}, μ_3 is the gravitational parameter of the third body in km^3·s^{-2}, $\vec{r}_{sat/3}$ is the position vector of the third body relative to the satellite in km, and $\vec{r}_{\oplus/3}$ is the position vector of the third body relative to the Earth in km. Equation 42 is used to find the third body acceleration of each individual planet, the sum is taken to obtain the total acceleration.

E. Earth Oblateness

Since the Earth is not a perfect sphere, perturbing accelerations due to the Earth varies with where the spacecraft is located at. Using the Legendre functions in Table 8-2, the gravitational coefficients in Table D-1/2, and equations from page 548 from Fundamentals of Astrodynamics and Applications[4], the accelerations were modeled. Zonal, sectorial, and tesseral harmonics are accounted for from a Legendre function of $P_{2,0}$ (J_2) to $P_{4,4}$. Zonal harmonics, such as J_2 accounts for most of the gravitational effects due to oblateness; it represents bands of latitude. Sectorial harmonics and tesseral harmonics represents bands of longitude and specific regions of the Earth (in a checkered pattern), respectively.[4] To calculate the acceleration, the position and velocity vector of the satellite had to be converted from an Earth-Centered Inertial (ECI) coordinate frame to an Earth-Centered, Earth-Fixed (ECEF) coordinate frame using the following equations,[4,5]

$$T = \begin{bmatrix} cos(\theta_G) & sin(\theta_G) & 0 \\ -sin(\theta_G) & cos(\theta_G) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{43}$$

$$\dot{T} = \begin{bmatrix} -\omega_E sin(\theta_G) & \omega_E cos(\theta_G) & 0 \\ -\omega_E cos(\theta_G) & -\omega_E sin(\theta_G) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{44}$$

$$\ddot{T} = \begin{bmatrix} -\omega_E^2 cos(\theta_G) & -\omega_E^2 sin(\theta_G) & 0 \\ \omega_E^2 sin(\theta_G) & -\omega_E^2 cos(\theta_G) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{45}$$

$$\vec{r}_{ECEF} = T\vec{r}_{ECI} \tag{46}$$

$$\vec{v}_{ECEF} = T\vec{v}_{ECI} + \dot{T}\vec{r}_{ECI} \tag{47}$$

$$\vec{a}_{ECEF} = T\vec{a}_{ECI} + 2\dot{T}\vec{v}_{ECI} + \ddot{T}\vec{r}_{ECI} \tag{48}$$

where T is the transformation matrix from ECI to ECEF, θ_G is the Greenwich sidereal time in rad, $\omega_E$is the inertial rotation rate of the Earth of 7.2921×10^{-5} in rad/s.

F. Gravity Gradient

The most general form of the equation in an inverse square gravity field is given by[6]

$$\vec{T}_g = \frac{3GM_e}{R^5} \times I\vec{r} \tag{49}$$

where T_g is the gravity torque, G is the gravitational constant, M_e is the mass of the Earth, r is the distant between the center of the earth and the center of the spacecraft. I is the moment of the inertia in the LVLH reference frame and might not be a principle inertia matrix.

IV. Control Actuators Dynamics

A. Reaction/Momentum Wheels

The first type of control system that can be modeled onto the spacecraft are reaction or momentum wheels. The difference between a reaction wheel and a momentum wheel is that a reaction wheel only acts to remove disturbances from the spacecraft while a momentum wheel can hold a significant amount of momentum to make it harder for disturbances to affect the system while also removing the disturbances. Physically they are identical, only difference being momentum wheels are typically larger. The program allows you to model both, but for now they will be referred to as only reaction wheels.

The first step is to determine how the reaction wheels will be placed in the spacecraft. A common setup for reaction wheels is a four wheel pyramidal setup as seen in Fig. 1, where all four wheels are in the same plane but are angled at a specific β angle. This is the setup that the program uses. The β angle can be changed as well as all the

parameters of the wheel but a few assumptions cannot be changed. Such as the wheels and the angle β are identical and they are aligned to a principle axis.

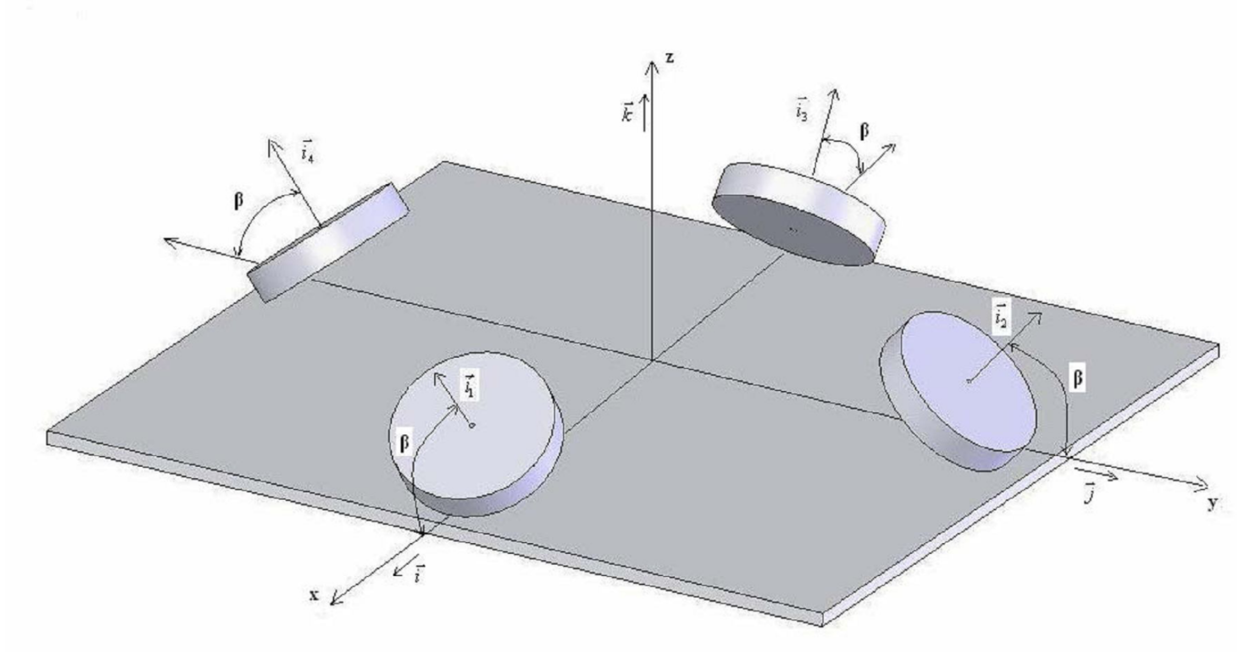

Figure 1. Pyramidal reaction wheel setup.

The transformation matrices of the reaction wheels from the wheel frame to the body frame are given by[7]

$$R_{w1} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \tag{50}$$

$$R_{w2} = \begin{bmatrix} 0 & 1 & 0 \\ \cos\beta & 1 & -\sin\beta \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \tag{51}$$

$$R_{w3} = \begin{bmatrix} -\cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \tag{52}$$

$$R_{w4} = \begin{bmatrix} 0 & 1 & 0 \\ -\cos\beta & 1 & \sin\beta \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \tag{53}$$

where R_w are the wheel rotation matrices. The angular momentum of the wheels needs to be equal and opposite to the momentum of the spacecraft. This relation is described by[7]

$$-\sum_{i=1}^{4}\left(R_{wi}\begin{bmatrix} I_r^{wi}\dot{\omega}_{wi} \\ 0 \\ 0 \end{bmatrix} + \vec{\omega}\times R_{wi}\begin{bmatrix} I_r^{wi}\omega_{wi} \\ 0 \\ 0 \end{bmatrix}\right) = I\times\dot{\vec{\omega}} + \vec{\omega}\times I\vec{\omega} \tag{54}$$

where $\vec{\omega}$ is the angular velocity of the spacecraft, $\dot{\vec{\omega}}$ is the angular acceleration of the spacecraft, I is the body frame inertia matrix of the spacecraft, I_r^{wi} is the inertia of the i'th wheel about its spin axis, ω_{wi} is the angular velocity of the i'th wheel and $\dot{\omega}_{wi}$ is the angular acceleration of the i'th wheel. The left hand side of the equation is the wheel momentum while the right hand side is the spacecraft momentum. The right hand side is also equal to the commanded torques and thus the equation can be rewritten as[7]

$$-\sum_{i=1}^{4}\left(R_{wi}\begin{bmatrix}I_r^{wi}\dot{\omega}_{wi}\\0\\0\end{bmatrix}+\vec{\omega}\times R_{wi}\begin{bmatrix}I_r^{wi}\omega_{wi}\\0\\0\end{bmatrix}\right)=\begin{bmatrix}T_x\\T_y\\T_z\end{bmatrix} \tag{55}$$

where T_x, T_y and T_z are the command torques in the x, y and z directions. The angular velocity of the wheel, ω_{wi}, is a known quantity that can be directly measured onboard and therefore can be subtracted from the commanded torque to obtain[7]

$$\begin{bmatrix}T_x\\T_y\\T_z\end{bmatrix}+\sum_{i=1}^{4}\left(\vec{\omega}\times R_{wi}\begin{bmatrix}I_r^{wi}\omega_{wi}\\0\\0\end{bmatrix}\right)=-\sum_{i=1}^{4}\left(R_{wi}\begin{bmatrix}I_r^{wi}\dot{\omega}_{wi}\\0\\0\end{bmatrix}\right) \tag{56}$$

Now, we can redefine the left hand side as $\begin{bmatrix}\hat{T}_x & \hat{T}_y & \hat{T}_z\end{bmatrix}^T$, and $[I_r^{w1}\dot{\omega}_{w1} \;\; I_r^{w2}\dot{\omega}_{w2} \;\; I_r^{w3}\dot{\omega}_{w3} \;\; I_r^{w4}\dot{\omega}_{w4}]^T$ as $[T_1 \;\; T_2 \;\; T_3 \;\; T_4]^T$. Using Eqs. (50-53) we can rewrite and expand Eq. (56) as[7]

$$\begin{bmatrix}\hat{T}_x\\\hat{T}_y\\\hat{T}_z\end{bmatrix}=\begin{bmatrix}\cos\beta & 0 & -\cos\beta & 0\\0 & \cos\beta & 0 & -\cos\beta\\\sin\beta & \sin\beta & \sin\beta & \sin\beta\end{bmatrix}\begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix} \tag{57}$$

The reaction wheel torques are desired, but the matrix relating the desired torques to the reaction wheel torques is not square and therefore non-invertible. Sidi[8] describes that one possible way to distribute the torques between the four reaction wheels is to define $\begin{bmatrix}\hat{T}_x/\cos\beta & \hat{T}_y/\cos\beta & \hat{T}_z/\sin\beta\end{bmatrix}^T$ as $\begin{bmatrix}\hat{T}'_x & \hat{T}'_y & \hat{T}'_z\end{bmatrix}^T$ so that[7]

$$\begin{bmatrix}\hat{T}'_x\\\hat{T}'_y\\\hat{T}'_z\end{bmatrix}=\begin{bmatrix}1 & 0 & -1 & 0\\0 & 1 & 0 & -1\\1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix}=A_w\begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix} \tag{58}$$

The matrix A_w is still not square but by taking the right pseudo-inverse of A_w the equation becomes[7]

$$\begin{bmatrix}T_1\\T_2\\T_3\\T_4\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1 & 0 & 1/2\\0 & 1 & 1/2\\-1 & 0 & 1/2\\0 & -1 & 1/2\end{bmatrix}\begin{bmatrix}\hat{T}'_x\\\hat{T}'_y\\\hat{T}'_z\end{bmatrix} \tag{59}$$

Now we have the equation that obtains the torques on all four wheels given the command torques of the spacecraft. The next step is to determine how to control the spacecraft and how to obtain the command torques. To control the spacecraft a quaternion error controller was used because it does not suffer from gimbal locks as would an Euler control method and performs well with large commanded angles. The quaternion error is defined as[8]

$$\vec{q}_e=\begin{bmatrix}q_{c4} & q_{c3} & -q_{c2} & q_{c1}\\-q_{c3} & q_{c4} & q_{c1} & q_{c2}\\q_{c2} & -q_{c1} & q_{c4} & q_{c3}\\-q_{c1} & -q_{c2} & -q_{c3} & q_{c4}\end{bmatrix}\begin{bmatrix}-q_{s1}\\-q_{s2}\\-q_{s3}\\q_{s4}\end{bmatrix} \tag{60}$$

where q_e is the quaternion error matrix, q_c is the commanded quaternion vector and q_s is the spacecraft quaternion vector. Finally the control law to obtain the command torques is given by[8]

$$\begin{aligned}T_x &= -2K_x q_{1e} q_{4e} + K_{xd} p\\T_y &= -2K_y q_{2e} q_{4e} + K_{yd} q\\T_z &= -2K_z q_{3e} q_{4e} + K_{zd} r\end{aligned} \tag{61}$$

where K are control gains and, p, q and r are the body roll rates in x, y and z direction, respectively. Finally we have the dynamics of the spacecraft where we want to find the acceleration of the spacecraft, which is used to update the

location of the spacecraft, the spacecraft quaternion, reaction wheel speed/saturation and etc. The spacecraft acceleration equation come from Curtis and is given by[3]

$$\vec{\alpha} = I_s^{-1}\left\{\vec{T}_{ex} - \left(\vec{\omega} \times (I_s\vec{\omega})\right) - \sum_{i=1}^{4}\left(\vec{\omega} \times R_{wi}\begin{bmatrix} I_s^{wi}\omega_{wi} \\ 0 \\ 0 \end{bmatrix}\right) - \sum_{i=1}^{4}\left(R_{wi}\begin{bmatrix} I_s^{wi}\alpha_{wi} \\ 0 \\ 0 \end{bmatrix}\right)\right\} \tag{62}$$

where $\vec{\alpha}$ is the angular acceleration of the spacecraft, $\vec{T}_{ex}$ are the external torques (in body frame) and α_{wi} is the acceleration of the i'th wheel. From the acceleration the body rates can easily be found by taking a derivative.

B. Control Moment Gyros

The second type of control system that can be modeled onto the spacecraft is a control momentum gyro (CMG). The CMG is similar to a reaction wheel but with the main difference of the wheel being attached to frame with two gimbals which allows the wheel to rotate to a desired direction. CMG's are also usually bigger and are also typically faster and are often used in large spacecraft or spacecraft that require large maneuvers.

The setup used in this simulation is a single CMG at the center of the spacecraft. The setup can be seen in Fig. 2.

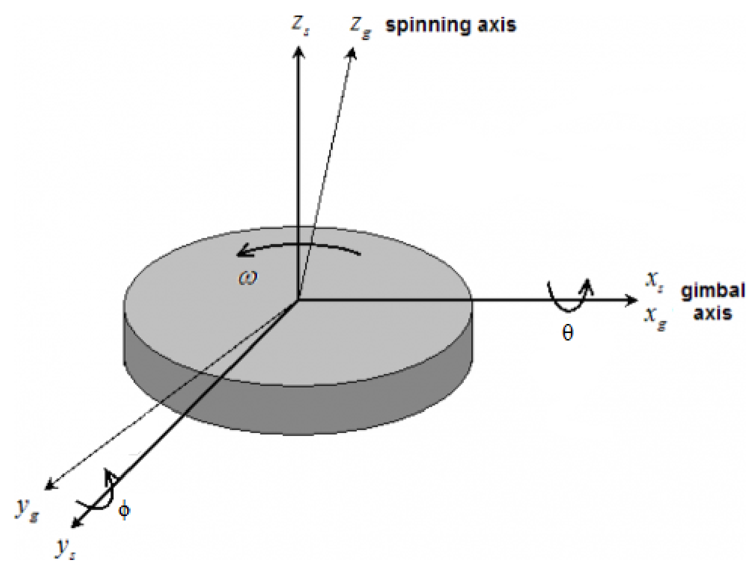

Figure 2. Control moment gyro setup.

θ and □ are the gimbal angles and ω is the angular velocity of the wheel. As the CMG is aligned the body axis no coordinate transformation is necessary. The angular momentum of the CMG needs to be equal and opposite to the angular momentum of the spacecraft in order to stabilize it and the relation is described by[8]

$$-I_{CMG} \times \dot{\vec{\omega}}_{CMG} + \vec{\omega}_{CMG} \times I_{CMG}\vec{\omega}_{CMG} = I \times \dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} \tag{63}$$

where $\vec{\omega}$ is the angular velocity of the spacecraft, $\dot{\vec{\omega}}$ is the angular acceleration of the spacecraft, I is the body frame inertia matrix of the spacecraft, I_{CMG} is the inertia of the CMG, $\vec{\omega}_{CMG}$ is the angular velocity of the CMG and finally $\dot{\vec{\omega}}_{CMG}$ is the angular acceleration of the CMG. Once again the right hand side is equal to the command Torque generated by the control thus Eq. (63) can be rewritten as[8]

$$-I_{CMG} \times \dot{\vec{\omega}}_{CMG} + \vec{\omega}_{CMG} \times I_{CMG}\vec{\omega}_{CMG} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{64}$$

where T_x, T_y and T_z are the command torques in the x, y and z directions. The angular velocity of the wheel, $\vec{\omega}_{CMG}$, is a known quantity that can be directly measured onboard and therefore can be subtracted from the commanded torque to obtain[8]

$$-I_{CMG} \times \dot{\vec{\omega}}_{CMG} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} - \vec{\omega}_{CMG} \times I_{CMG} \vec{\omega}_{CMG} \tag{65}$$

The right hand side can be rewritten as $\begin{bmatrix} \hat{T}_x & \hat{T}_y & \hat{T}_z \end{bmatrix}^T$ while the left hand side is simply the torque of the CMG so the equation becomes[8]

$$\begin{bmatrix} \hat{T}_x \\ \hat{T}_y \\ \hat{T}_z \end{bmatrix} = -\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}_{CMG} \tag{66}$$

Where $[T_x \quad T_y \quad T_z]_{CMG}^T$ is the torque of the CMG. Now we have the equation that obtains the torque for the CMG given the command torques of the spacecraft. The torque of the CMG can be rewritten to include the gimbal angles and the torque of the actual wheel.[3]

$$\begin{bmatrix} \hat{T}_x \\ \hat{T}_y \\ \hat{T}_z \end{bmatrix} = -\begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} T_{CMG_\omega} \tag{67}$$

where T_{CMG_ω} is the torque of the wheel in the CMG system. The CMG uses the same controller and quaternion feedback system as the reaction wheels as seen in Eq. (60) and (61). Finally the dynamics of the spacecraft where we want to find the acceleration of the spacecraft can be known, which is used to update the location of the spacecraft, the spacecraft quaternion, wheel speed/saturation and etc. The spacecraft acceleration equation come from Curtis and is given by[3]

$$\vec{\alpha} = I_s^{-1}\left\{\vec{T}_{ex} - \left(\vec{\omega} \times (I_s\vec{\omega})\right) - I_{CMG} \times \dot{\vec{\omega}}_{CMG} - \vec{\omega}_{CMG} \times I_{CMG}\vec{\omega}_{CMG}\right\} \tag{68}$$

where $\vec{\alpha}$ is the angular acceleration of the spacecraft and $\vec{T}_{ex}$ are the external torques (in body frame). From the acceleration the body rates can be found by taking a derivative.

C. Reaction Thrusters

The final type of control system is reaction thrusters which can provide a torque and force onto the spacecraft when fired. But, the force due to the thrusters is ignored for simplicity. The torque is dependent on the location and direction of the thruster relative to the center of mass of the spacecraft. Thrusters are common on a lot of spacecraft and are also used in conjunction with reaction wheels as well as CMGs for dumping excess momentum in the wheels. The basic equation of the Torque generated by a thruster is given by[3]

$$\vec{T}_{Thrust} = \vec{r} \times \vec{F}_{Thrust} \tag{69}$$

where $\vec{T}_{Thrust}$ is the torque generated by the thruster, $\vec{r}$ is the position vector of the thruster relative to the center of mass and $\vec{F}_{Thrust}$ is the force generated by the thruster. The force of the thruster and the position vector are assumed to be fixed quantities. The simulation allows for up to 12 different thrusters all with different locations and thrust vectors. An algorithm was built to be able to find the torque generated by firing all combinations of thrusters up to a total of six at a time. Which combination of thrusters that are fired is once again determined by the quaternion control law seen in Eq. (61). The thruster combination that is closest to the command torque is chosen and fired for the duration of the pulse time (which can be set by the user) before a new combination is chosen to fire. The spacecraft acceleration equation for the spacecraft with thrusters is given by[3]

(70)

where is the angular acceleration of the spacecraft and are the external torques (this needs to be in body frame). From the acceleration the body rates can be found by taking a derivative. To allow for combinations control with thrusters and another type of control system (for momentum dumping, as an example), simply add the torque generated by thrusters to Eq. (62) or (68).

V. GUI Instructions

The first thing that is required is to have the GUI be in the current directory in MATLAB before running, as seen in Fig. 3. To start the program run 'SatelliteGUI.m' by typing SatelliteGUI in the command line or by opening the file and running the script.

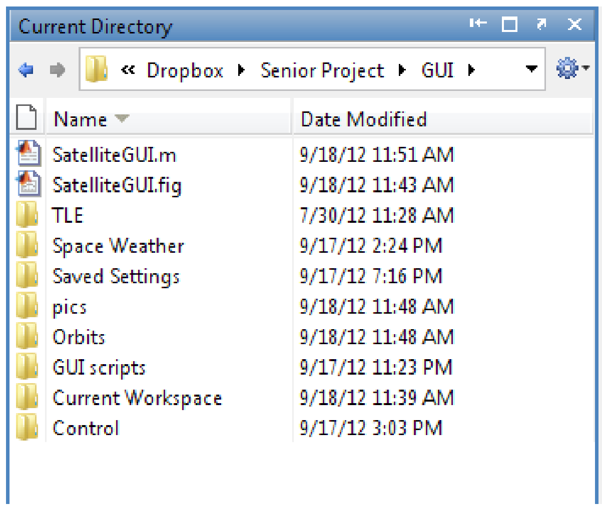

Figure 3. The GUI's main directory in MATLAB.

The main page of the GUI is the most important page as it is where simulations are chosen and ran. "Events" are a core part of the program. They allow the user to separate the spacecraft simulation into an infinite amount of parts. For example, one event can be reaction wheel control for a few weeks while the next event can be a thruster controlled despin maneuver. To create an event one has to simply hit the "Add Event" button on the main page. Each event allows you to choose what kind of event it is (Orbits, thruster control, CMG control and etc.) and how long to run the event for. Events use the parameters of the previous event(s) to continue the spacecraft simulation (except for orbit only events) including date, wheel speed, spacecraft mass, attitude, altitude and etc. Events must be run in order for accurate numbers. The main page also has coordinate frames for reference to the user as well as the final parameters of the last event that ran for quick reference. Finally, the save and load buttons allows the user to save the current simulations and load other simulations. The data for events that have been run are also saved so there is no need to rerun events after they are loaded. The page can be seen in Fig. 4.

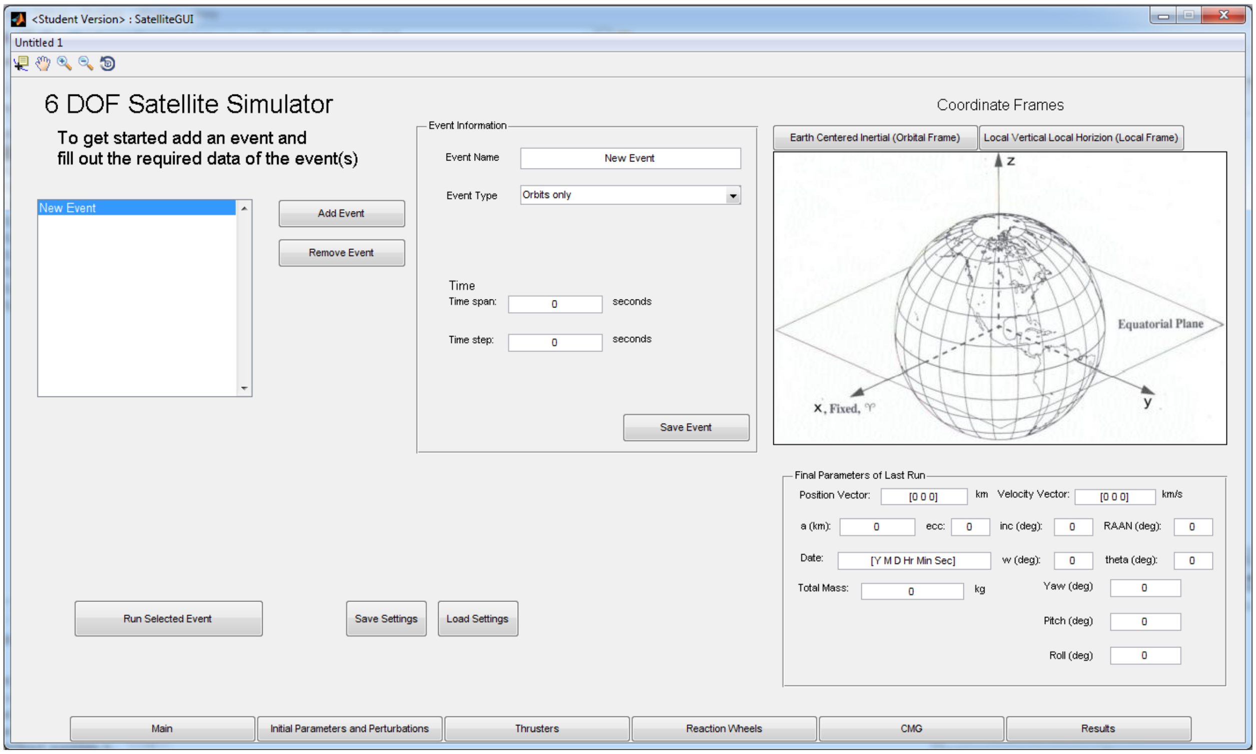

Figure 4. Main GUI page.

The second GUI page is the initial parameters and perturbations page. This page holds all general initial parameters of the spacecraft as well as all the information required for perturbation modeling. The program allows for three different methods of determining initial orbital parameters. The first is by putting a .txt file of a TLE in the TLE directory and loading it to the program, the second method is to use position and velocity vectors and the final method is to use classical orbital elements.

For perturbation modeling the GUI allows for the user to choose which ones are to be modeled during the simulation through checkboxes. Of course, there is information that needs to be inputted for every perturbation that is turned on. For SRP the most recent version of the space weather .txt file should be placed in the space weather directory and loaded into the program (the most recent version can be found at http://www.celestrak.com/). All the boxes that are for attitude simulation are not required for an orbits only event (such as initial attitude angles). This page can be seen in Fig. 5.

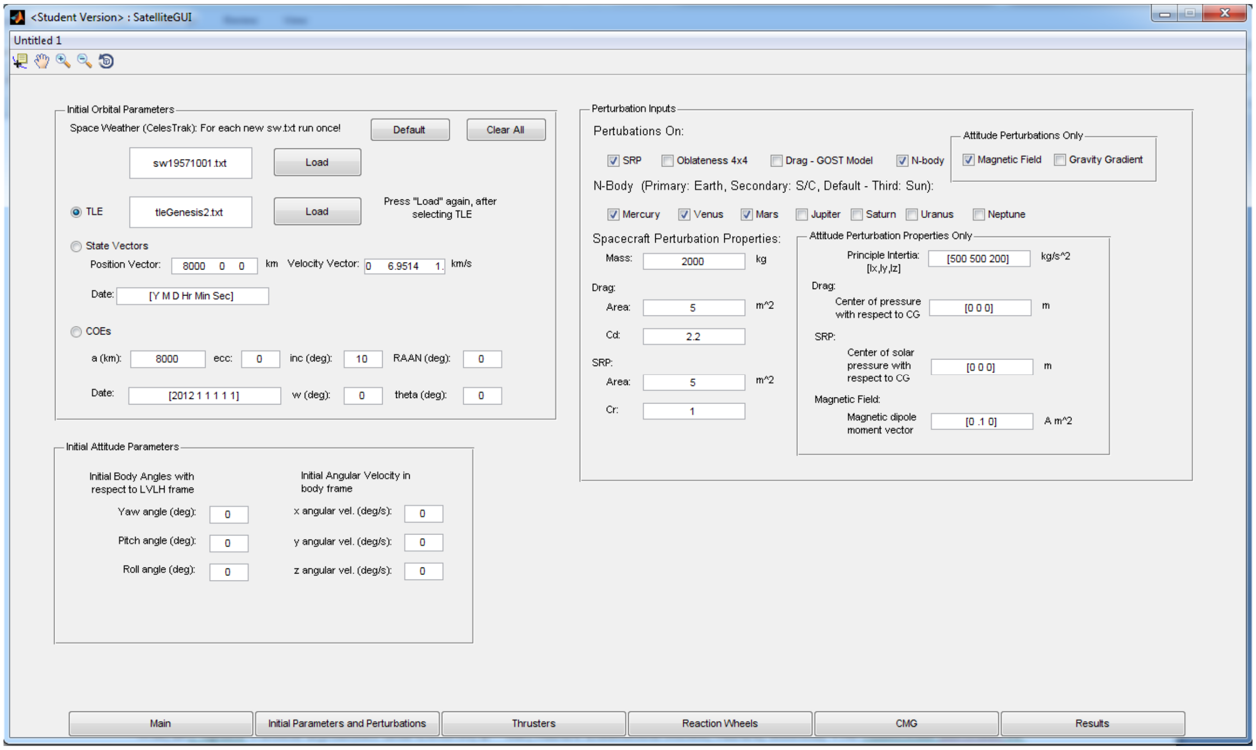

Figure 5. Initial parameters and perturbations GUI page.

The thrusters page holds all the information about reaction thrusters on the spacecraft, as can be seen in Fig. 6. This page only needs to be filled out if thrusters are used on the spacecraft. The inputs include the force vector, position vector and mass flow rate of each thruster up to a total of 12 as well as the standard six control gains.

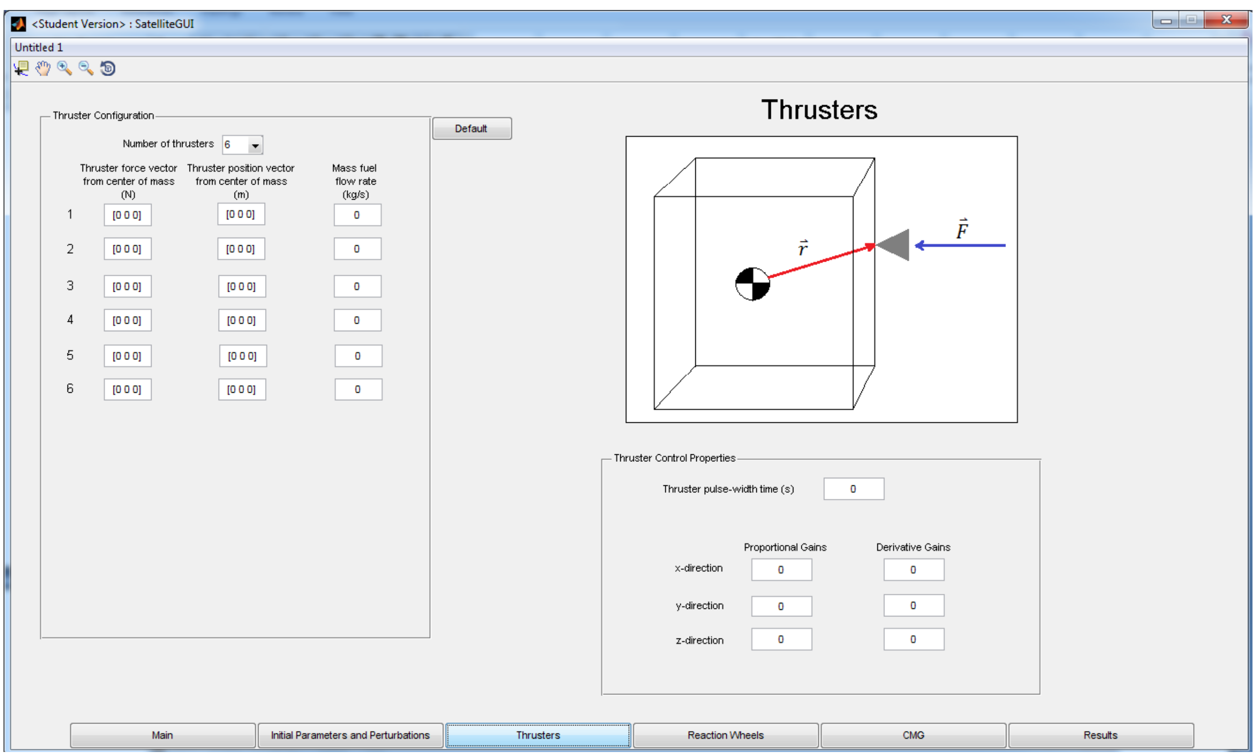

Figure 6. Thrusters GUI page.

Figure 7 is the reaction wheel page that holds all of the information about reaction wheels inside the spacecraft. This page only needs to be filled out if reaction wheels are used on the spacecraft. The inputs include the inertia of the four wheels (assumed to be the same for all four) the beta angle and initial conditions of the reaction wheels as well as the standard six control gains.

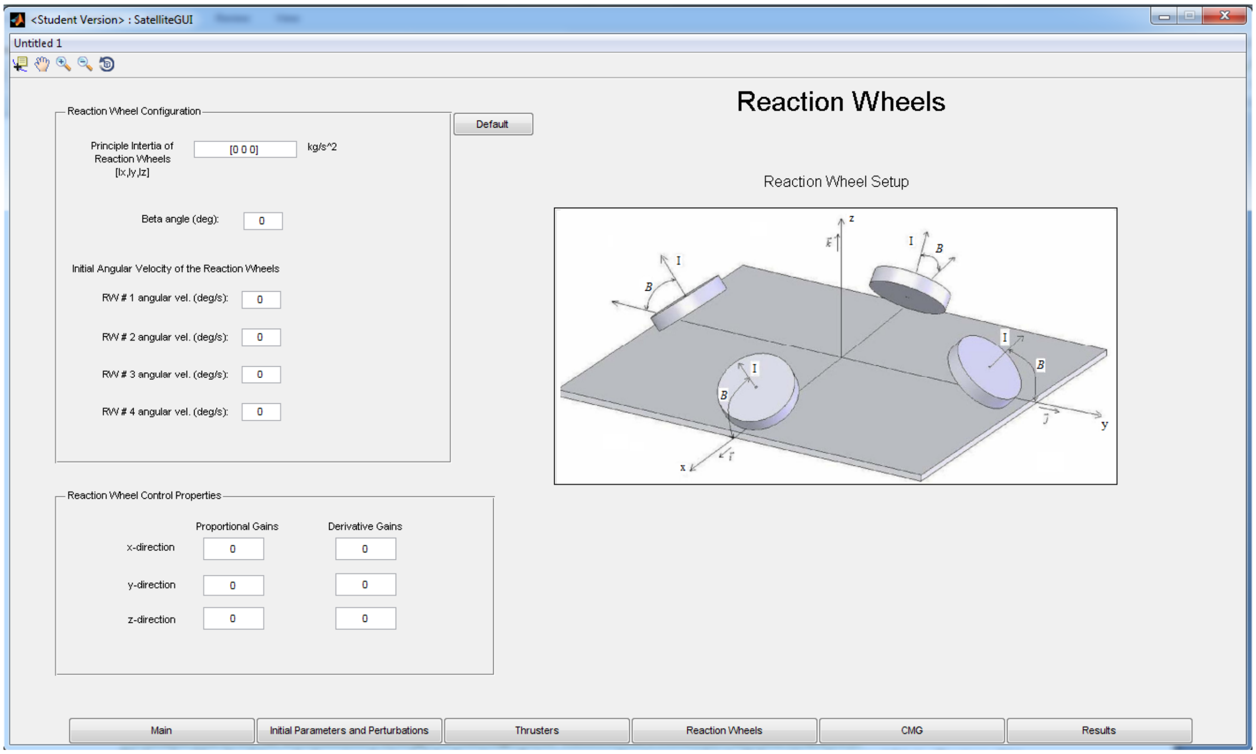

Figure 7. Reaction wheels GUI page.

The CMG page holds all the information about the control moment gyro inside the spacecraft, as can be seen in Fig. 8. This page only needs to be filled out if a CMG is used on the spacecraft. The inputs include the inertia of the wheel, max speed of the gimbal angles and initial conditions of the CMG as well as the standard six control gains.

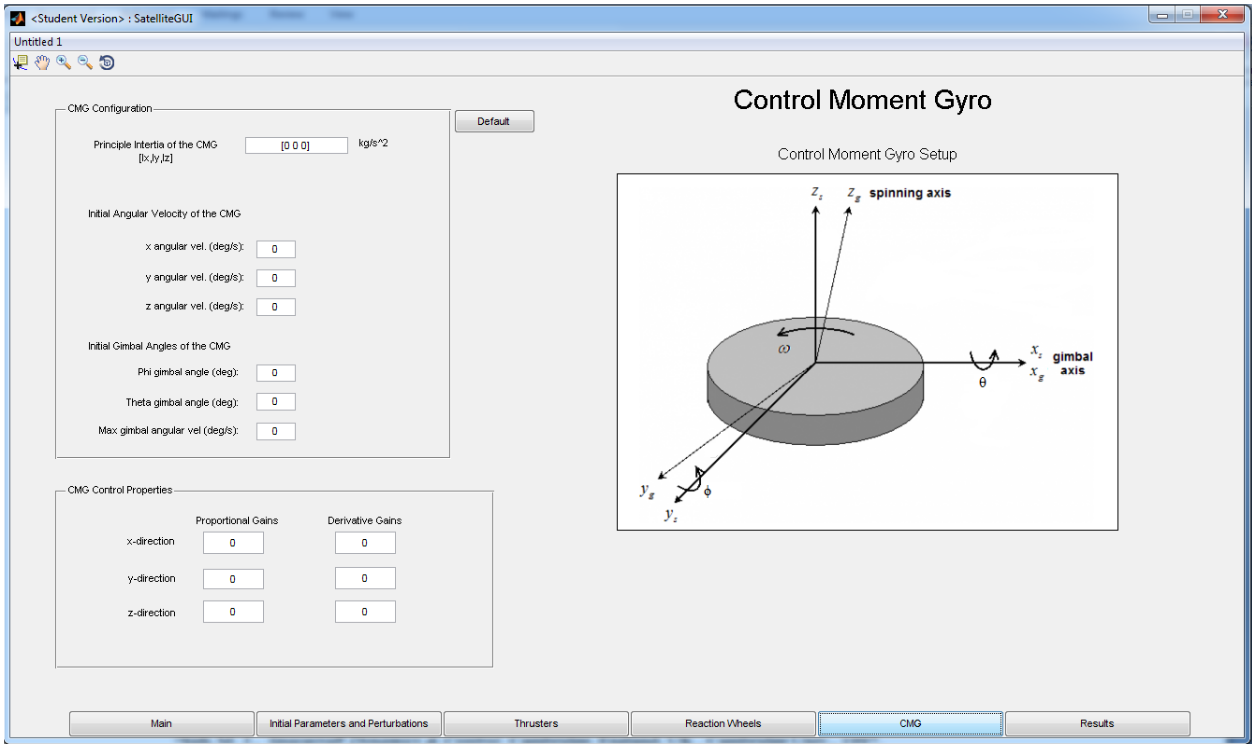

Figure 8. Control moment gyro GUI page.

The final page is the results page, seen in Fig. 9. This page has two graphs with dropdown menus that allow the user to change what data to plot. The top dropdown menu changes what event data to look at. The results include position, velocity, attitude, wheel speeds, mass, disturbances, and much more.

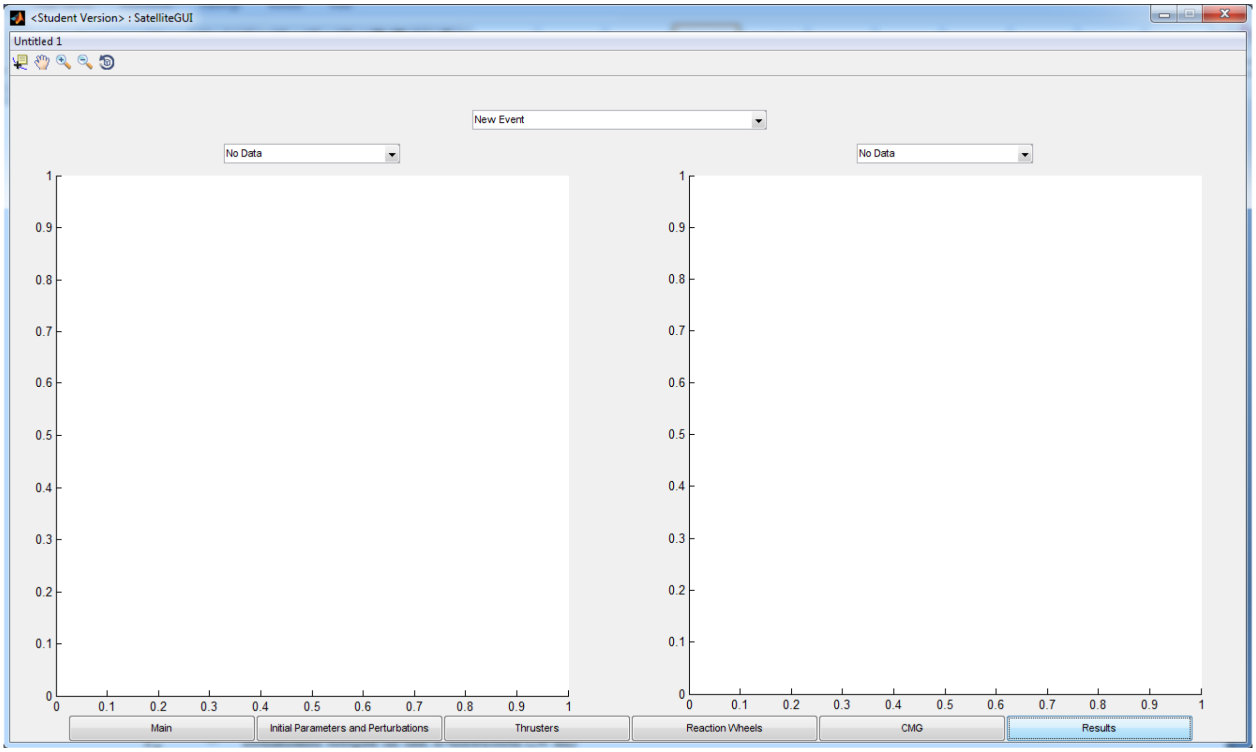

Figure 9. Results GUI page.

VI. Results and Discussion

Most of the individual MATLAB scripts that are combined to create the program were individually tested against examples from the references to prove their validity. This test is to examine the program as a whole. A test case of a spacecraft with reaction wheel control and thruster despin will be used to discuss the results of the program. The events can be seen in Fig. 10. The first event is an empty event (to test if the system can handle such situations); the first real event is a propagation of an orbit with reaction wheel attitude control and a change in attitude. The second event is a despin maneuver with thrusters to dump the momentum from the wheels and the last event is a continuation of reaction wheel attitude control.

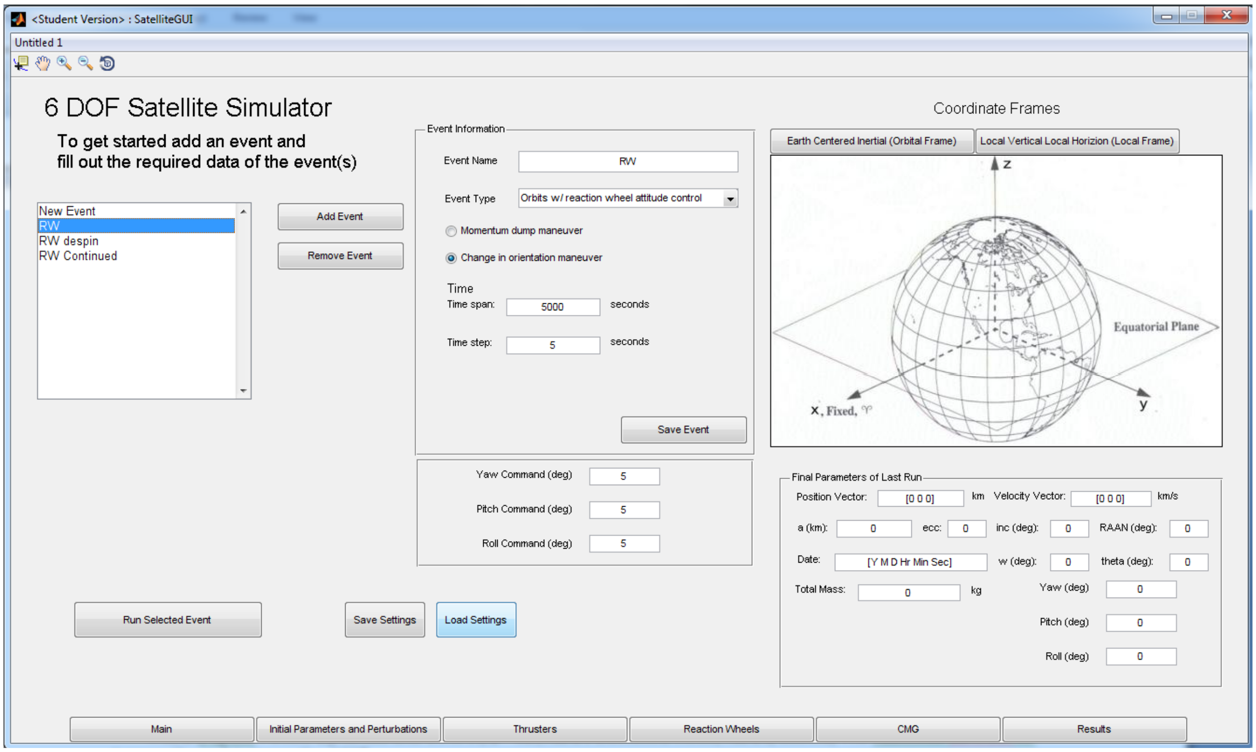

Figure 10. Main GUI page for reaction wheel test case.

The next three Figs., 11, 12 and 13, show the initial conditions, perturbations active as wheel as all the information required for thruster and reaction wheel control. All perturbations are active for testing. This test case has eight reaction thrusters for thruster control. The initial orbit for the test has a semi-major axis of 8000 km, inclination of 10 degrees, and, with an eccentricity, RANN, ω, and θ of zero degrees. The initial date is January 1, 2012 at 1:01:01 (ZULU), with a spacecraft total mass of 2000 kg, cross-sectional area of 5 m^2, coefficient of drag of 2.2, and coefficient of reflectivity of 1 (black body). The initial attitude for the test is set at zero degrees for yaw pitch and roll. The commanded attitude is set at five degrees for all three. The reaction wheels set to have an initial spin of 200 $deg \cdot s^{-1}$ in order to see the effects of the despin maneuver.

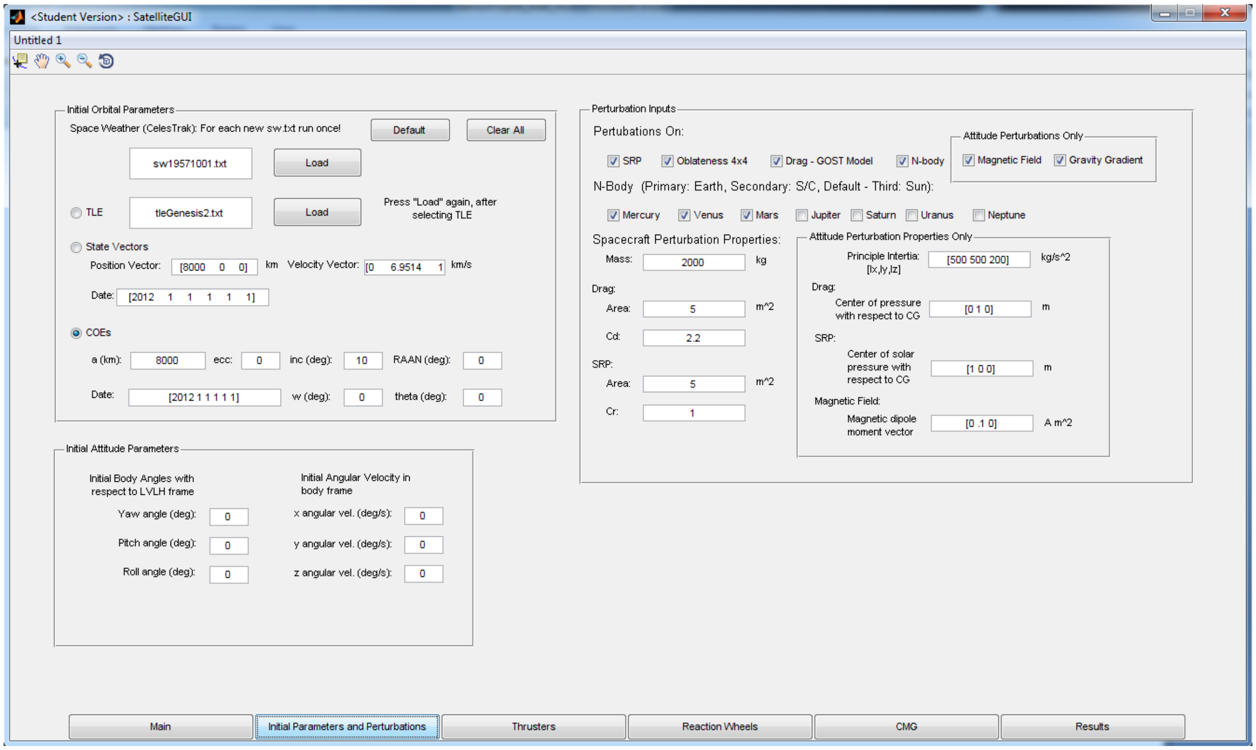

Figure 11. Initial parameters and perturbations GUI page for reaction wheel test case.

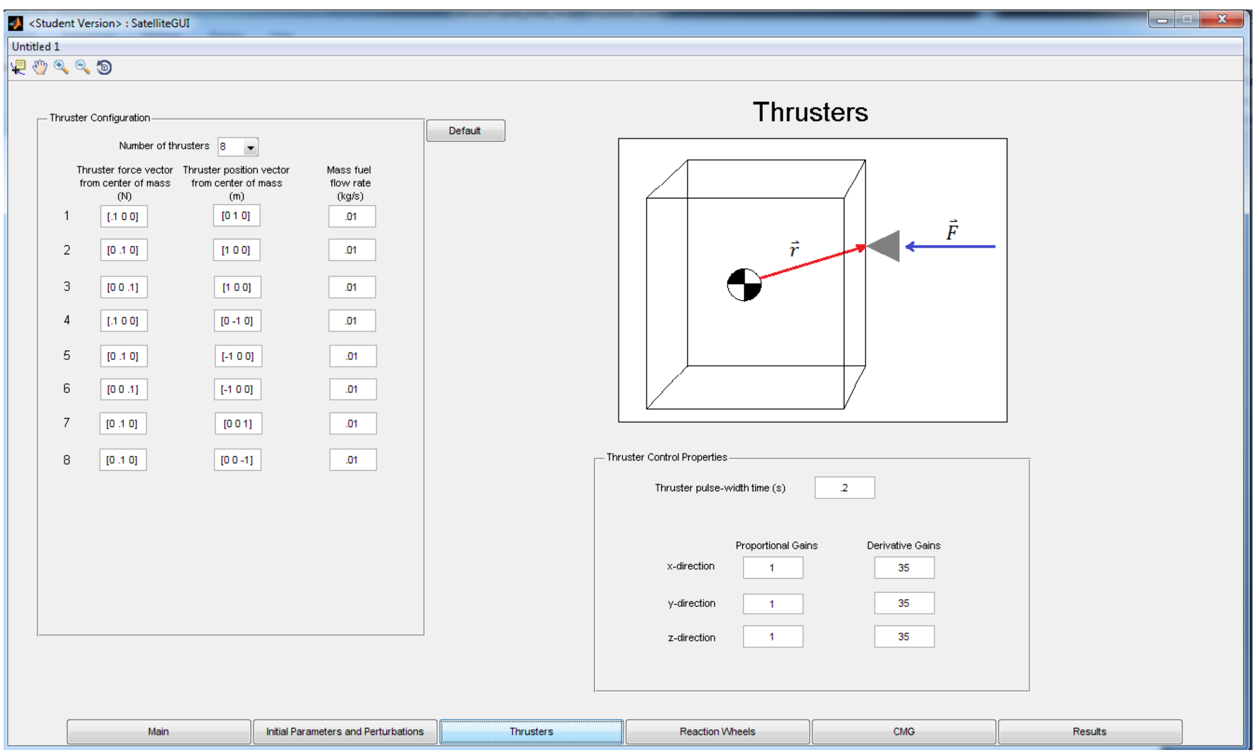

Figure 12. Thrusters GUI page for reaction wheel test case.

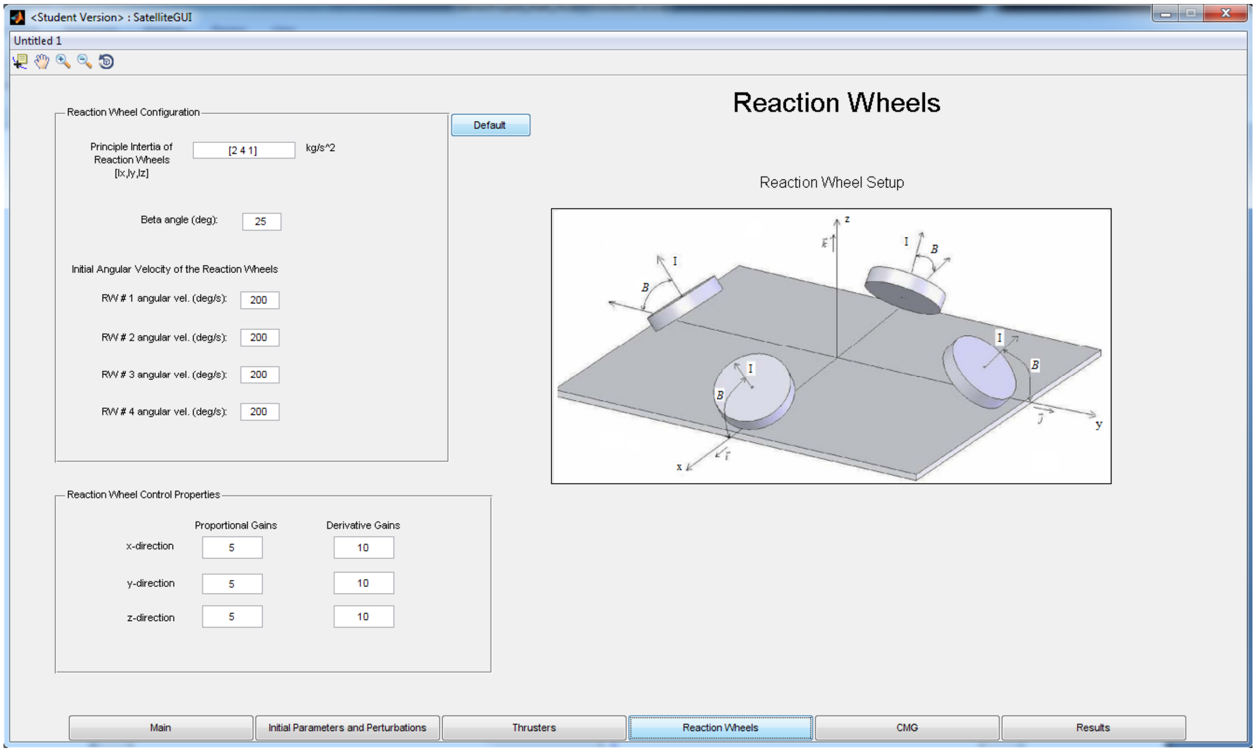

Figure 13. Reaction wheel GUI page for reaction wheel test case.

Figure 14 shows an example set of results of the first event, which again is a change in attitude maneuver with reaction wheel control. The left side shows the orbit around Earth while the right side shows the speed of reaction wheel 1 as it goes through a change in attitude maneuver (the control gains were not optimized for this test). The change in reaction wheel speed is plausible since the control gains were never optimized; the wheel speeds will oscillate due to overshoots as it tries to reach the commanded maneuver.

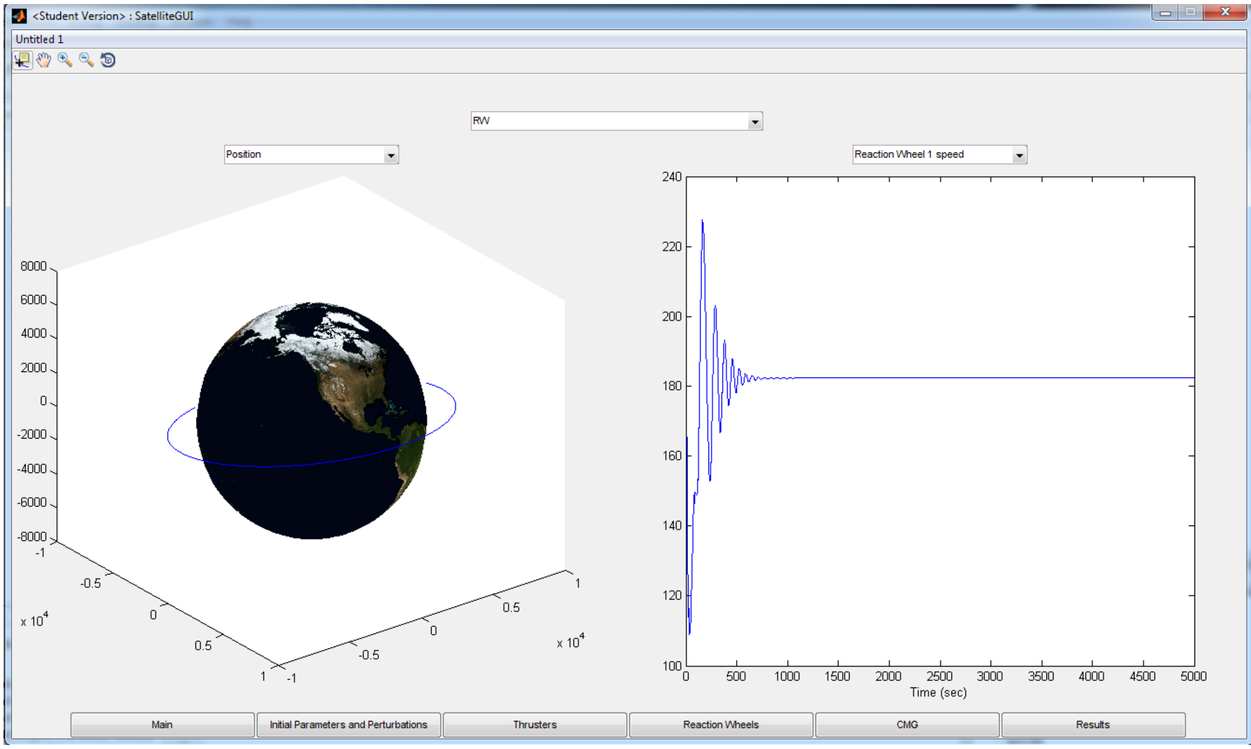

Figure 14. Example results GUI page for first event.

Figure 15 shows an example set of results of the second event, which is the despin maneuver. The left side shows the shows the speed of reaction wheel 1 as the momentum is dumped out of it, and the right side shows the change in attitude of the spacecraft during the dumping.

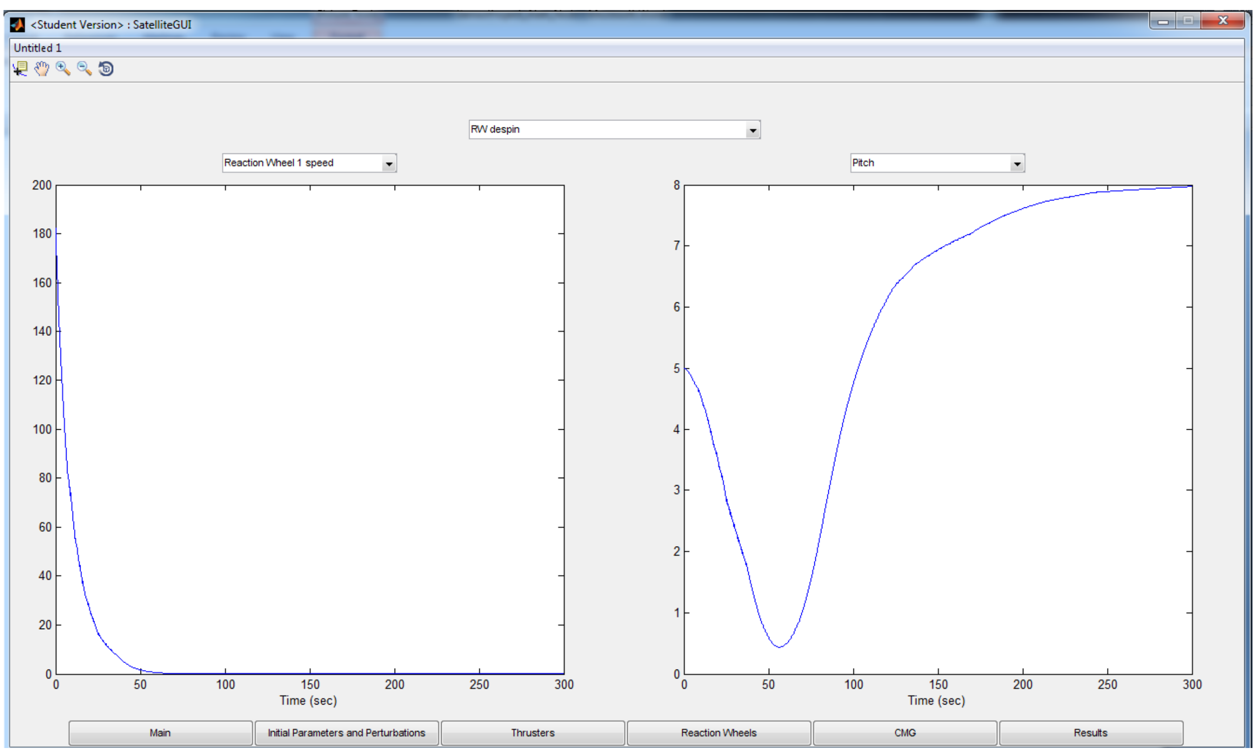

Figure 15. Example results GUI page for second event.

Figure 16 shows an example set of results of the third event, which is again, reaction wheel control. The left side shows the shows the total acceleration that is felt by the spacecraft due to perturbations, and the right side shows the individual torques produced due to perturbations as well. The perturbations felt by the spacecraft are reasonable since with an orbit radius of 8000 km, drag is nearly negligible and about on the same order as SRP and magnetic torques.

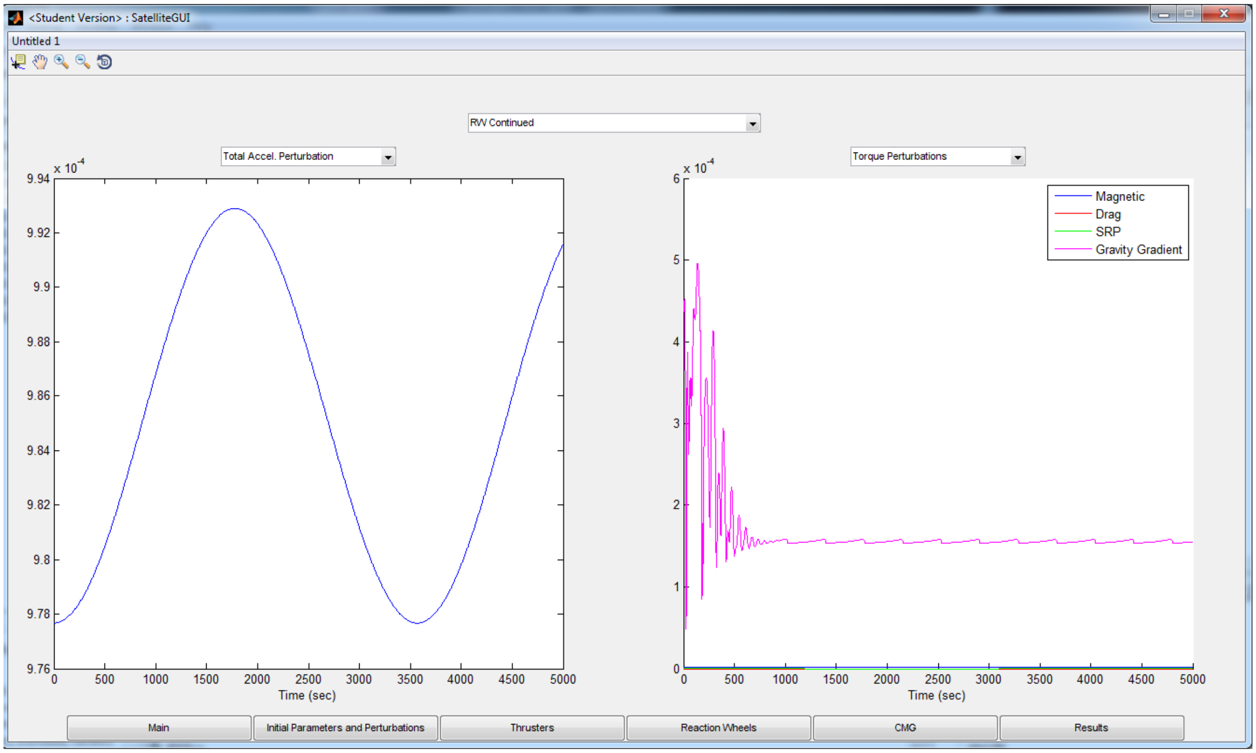

Figure 16. Example results GUI page for third event.

The GUI has been shown to work with reaction wheels and thruster, but other test cases have shown that CMGs seem to be modeled just as well as the other two. This program is quite flexible and is shown to be able to give the user a wide array of different performance and disturbance information.

VII. Conclusion

This program was created to make a user friendly simulation of an Earth orbiting spacecraft. The program is capable of propagating an orbit with perturbations and performing attitude maneuvers for three-axis stabilized spacecraft. While accounting for all these types of perturbations and control actuators, it should be noted that certain elements were ignored, such as the fact that reaction thrusters can change the position of spacecraft (not just rotationally) and the moon has an effect on the spacecraft. With the validation of individual parts of the program, due to its complexity and variability, the results of the simulation test prove to be reasonable. Further improvements can be made by reducing some of the assumptions made, as well as adding in station-keeping maneuvers. To see the MATLAB code for this program, please contact the Cal Poly Aerospace Engineering Department.

References

[1]Wertz, J. R., Everett, D. F., et al., *Space Mission Engineering: The New SMAD*. Microcosm Press, Hawthorne, CA, 2011.

[2]Wertz, J. R., *Orbit & Constellation Design & Management*. Microcosm Press, Hawthorne, CA, 2001.

[3]Curtis, H. D., *Orbital Mechanics for Engineering Students*, Elsevier Ltd, Mass., 2005.

[4]Vallado, A., D., *Fundamentals of Astrodynamics and Applications*, 3rd ed., Microcosm Press, Hawthorne, CA, 2007.

[5]"Astrodynamic Coordinates," *Orbital Mechanics with Numerit* [online pdf], URL: http://www.cdeagle.com/omnum/pdf/csystems.pdf [cited 13 September 2012].

[6]Schaub, H., Junkins, J. L., Analytical Mechanics of Aerospace Systems, AIAA Education Series, AIAA, Reston, VA. American Institude of Aeronautics and Astronautics, 2002.

[7]Downs, C., M., *Adaptive Control Applied to the Cal Poly Spacecraft Attitude Dynamics Simulator*, California Polytechnic State University, San Luis Obispo, 2009.

[8]Sidi, M. J., *Spacecraft Dynamics & Control*. Cambridge, England, UK , Cambridge Univ., 1997.