SPASIM: A SPACECRAFT SIMULATOR

Carlos A. Liceaga
NASA Langley Research Center
8 Langley Blvd., M/S 328
Hampton, VA 23681-0001
c.a.liceaga@larc.nasa.gov

ABSTRACT

The SPAcecraft SIMulator (SPASIM) simulates the functions and resources of a spacecraft to quickly perform conceptual design (Phase A) trade-off and sensitivity analyses and uncover any operational bottlenecks during any part of the mission. Failure modes and operational contingencies can be evaluated allowing operational planning (what-if scenarios) and optimization for a range of mission scenarios. The payloads and subsystems are simulated, using a hierarchy of graphical models, in terms of how their functions affect resources such as propellant, power, and data. Any of the inputs and outputs of the payloads and subsystems can be plotted during the simulation or stored in a file so they can be used by other programs. Most trade-off analyses, including those that compare current versus advanced technology, can be performed by changing values in the parameter menus. However, when a component is replaced by one with a different functional architecture, its graphical model can also be modified or replaced by drawing from a component library. SPASIM has been validated using several spacecraft designs that were at least at the Critical Design Review level. The user and programmer guide, including figures, is available on line as a hypertext document. This is an easy-to-use and expandable tool which is based on MATLAB® and SIMULINK®. It runs on Silicon Graphics Inc. workstations and personal computers with Windows 95™ or NT™.

INTRODUCTION

The primary function of the SPAcecraft SIMulation (SPASIM) software is to create a virtual environment to simulate a spacecraft. The simulation includes the spacecraft's operation and the interaction of multiple subsystems as a function of time and resources. SPASIM presents this virtual environment to the user in a graphical/object-oriented interface to enhance usability and integration.

SPASIM defines a hierarchy of block diagrams wired together along with parameters that describe operational and performance characteristics that yields a well documented functional spacecraft model. The top-level block diagram is shown in Figure 1. Each block within a graphical user interface (GUI) window defines a function or a hierarchy of lower level blocks. Blocks at the lowest level invoke MATLAB® or SIMULINK® code. The GUI presents a dialog box to the user that allows changes to be made to a block's parameters before simulation starts. Lines connecting the blocks transmit values such as those used to represent orbital information and spacecraft resources. Examples of these resources are propellant, power, and data.

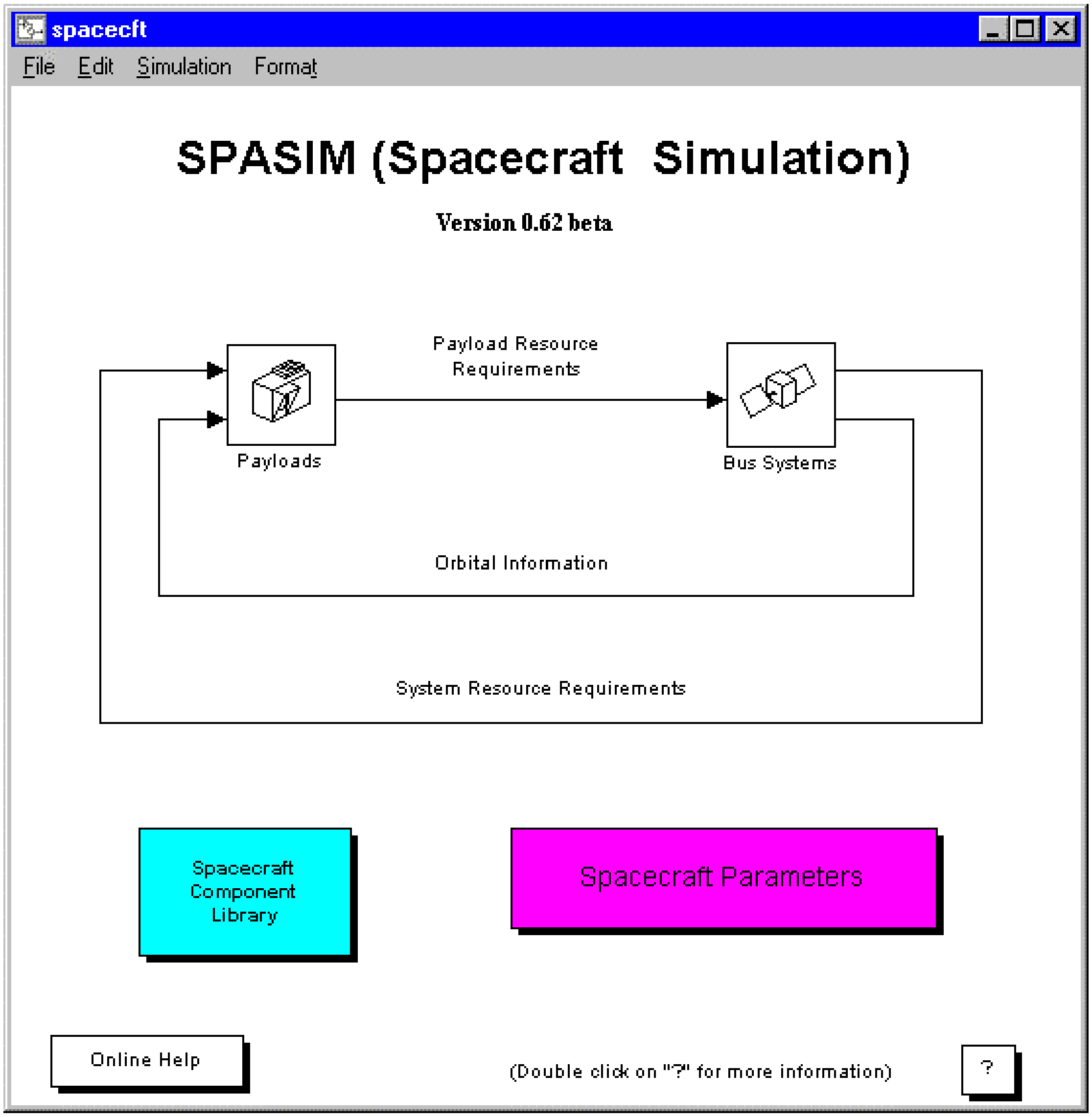


Figure 1: Main Window

The user can analyze subsystem interactions during a simulation by displaying a dynamic plot of any block's input or output values. A large selection of 26 predefined plots may be chosen by clicking on the "Spacecraft Parameters" button and choosing the plot menu from the pop-up window.

SPASIM includes a library of models of spacecraft payloads and subsystem components that represent a range of functionality. The user and programmer guide (Liceaga *et al.* 1997), including figures, may be accessed by clicking on the "Online Help" button. It is also available on the World Wide Web at http://freedom.larc.nasa.gov/projects/spasim/help.html. In addition, each major window has help specific to that window which can be accessed by clicking on the "?" button.

SPASIM is one of the tools in the Satellite System Design and Simulation Environment (Ferebee, Troutman, and Monell 1997). This environment also includes a design and sizing tool and a component database.

SPASIM has been initially implemented for uncrewed, earth-orbiting spacecraft. It can be ported to any platform on which MATLAB® and SIMULINK® can run. It is now running on Silicon Graphics Inc. workstations and personal computers with Windows 95™ or NT™.

SPACECRAFT PAYLOADS

The payloads are the instruments the spacecraft carries to accomplish its mission. By default, there are four payloads, which are shown in Figure 2.

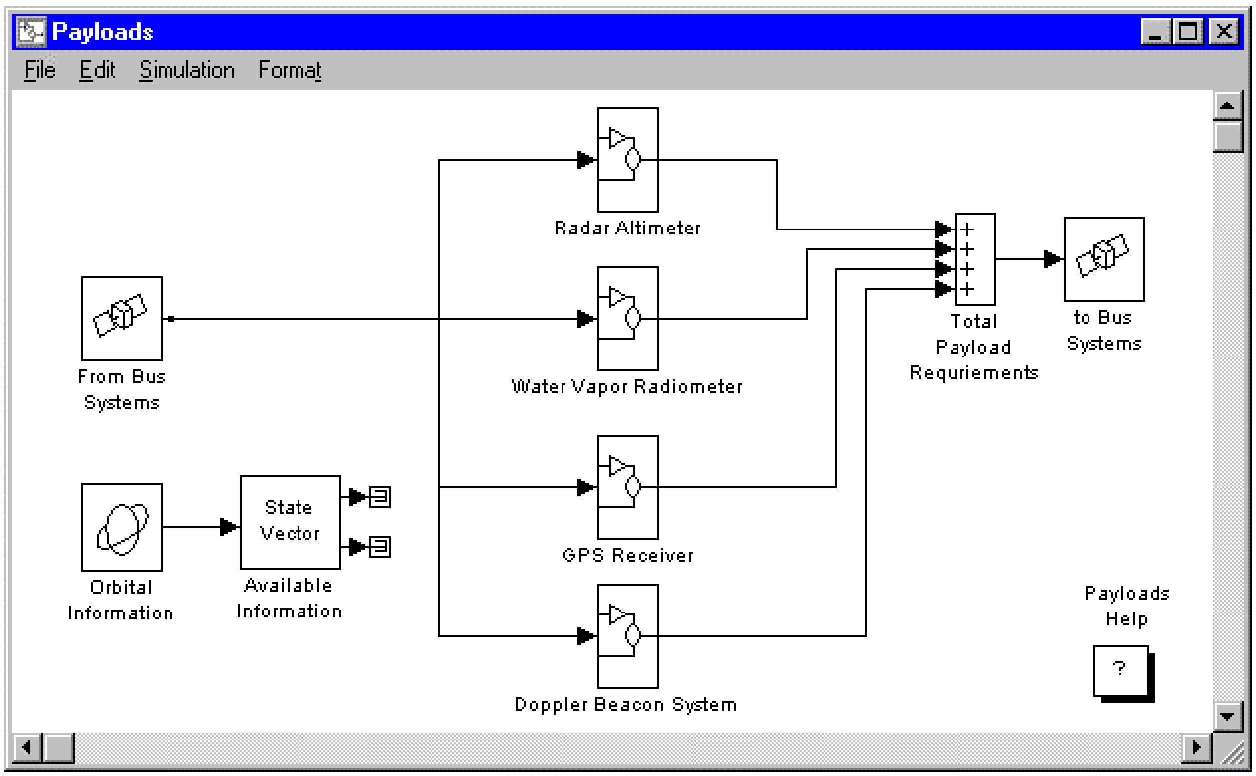


Figure 2: Spacecraft Payloads

Resource requirements and state information from the subsystems flow in from the left. They are distributed to the payloads which output their requirements. Finally, the requirements of each payload are added and feed back to the subsystems as the payload resource requirements.

All payloads share a standard interface. They have access to the same state and orbital information from the subsystems. They can also add to any of the spacecraft resource requirements.

SPACECRAFT SUBSYSTEMS

The spacecraft subsystems provide the resources required for the mission to be accomplished. Together they make up what is commonly called the spacecraft bus. As shown in Figure 3, the spacecraft bus is modeled as being composed of the following subsystems: power; thermal; propulsion; guidance, navigation, and control (GNC); communication and tracking (CT); and command and data handling (CDH).

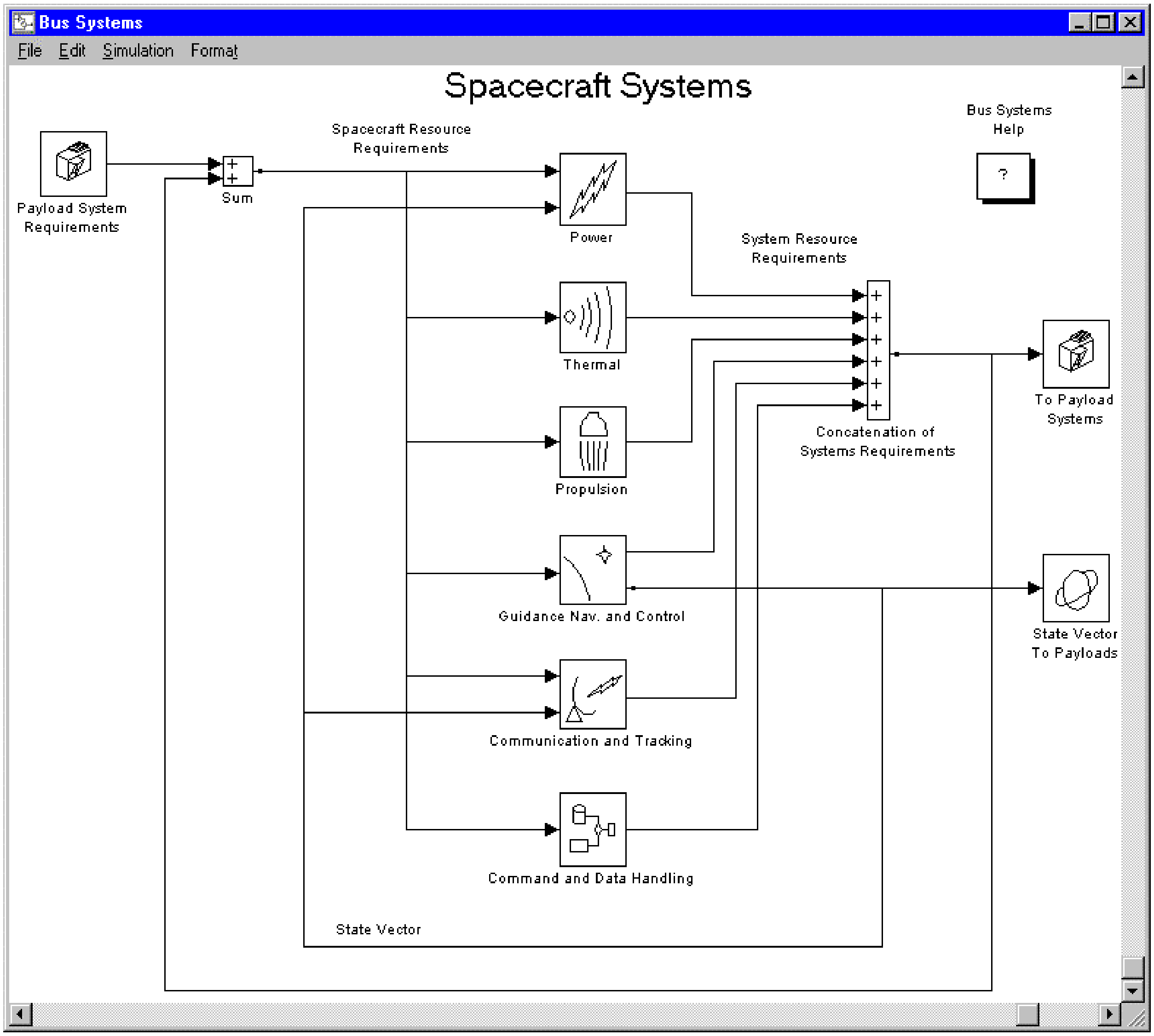


Figure 3: Spacecraft Subsystems

Payload resource requirements flow in from the left. The subsystem requirements are added and feed back to the subsystems as the spacecraft resource requirements.

All subsystems share the same standard interface that the payloads have. They have access to the same spacecraft resource requirements and orbital information. They can also add to any of the spacecraft resource requirements.

This standard interface provides modularity and facilitates independent maintenance of the models by the instrument and subsystem experts. It also facilitates the incorporation of externally developed models.

Each subsystem has a parameter menu. These parameters are used as constants or the initial value of variables.

Power

The purpose of the power subsystem, shown in Figure 4, is to generate, store, and distribute electrical energy. It is implemented through a solar array (produces power), a battery (stores power), and a charge unit (controls power).

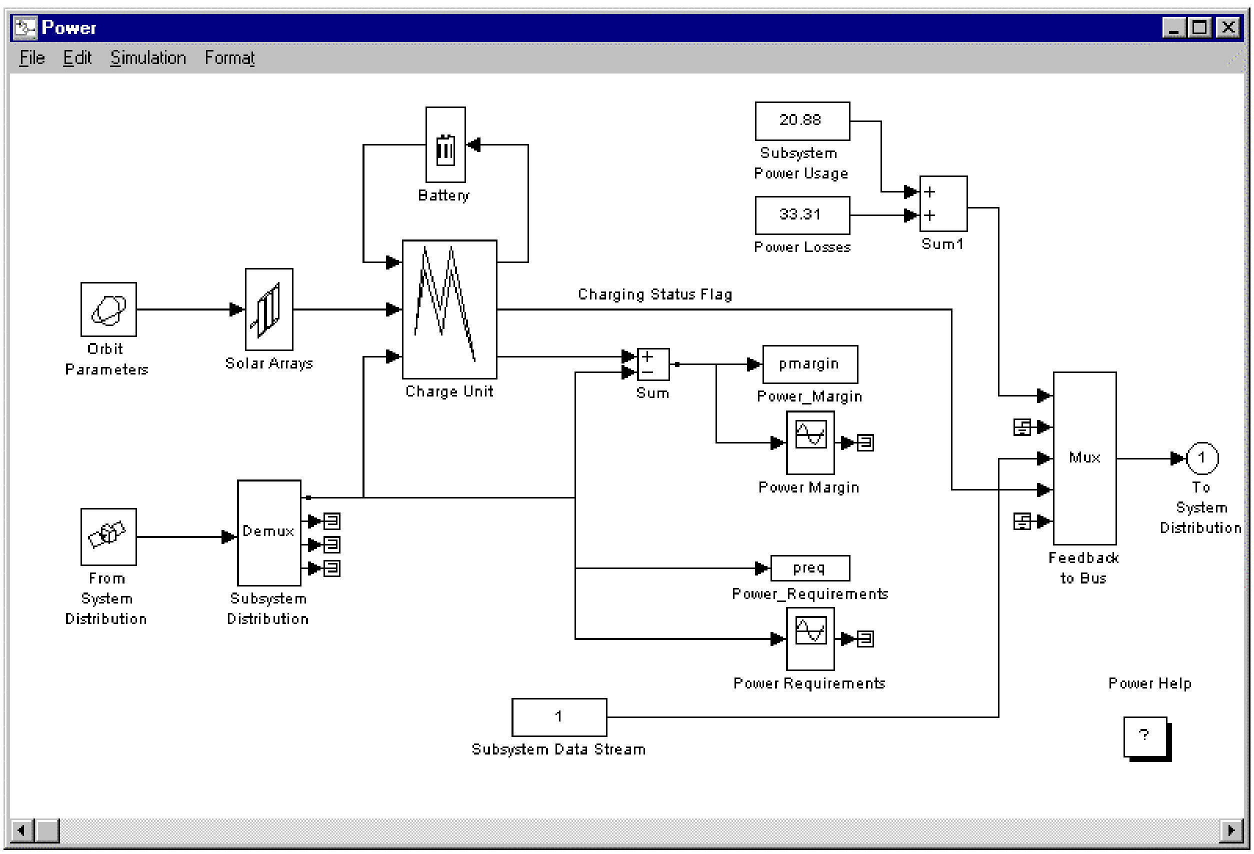


Figure 4: Power Subsystem

The parameters in the main power menu, shown in Figure 5, include the: average minimum solar flux, solar cell type and efficiency, solar array active area, solar cell degradation factor, initial solar array efficiency, solar array shadow file name, solar cell in-service time, power bus efficiency and nominal voltage, and number of degrees of freedom (DOF) of the solar array. The solar array shadow file is an ASCII file, loaded before the start of the simulation, and used by a two-dimensional look-up function in SIMULINK®. This look-up function linearly interpolates the fraction (0.0 to 1.0) of the solar array area available at specific alpha and beta angles for the spacecraft in a nominal local vertical local horizontal (LVLH) flight attitude.

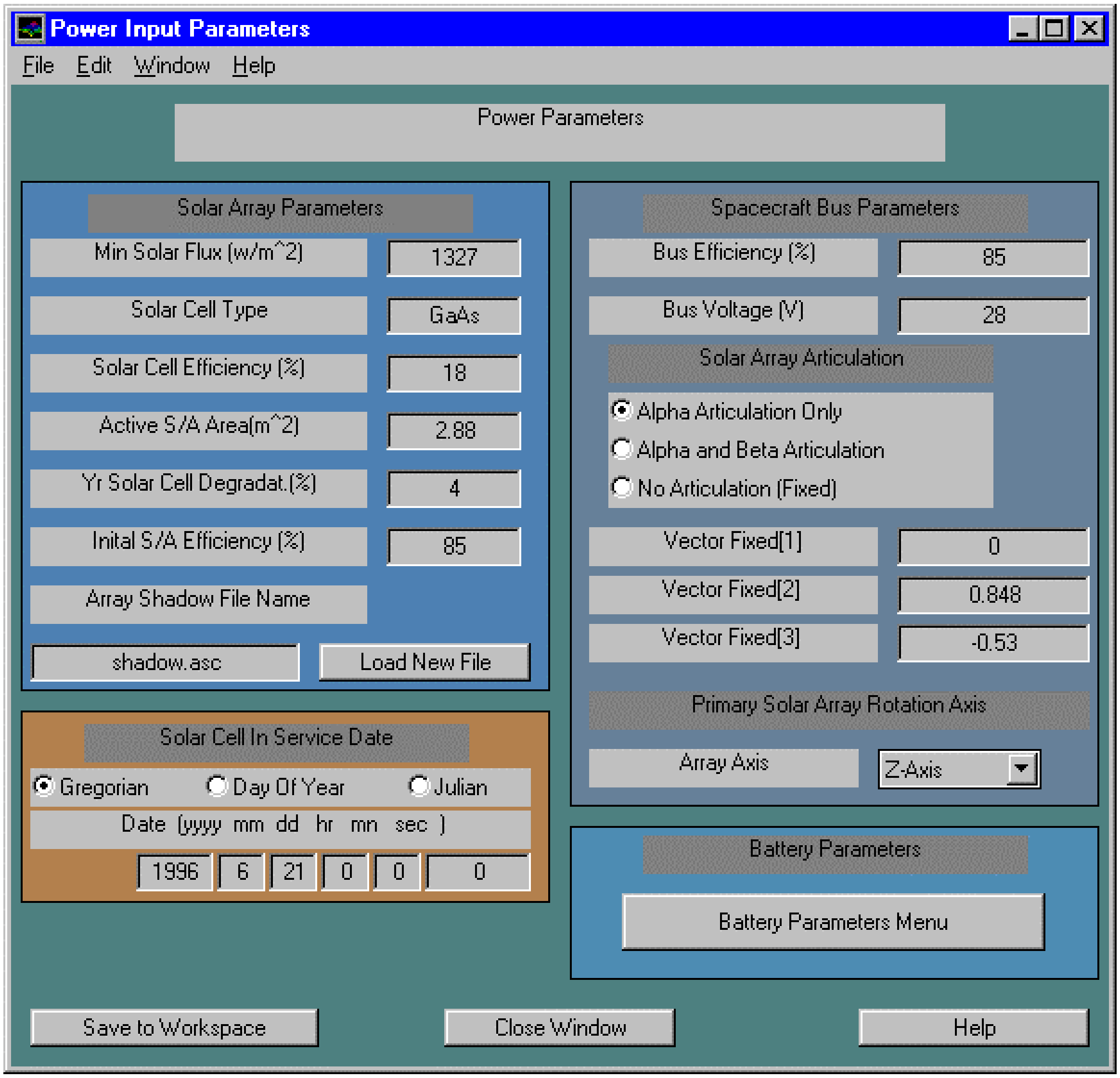


Figure 5: Power Parameters

The parameters in the battery menu, shown in Figure 6, are: maximum end-of-life energy capacity, cell type, charging efficiency, initial charge, minimum depth of discharge (DOD), and maximum charge and discharge currents. The minimum DOD indicates how much the battery needs to be discharged, after obtaining a full charge, before it will be charged again.

The inputs to this model are the: spacecraft power requirement, sun vector, earth vector, and sunlight flag. Its outputs are a battery charging flag, the power it requires, and the rate at which it generates data. This model has predefined plots of time versus: spacecraft power requirements, power margin, depth of discharge, solar alpha, solar beta, shadow table result, initial power factor, and power factor.

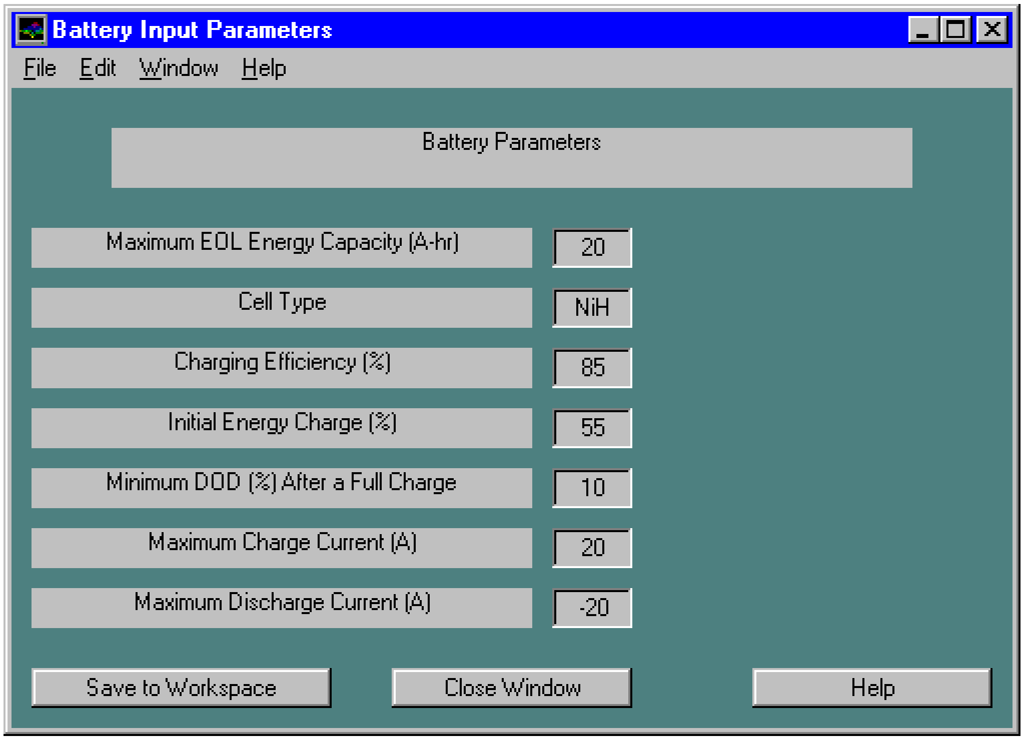


Figure 6: Battery Parameters

<u>Thermal</u>

The purpose of the thermal subsystem, shown in Figure 7, is to maintain spacecraft components within specified temperature limits. The model implemented in SPASIM assumes a cold biased system in which given regions are designed to operate below the upper temperature limits of the components therein. Since this often results in the temperatures, in that region, falling below the lower limits of the components, thermostatically controlled heaters are often employed. Components that can't be effectively cold biased are cooled by passive (stored cryogen and radiative) and/or active (closed cycle and thermoelectric) coolers.

This model also allows the user to simulate custom heater/cooler power profiles. The heaters/coolers are either keyed to a day/night switch or are available on an on-demand basis. Once on, the heaters/coolers will follow a user defined power profile. Note that because of their more restrictive temperature requirements, batteries are assumed to be on a different cold biased loop where heaters are keyed to the charge state of the batteries.

The parameters in the thermal menu, shown in Figure 8, specify the power requirements for: a cryocooler; the battery heater used when neither charging nor discharging; and the payload, propellant, and GNC electronics heaters used during eclipse. The inputs to this model are the battery charging flag from the power subsystem and the sunlight flag from the GNC subsystem. Its outputs are the power it requires and the rate at which it generates data. This model has predefined a plot of time versus thermal power requirement.

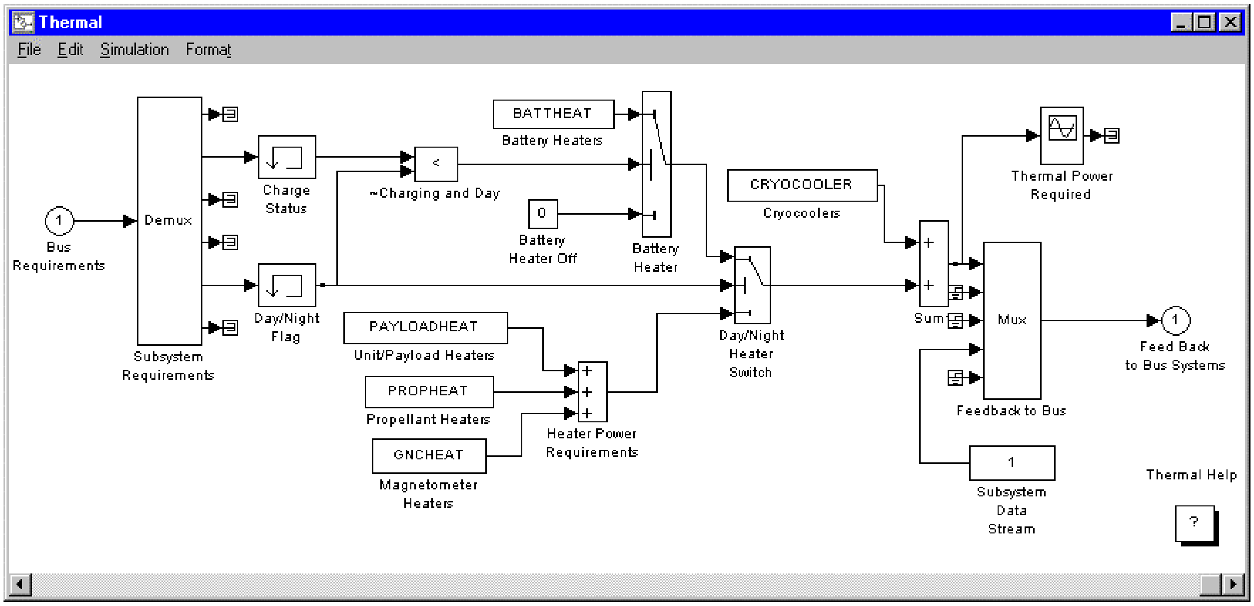


Figure 7: Thermal Subsystem

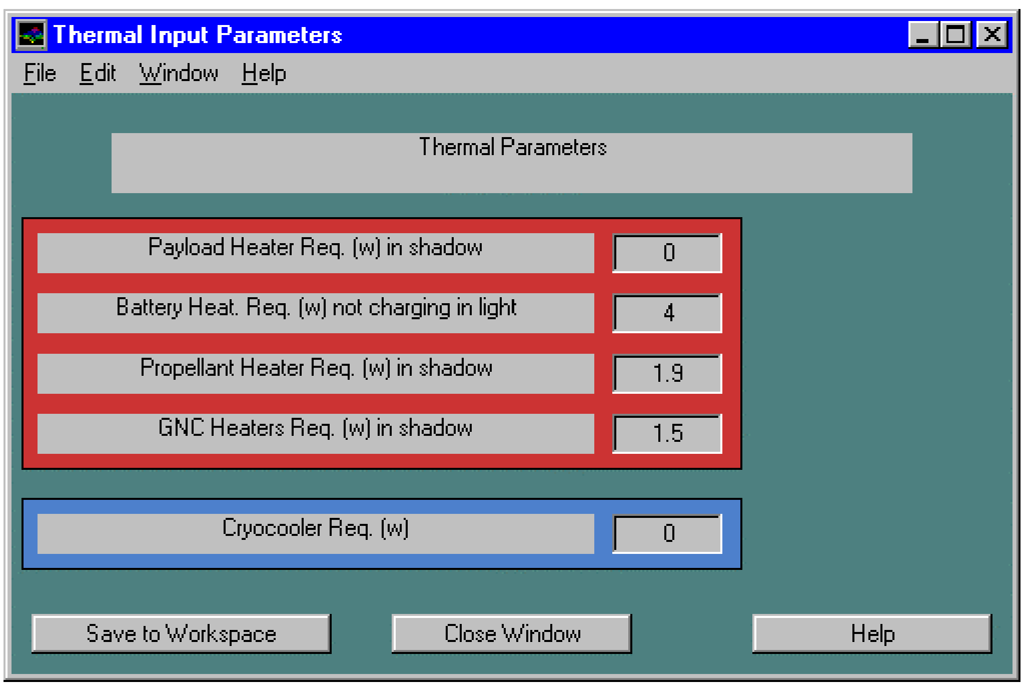


Figure 8: Thermal Parameters

Propulsion

The purpose of the propulsion subsystem, shown in Figure 9, is to provide the thrust required to maintain or change the spacecraft's orbit and attitude. This subsystem uses two resources, propellant and power. The model implemented is strictly an event driven process. Until the GNC subsystem is modeled as a mass-accurate system, this model will stay as a stochastic model. The subsystem is implemented through twelve thrusters. Four thrusters are required to provide the positive and negative torques about each of the three axes.

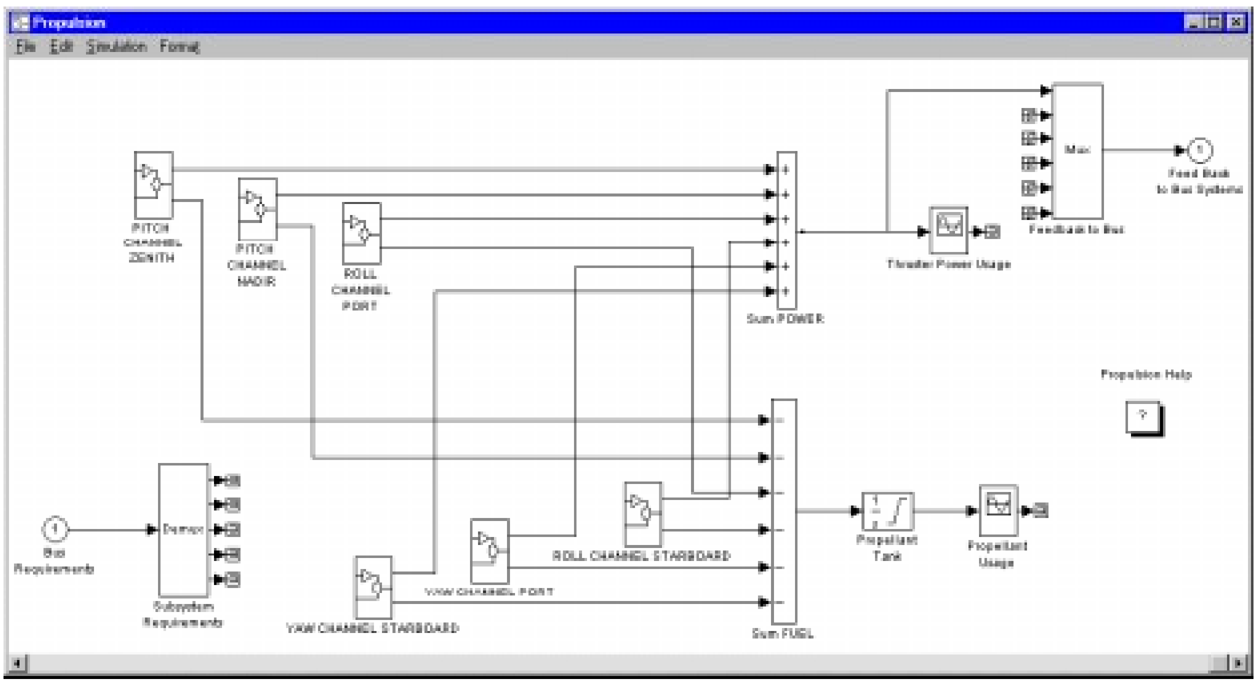


Figure 9: Propulsion Subsystem

The parameters in the propulsion menu, shown in Figure 10, specify the: initial fuel load contained in the tanks for the mission, specific impulse of the propulsion subsystem, minimum time a thruster remains on after an on/off command is issued, power per thruster, amount of fuel burned for each second of engine firing time, number of operational propulsion events to occur in a year, and time the first event will occur from the start of the simulation. The output of this model is the power it requires. This model has predefined plots of time versus thruster power requirement and propellant tank level.

The following three assumptions were used in creating the stochastic resource model for the propulsion subsystem. First, thruster events are modeled as periodic throughout one year. Second, thruster events will have a duration of the thruster's minimum impulse time. Third, a thruster firing sequence will have a duration of two hours.

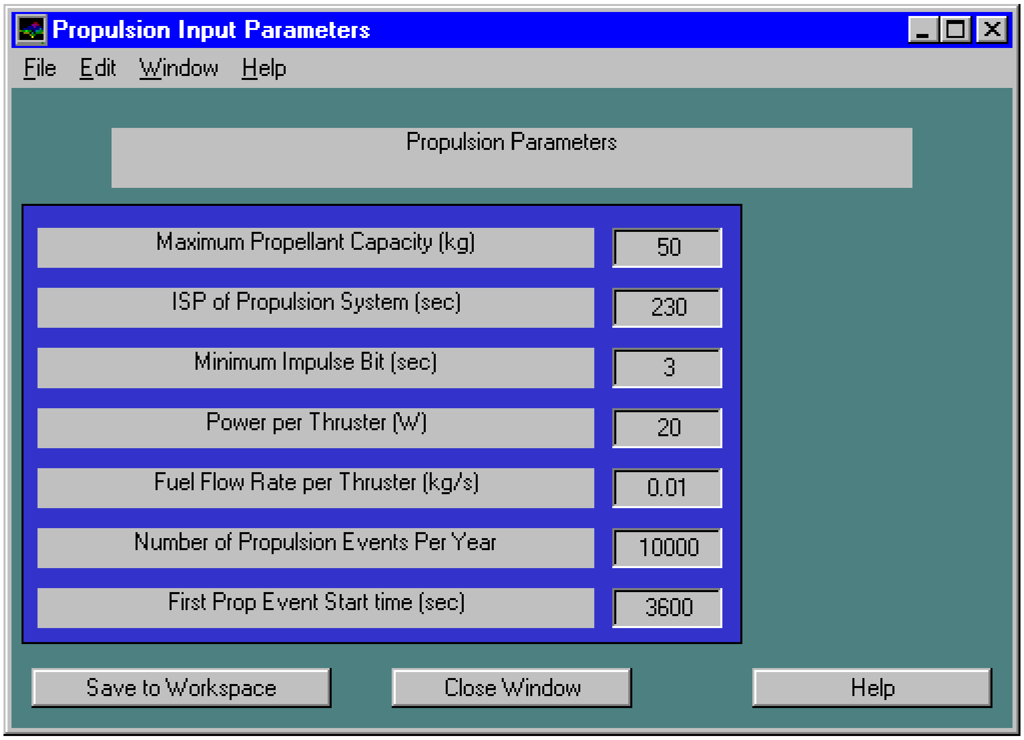


Figure 10: Propulsion Parameters

Guidance, Navigation, and Control

The GNC subsystem is shown in Figure 11. The purpose of a typical GNC subsystem is to provide orbital and attitude determination and control. However, passive or active attitude control isn't simulated. Instead, attitude motion can be prescribed with one of the following four methods: fixed, oscillatory, maneuver, or user prescribed. In fixed, the user specifies an initial attitude and the spacecraft is held fixed at that attitude. In oscillatory, the user specifies amplitudes and frequencies for the three axes. In maneuver, the user specifies up to 10 attitudes and the time in the simulation when they will be reached. In user prescribed, an input file with a proper attitude history is given. This can be the result of an off-line three DOF or a six DOF simulation. It is assumed that the control system can meet the user prescribed attitude profile. There are two attitude modes available to the user. An Earth oriented LVLH mode and an inertial mode. When in inertial mode, the user can specify a spin rate.

The parameters in the main GNC menu, shown in Figure 12, specify the: attitude flight mode, either LVLH or inertial; initial spacecraft attitude; spacecraft spin rate and axis; attitude history type, either fixed, oscillatory, maneuver, or user prescribed; repeat attitude history flag, if maneuver or user prescribed no value matching at beginning or end; oscillatory amplitudes and frequencies for the three axes; and user prescribed attitude history file name. The parameters in the orbit menu, shown in Figure 13, specify the: launch time, simulation start time, spacecraft lifetime, and epoch time. For this epoch time, they also specify the: apogee, perigee, inclination, argument of periapsis, and ascending node.

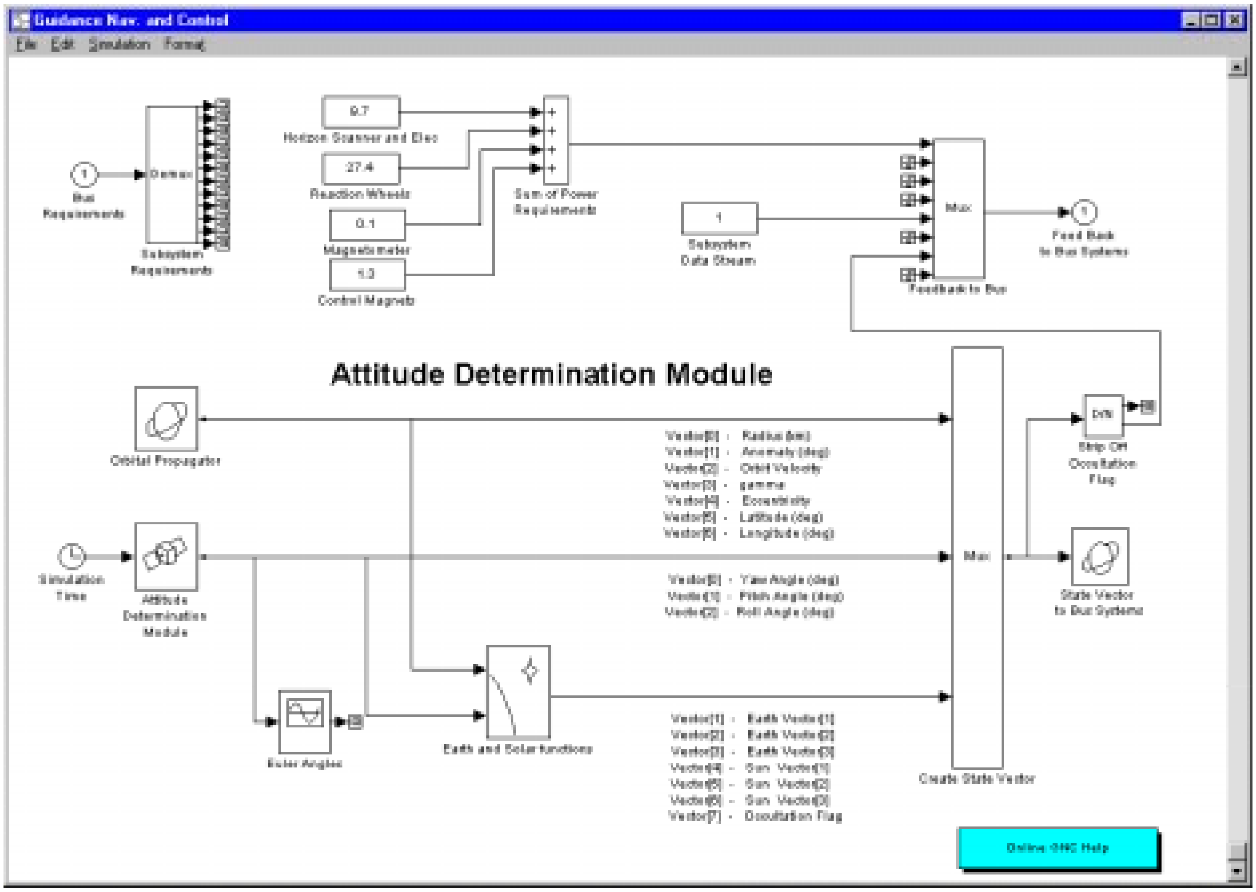


Figure 11: Guidance, Navigation, and Control Subsystem

The orbital information this model outputs is calculated by three submodels: orbital propagator, attitude determinator, and earth and solar pointer. The orbit propagator integrates a simple two body orbit equation using a Runge-Kutta type integrator based on Fehlberg’s seventh-order formula to calculate: radius, true anomaly, velocity, flight path angle, eccentricity, longitude, and latitude. The attitude determinator calculates the spacecraft’s attitude in terms of yaw, pitch, and roll angles. The earth and solar pointer outputs the: sun vector, earth vector, and sunlight flag.

The model also outputs the power it requires and the rate at which it generates data. This model has defined plots of time versus: yaw, pitch, roll, earth vector, sun vector, and sunlight. It has also defined a plot of latitude versus longitude.

Communications and Tracking

The purpose of the CT subsystem, shown in Figure 14, is to communicate with ground stations to download data and upload commands. The carrier-to-noise ratio of both the telemetry downlink and the command uplink is calculated as a figure of merit for the channel carrying capability of the link.

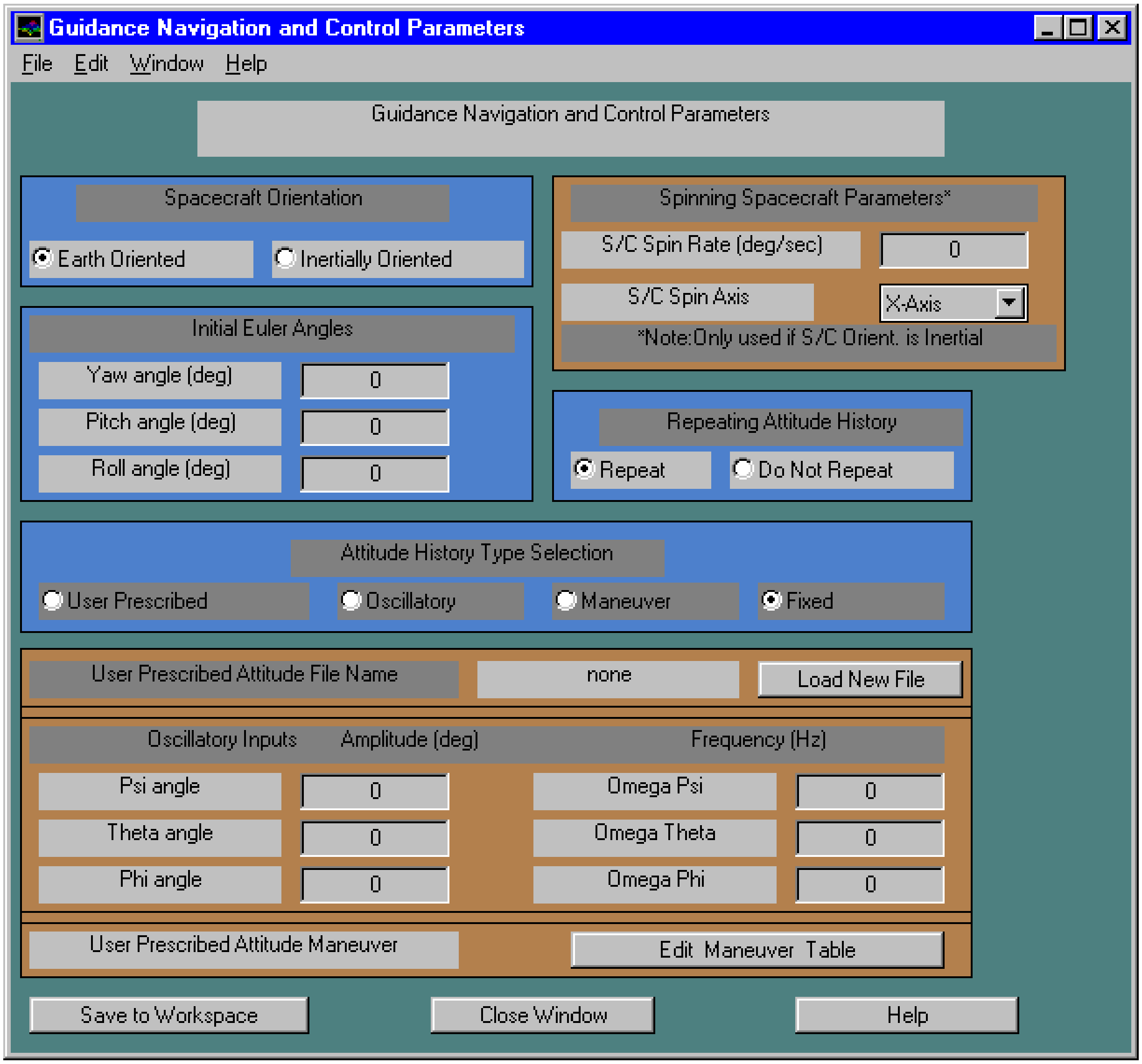


Figure 12: Guidance, Navigation, and Control Parameters

The power flux density (PFD) is calculated at both the ground station site and at the spacecraft during a contact coverage period. This feature helps determine if the link has enough power at the receiver to accept a data transfer link. Because of the change in slant range due to continuously changing position of the spacecraft with respect to the ground station, calculating the PFD gives an indication of the received power range while the ground contact is made. It can help in the design and analysis process for either a ground station or a spacecraft communications subsystem.

The radio frequency (RF) link margin between the space and ground communication segments is calculated to give an indication of RF performance levels available to maintain adequate communication. Based upon requirements such as bit-error-rate, the link margin gives the theoretical potential of the link to perform to certain specifications. If the calculated performance exceeds requirements, savings can be realized by relaxing the ground or spacecraft communications subsystem design.

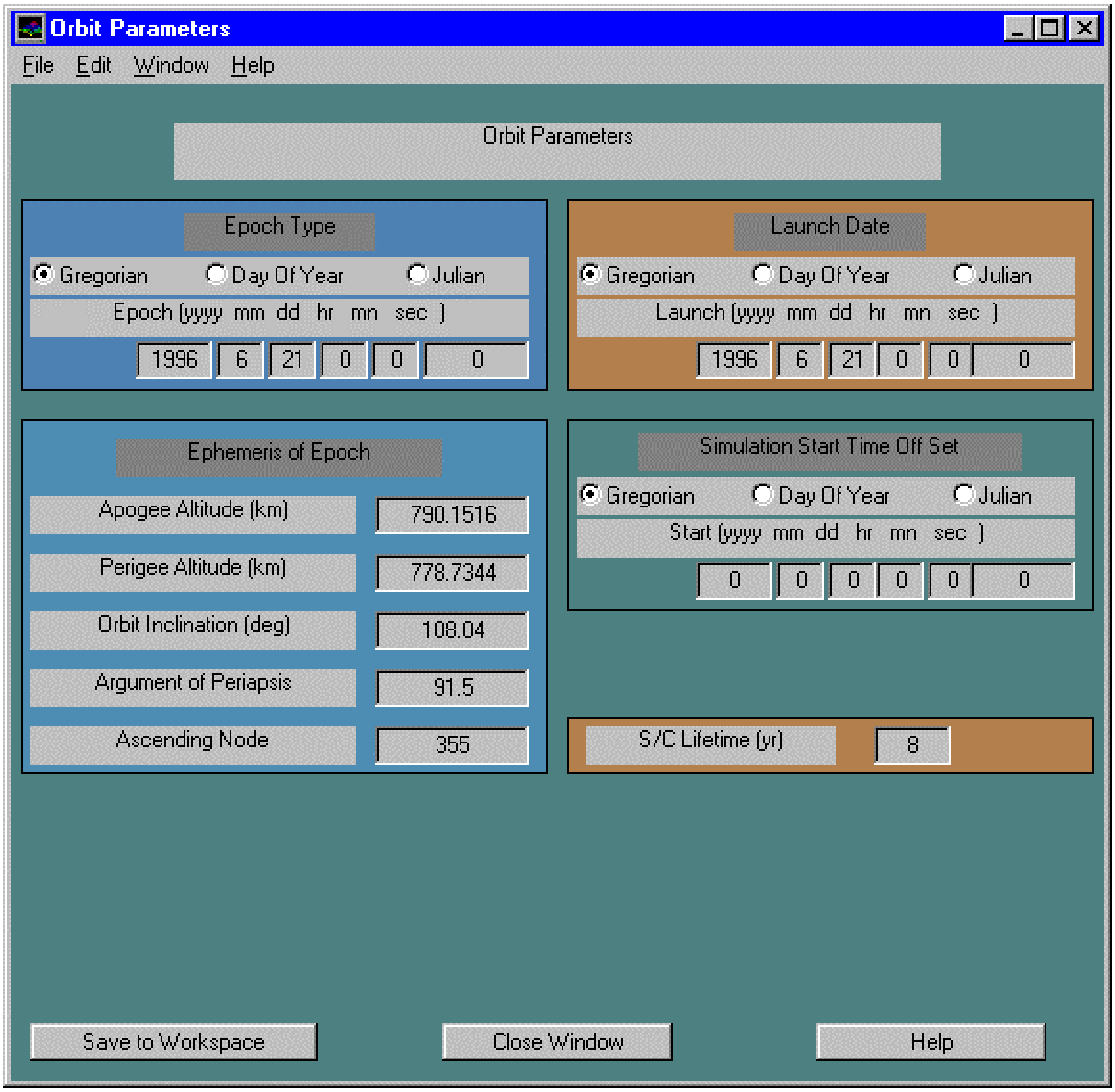


Figure 13: Orbit Parameters

The main CT menu, shown in Figure 15, includes parameters for the: transmitter, receiver, RF link, and typical ground station. It also includes flags to indicate whether Consultative Committee for Space Data Systems formatting and/or Reed-Solomon error coding are used. The transmitter parameters are: frequency, power, antenna beam width, antenna diameter, antenna type, line loss, antenna gain, and effective isotropic radiated power. The receiver parameters are: frequency, sensitivity, antenna beam width, antenna diameter, antenna type, antenna system noise temperature, antenna pointing error, and antenna gain. The RF link parameters are the uplink data rate and energy-per-bit to noise-density ratio (E_b/N_o), and the downlink data rate and E_b/N_o.

The parameters for a typical ground station are: antenna system noise temperature, antenna diameter, transmitter power, receiver noise bandwidth, minimum elevation, maximum time to acquire station, and maximum time to loss of signal.

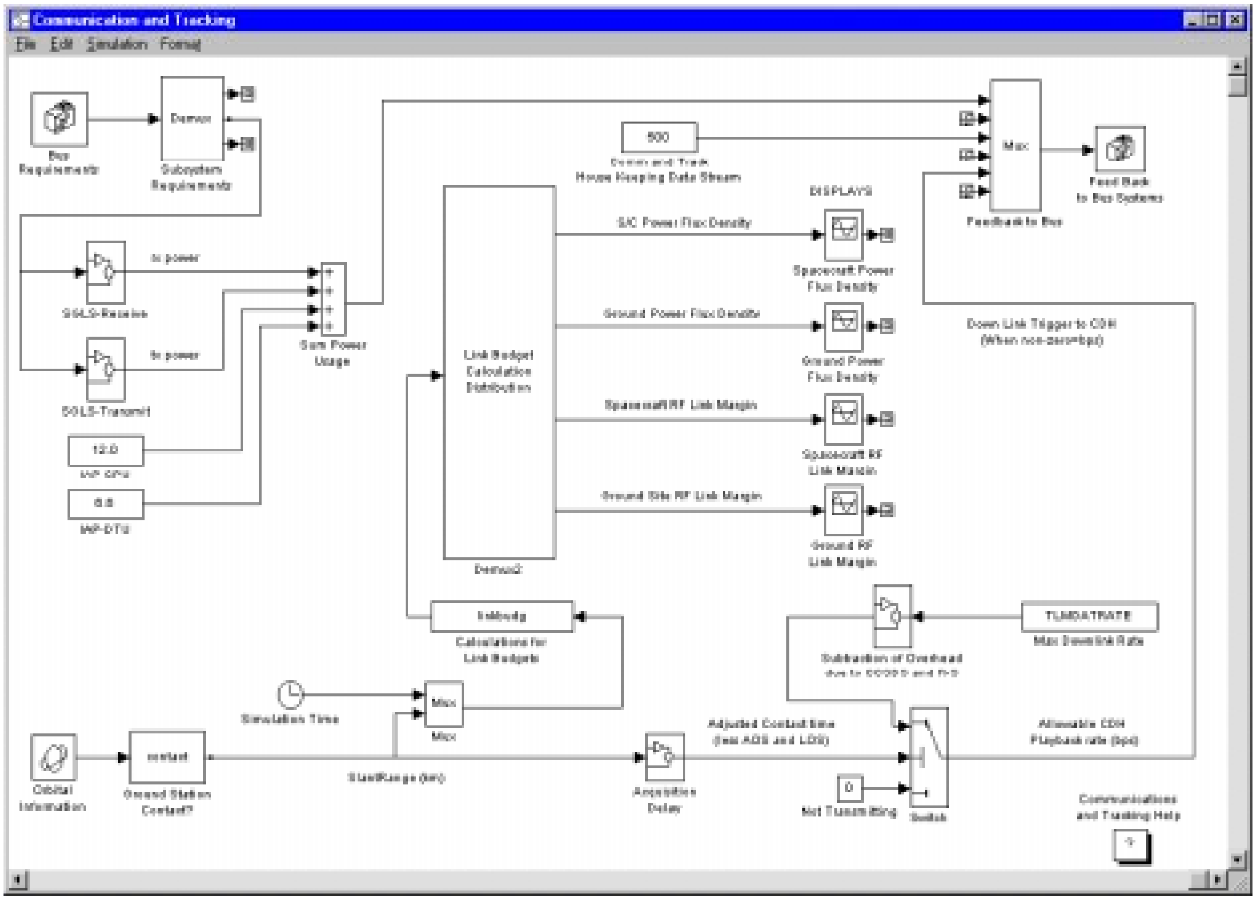


Figure 14: Communications and Tracking Subsystem

SPASIM also allows the user to select which ground stations are active through a ground station menu, shown in Figure 16. The other parameters in this menu are the ground station: name, latitude, longitude, and altitude. Currently there are 18 stations defined. The user can add or delete from this list through this menu.

The inputs to this model are the: radius, longitude, latitude, and net downlink rate from the CDH subsystem. This is the maximum rate at which the CDH subsystem can send data for downlinking. Its outputs are the power it requires, the rate at which it generates data, and the maximum net downlink rate. The maximum net downlink rate is the maximum rate at which the CT subsystem can accept data for downlinking. This model has predefined plots of time versus: downlink status, slant range, spacecraft power flux, spacecraft link margin, ground power flux, and ground link margin.

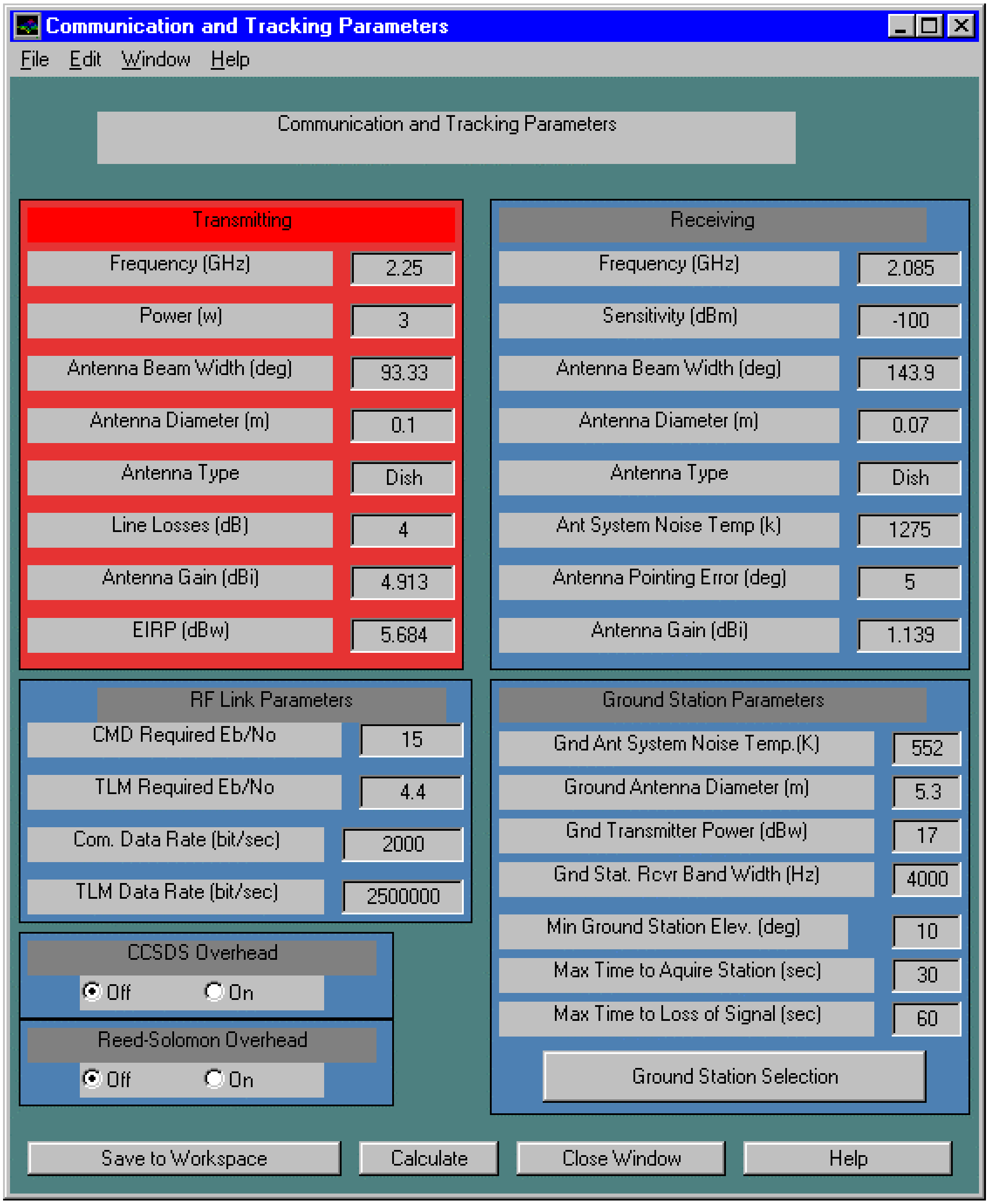
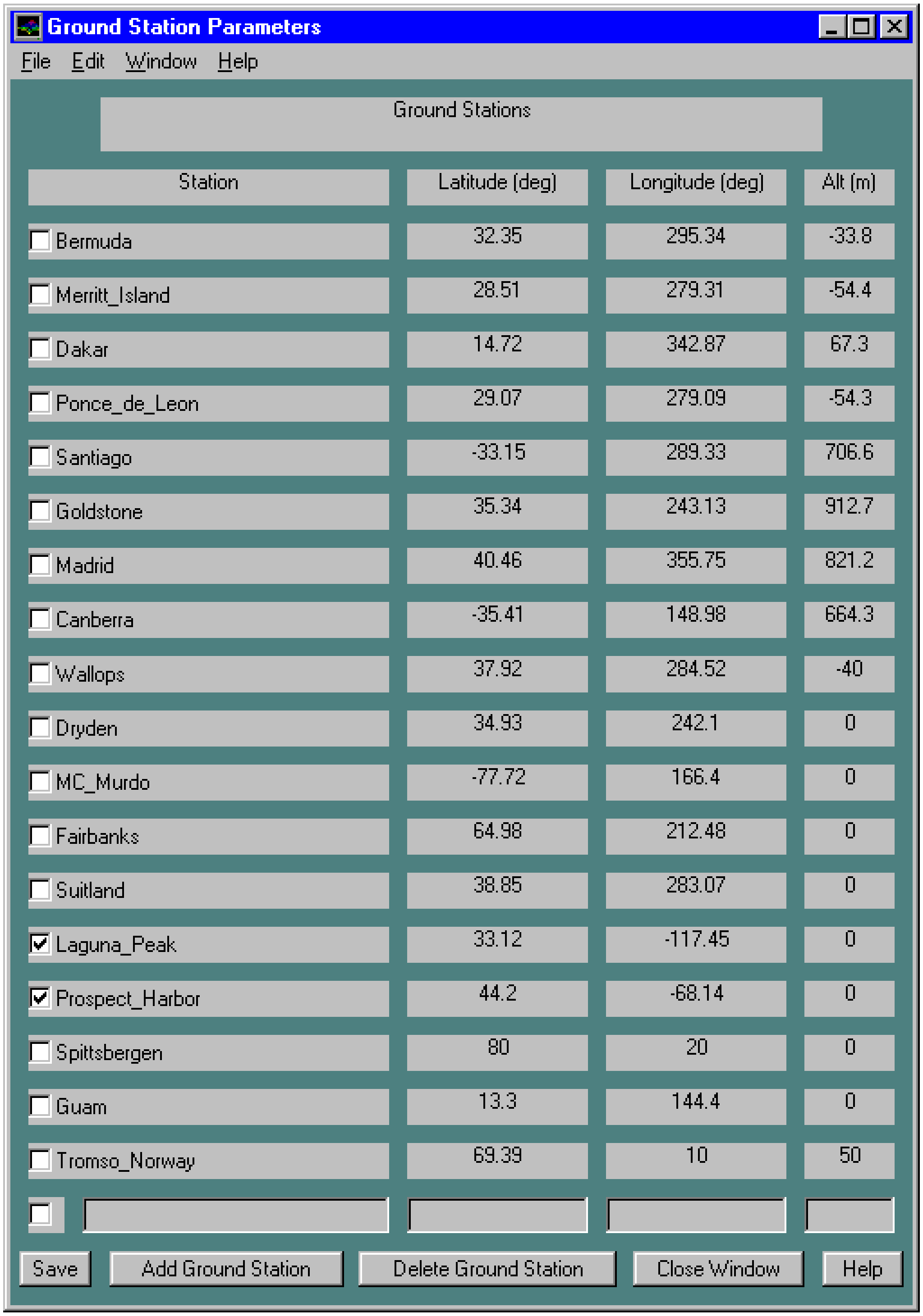


Figure 15: Communications and Tracking Parameters

Command and Data Handling

The purpose of the CDH subsystem, shown in Figure 17, is to store and retrieve data and commands and to execute those commands. Its model simulates the utilization of the data processing and storage capabilities and the requirements placed on other subsystems.



	Station	Latitude (deg)	Longitude (deg)	Alt (m)
☐	Bermuda	32.35	295.34	-33.8
☐	Merritt_Island	28.51	279.31	-54.4
☐	Dakar	14.72	342.87	67.3
☐	Ponce_de_Leon	29.07	279.09	-54.3
☐	Santiago	-33.15	289.33	706.6
☐	Goldstone	35.34	243.13	912.7
☐	Madrid	40.46	355.75	821.2
☐	Canberra	-35.41	148.98	664.3
☐	Wallops	37.92	284.52	-40
☐	Dryden	34.93	242.1	0
☐	MC_Murdo	-77.72	166.4	0
☐	Fairbanks	64.98	212.48	0
☐	Suitland	38.85	283.07	0
☑	Laguna_Peak	33.12	-117.45	0
☑	Prospect_Harbor	44.2	-68.14	0
☐	Spittsbergen	80	20	0
☐	Guam	13.3	144.4	0
☐	Tromso_Norway	69.39	10	50
☐				

Figure 16: Ground Station Parameters

The parameters in the CDH menu, shown in Figure 18, specify the: maximum net processing rate, maximum storage capacity, maximum record rate, maximum playback rate, pre-storage formatting overhead, pre-storage error coding overhead, sensor sampling rate, number of CDH analog sensors, number of CDH discrete sensors, unallocated data rate, unallocated instruction rate, and time stamp size. The maximum net processing rate is the number of instructions per second that the processor can allocate to the spacecraft's processing requirements. Operating system and other software overheads as well as required processor margins are deducted from the raw processor capability to get this number.

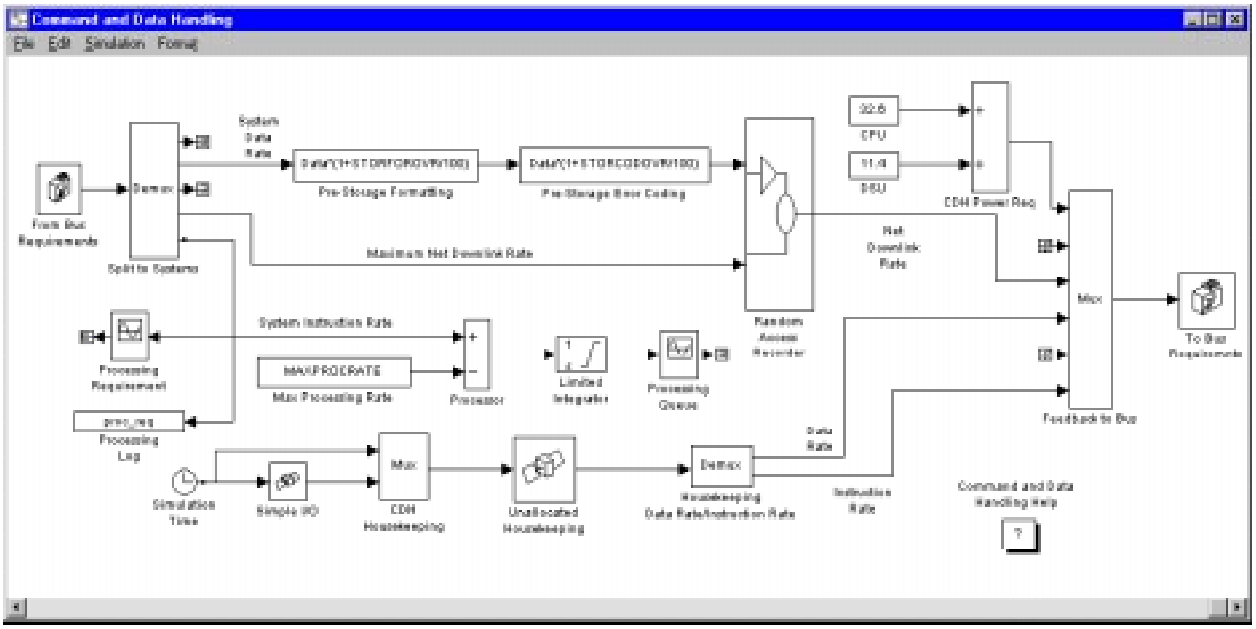


Figure 17: Command and Data Handling Subsystem

The sensor-sampling rate indicates how many times the spacecraft sensors need to be sampled per second. The data and instruction rates needed to support this function for the CDH subsystem are calculated based on this rate and how many analog and discrete sensors are in this subsystem. The unallocated data and instruction rates are the rates needed to support this function in the subsystems whose models do not calculate them. The time stamp size indicates the minimum number of bits needed to identify the time at which payload and housekeeping data was taken.

The inputs to this model are: the spacecraft data rate requirement, the spacecraft instruction rate requirement, and the maximum net downlink rate from the CT subsystem. The spacecraft data rate requirement is the rate at which payload and housekeeping data is generated. This data has to be stored until it can be downlinked.

The spacecraft instruction rate requirement is the number of instructions per second that the processor needs to execute to meet the requirements of the spacecraft. The rate at which the processor is executing instructions is subtracted from this input. The resulting rate is integrated to calculate the number of instructions queued.

The maximum net downlink rate is the maximum rate at which the CT subsystem can accept data for downlinking. This rate is zero when the spacecraft is not in contact with a ground station.

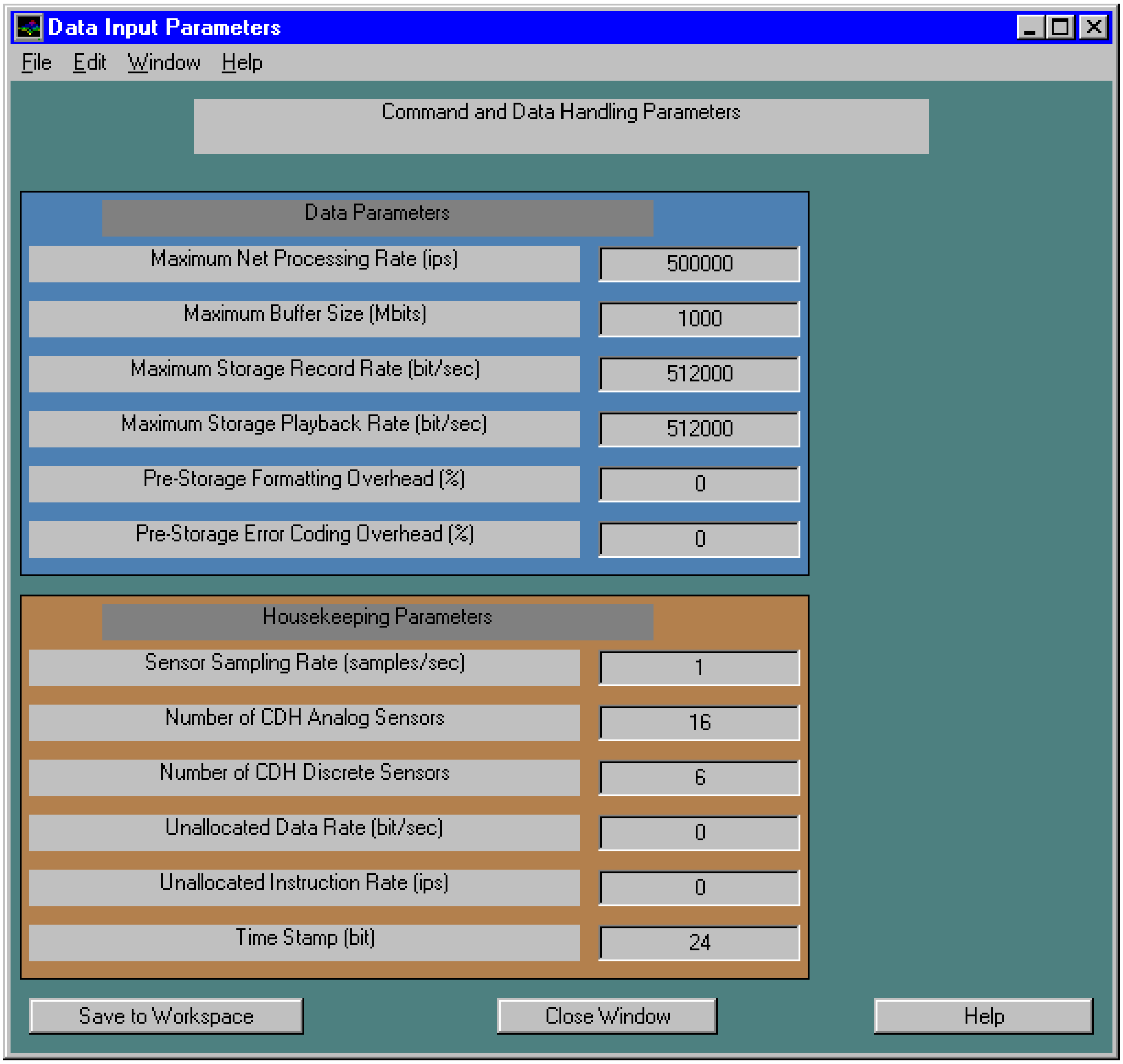


Figure 18: Command and Data Handling Parameters

This model has four outputs. Three of these are the contributions from this model to the spacecraft power requirement, the spacecraft data rate requirement, and the spacecraft instruction rate requirement. The fourth is the net downlink rate, which is the maximum rate at which the CDH subsystem can send data for downlinking. This model has predefined plots of time versus: data stored, spacecraft data rate requirement, data lost, spacecraft processing requirement, and processing queue.

RESULTS AND FUTURE WORK

On a 133 MHz Pentium laptop with Windows 95™ and 40 MB of RAM, SPASIM runs 26.67 times faster than real time. Six of SPASIM's 26 predefined plots are shown in the next six figures. Figure 19 shows the spacecraft's ground track. Figure 20 shows the slant range to a ground station. The data stored, which decreases when the spacecraft goes over a ground station, is shown in Figure 21. The power requirements, which increase when the spacecraft transmits

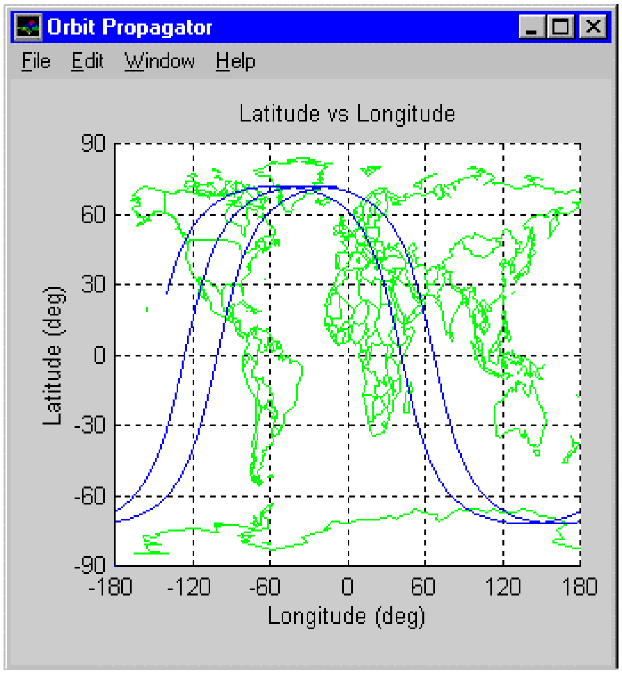


Figure 19: Orbit Track

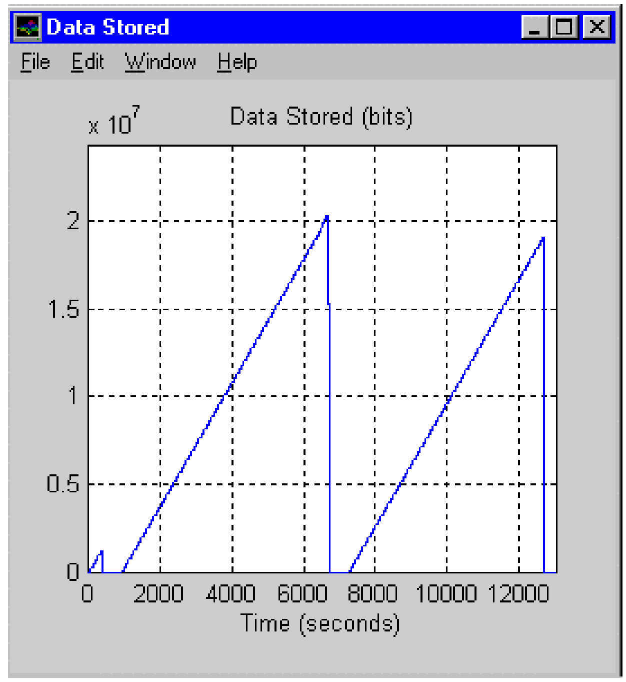


Figure 21: Data Stored

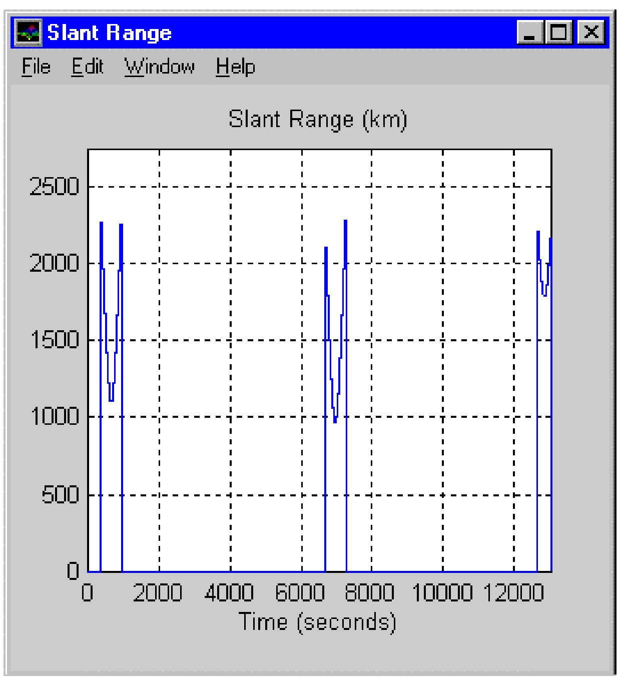


Figure 20: Slant Range

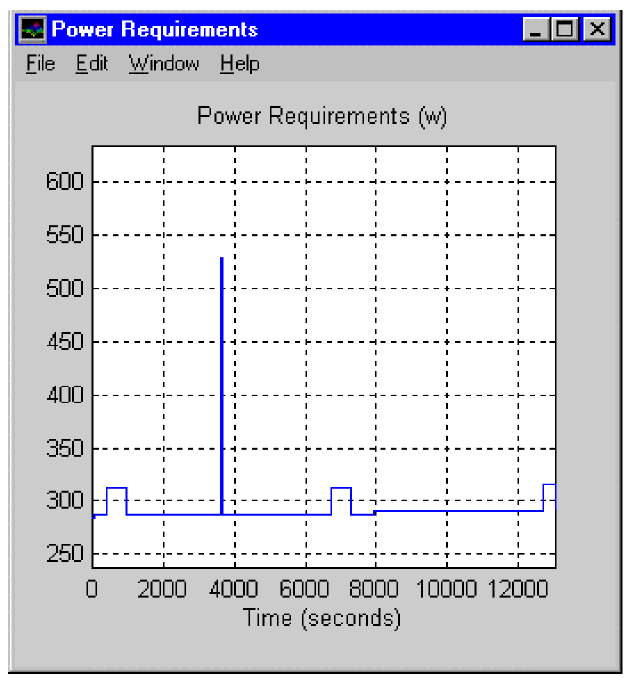


Figure 22: Power Requirement

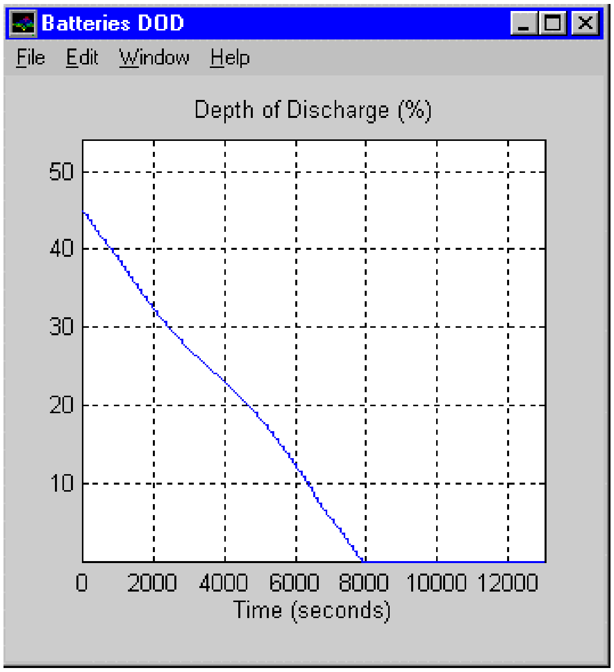


Figure 23: Depth of Discharge

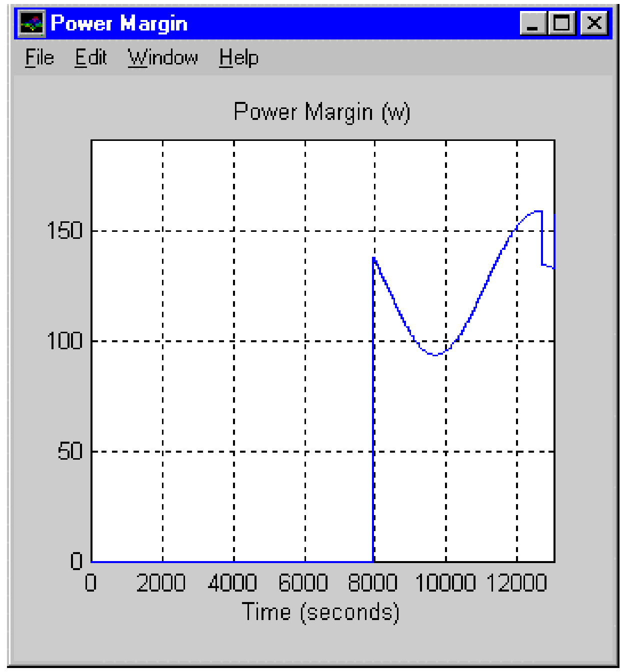


Figure 24: Power Margin

data, are shown in Figure 22. Figure 23 shows the DOD of the battery. The power margin, which is positive when the battery is fully charged, is shown in Figure 24.

SPASIM is now in beta testing at other NASA centers and within the U.S. aerospace industry. It is going to be validated further by applying it to as broad a range of spacecraft as possible and by comparing operational telemetry data with SPASIM predictions. It is also going to be expanded to crewed spacecraft like the space station and to planetary spacecraft.

SUMMARY

SPASIM can be used to validate spacecraft design and sizing estimates by performing an integrated time simulation of the spacecraft. This identifies resource bottlenecks or inadequacies resulting from simplified assumptions. Since SPASIM is a time based simulation, discrete events and duty cycles can be modeled and their resulting impacts can be assessed across all of the spacecraft. Failure modes and operational contingencies can be evaluated allowing the analyst to plan operations (what-if scenarios) and optimize the spacecraft performance for a range of mission scenarios. The SPASIM interface allows the analyst to easily change system functional architectures via block diagrams and to easily update performance characteristics of system components with parameter input menus. By changing specific parameters in a model, the user can assess the impacts of using different technologies.

SPASIM has been validated using several spacecraft designs that were at least at the Critical Design Review level. The user and programmer guide, including figures, is available on line as a hypertext document. This is an easy-to-use and expandable tool which is based on MATLAB® and SIMULINK®. It runs on Silicon Graphics Inc. workstations and personal computers with Windows 95™ or NT™.

ACKNOWLEDGEMENTS

SPASIM was developed for NASA by: R.F. Estes; J.T. Farmer; M.J. Ferebee, Jr.; G.G. Ganoe; P.A. Garn; M.L. Heck; C.A. Liceaga; Z.N. Martonavic; W.A. Sasamoto; R. Sinha; F.H. Stillwagen; C.C. Thomas; P.A. Troutman; and R. Van Valkenburg.

REFERENCES

Liceaga, C.A.; P.A. Troutman; R.F. Estes; J.T. Farmer; M.J. Ferebee, Jr.; G.G. Ganoe; P.A. Garn; M.L. Heck; Z.N. Martonavic; W.A. Sasamoto; R. Sinha; F.H. Stillwagen; C.C. Thomas; and R. Van Valkenburg. 1997. "SPASIM User's and Programmer's Guide." Technical Report. NASA LaRC, Hampton, VA.

Ferebee, M.J., Jr.; P.A. Troutman; and D.W. Monell. 1997. "Satellite Systems Design/Simulation Environment: A Systems Approach to Pre-Phase A Design." In *Proceedings of the 35th Aerospace Sciences Meeting and Exhibit* (Reno, NV, Jan.6-9). AIAA, Reston, VA.