APSCO CubeSat Competition

"Detailed Design Report"

Model In the Loop (MIL) Report

SPARCS Mission

IR-Sepehr (iSSS)

Jan 2025

1. Introduction

This report focuses on developing and implementing the Model-in-the-Loop (MIL) framework for the SPARCS mission, covering both system-level and subsystem-level aspects. For the modeling process, MATLAB Simulink has been utilized. The current specifications are based on SPARCS-B. However, the framework has been designed to allow easy adaptation for SPARCS-A with minimal modifications. The distinct parameters of the two CubeSats are configurable through the initialize script. The SPARCS MIL has been made available on SPARCS Mission GitHub[1].

2. System

First Layer of the SPARCS-B Model consists of three components

1. Model of Environment
2. Physical Domain
3. Software Domain.

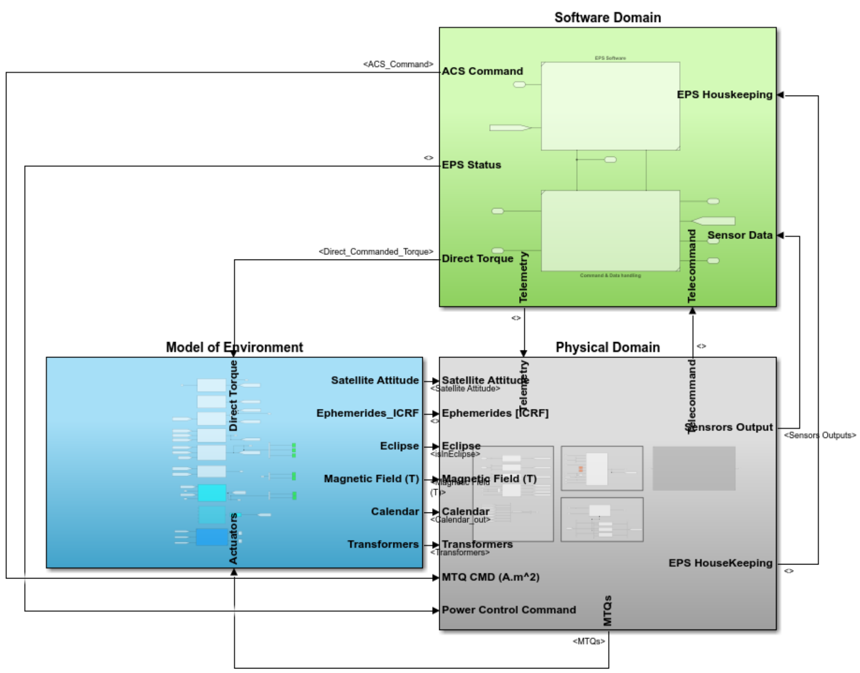

Figure 1- First Layer of the SPARCS-B Model

The **Model of Environment** simulates the space environment and all experiences SPARCS-B undergoes in its orbit. It includes modeling factors such as time, orbit, magnetic field, satellite kinetics and transformers, eclipse, satellite dynamics, disturbance torques, and environmental models relevant to the Electrodynamic Tether (EDT) payload. This enables the creation of an

[1] https://github.com/SPARCS-Mission

accurate space environment simulation for the satellite within the software. The environmental model is currently under development and becoming more refined.

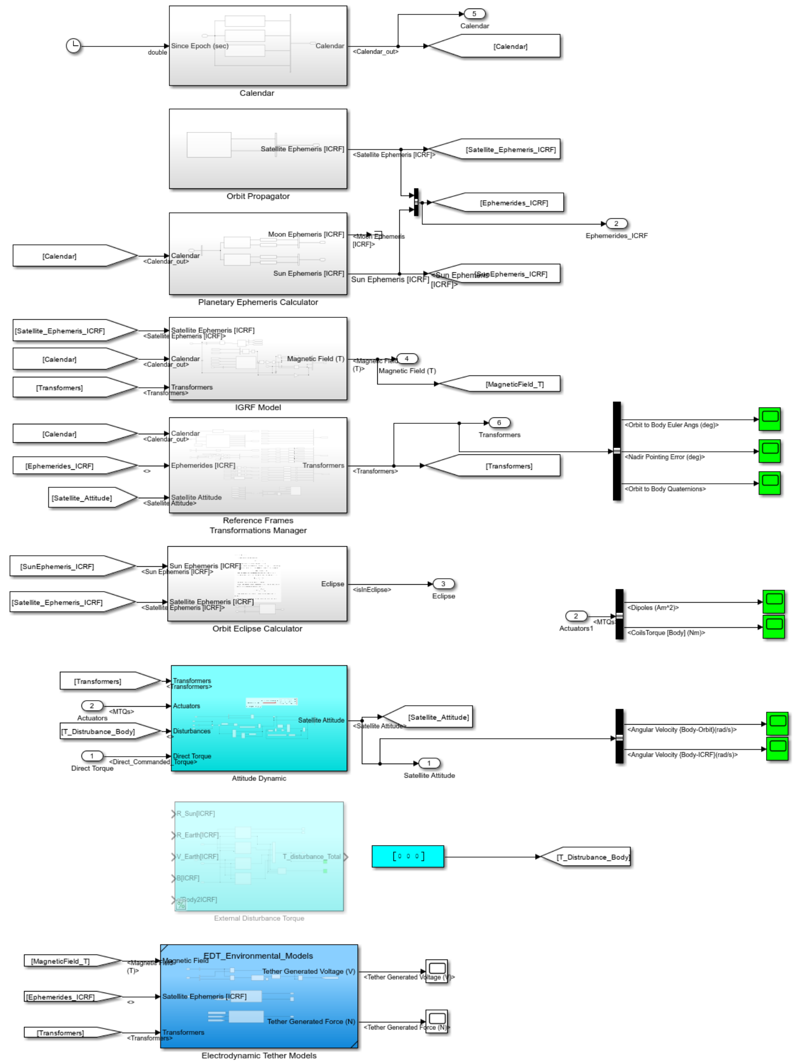

Figure 2- Model of Environment

The other two components, **Software Domain** and **Hardware Domain**, model the SPARCS-B CubeSat itself. In the **Hardware Domain**, everything that is physically present on the satellite is modeled. This includes power consumption of various components, sensors, actuators, communication systems(TT&C/ ISL), and the physical power subsystem (including the battery, PCDU, DC-converter, and solar cells).

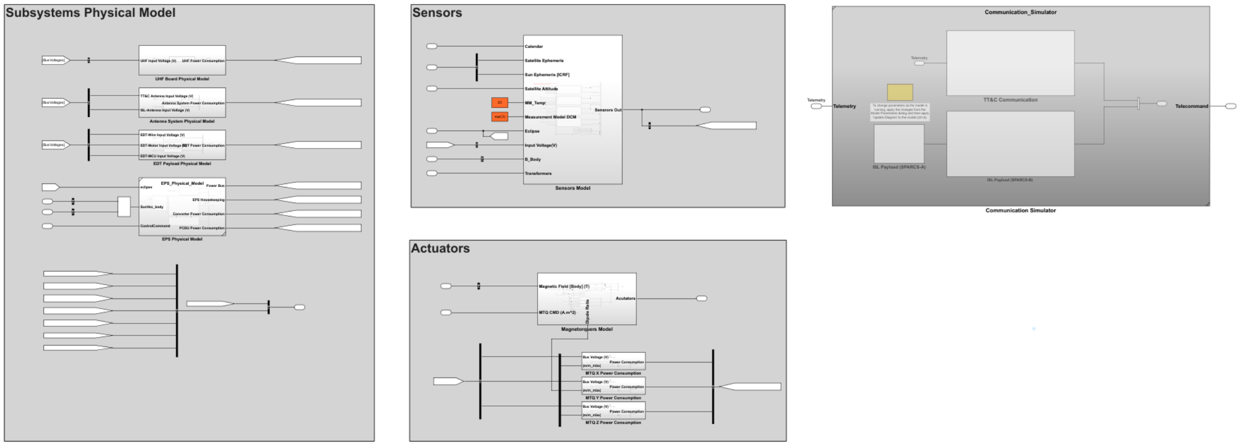

Figure 3- Physical Domain Model

Finally, the **Software Domain** encompasses the satellite's software simulation. Currently, it includes the EPS and C&DH software systems. It is important to note that the ADCS software, according to the SPARCS-B design, is integrated within the C&DH software, meaning there is no separate ADCS MCU or independent software for ADCS.

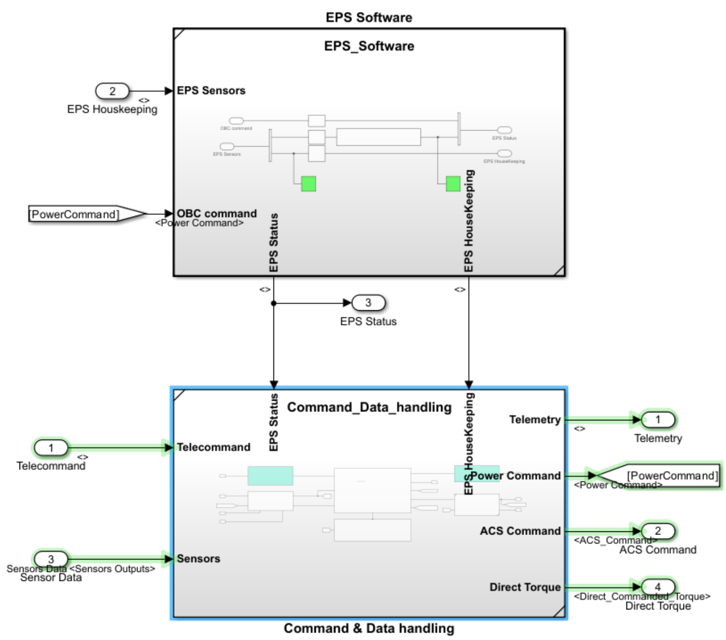

Figure 4- Software Domain Mode

3. Command & Data handling (C&DH)

The C&DH system has both physical and software models. C&DH physical model, within the **Physical Domain**, focuses solely on power consumption calculations. In the **Software Domain**, we have the C&DH software, which will be further discussed in terms of its external interfaces. Subsequently, the internal subsystems of the software and their respective internal interfaces will be covered.

3.1. C&DH Software external layer Interfaces

The **C&DH** subsystem is the central hub of the CubeSat's operation, enabling seamless communication between the satellite's various subsystems and its ground station. It processes incoming commands, manages the flow of telemetry and data, and ensures that the satellite operates efficiently and reliably. At the heart of this subsystem are several key interfaces that facilitate the exchange of information, each serving a unique purpose within the system.

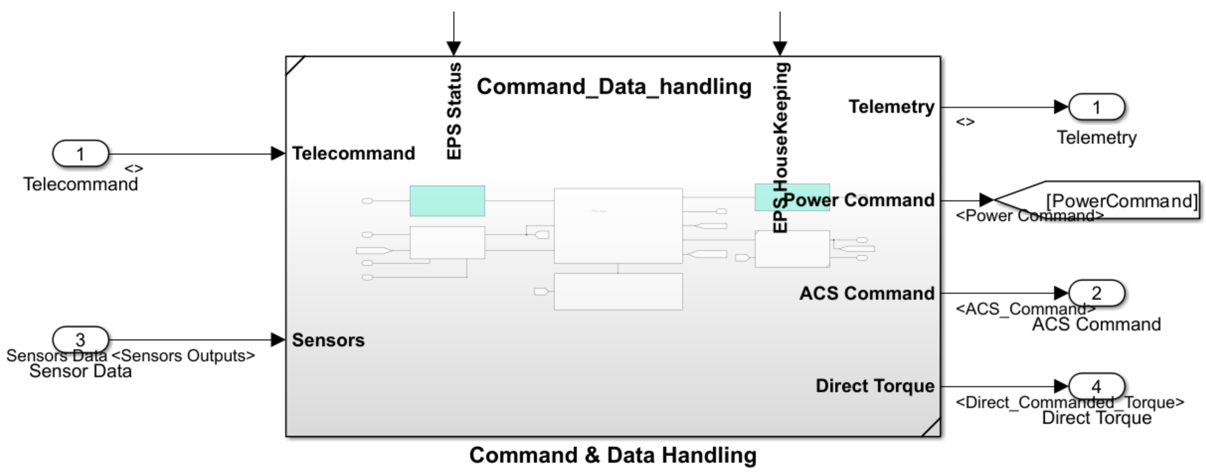

Figure 5- C&DH Software

3.1.1. Telecommand Interface

The telecommand interface is the primary channel through which the ground station communicates with the CubeSat. Telecommands, which are instructions sent from Earth, arrive at this interface and are subsequently decoded and executed by the C&DH subsystem. These commands may include tasks such as activating specific instruments, adjusting satellite orientation, or initiating data collection sequences. The telecommand interface plays a critical role in ensuring the CubeSat responds accurately and promptly to the mission's evolving requirements.

3.1.2. Sensor Data Interface

Another crucial input is the **sensor data interface**, which gathers information from the CubeSat's onboard sensors. These sensors monitor various parameters, such as temperature, orientation, and system health, providing real-time data critical for satellite operations. The C&DH subsystem processes this data, storing it for future transmission to the ground station or using it internally to make decisions, such as adjusting the satellite's attitude to optimize solar panel alignment or stabilize its orbit.

3.1.3. EPS status interface

The **EPS status interface** serves as a vital link between the Electrical Power Subsystem (EPS) and the C&DH subsystem. This interface provides updates on the power system's current status, including battery levels, energy generation rates, and overall power availability. By continuously monitoring the EPS status, the C&DH subsystem can make informed decisions to prioritize power usage, ensuring that critical systems remain operational even during periods of low energy availability.

3.1.4. Telemetry interface

On the output side, the **telemetry interface** is the conduit for transmitting processed data back to the ground station. This data includes health reports, mission data, and performance metrics collected and compiled by the C&DH subsystem. Telemetry ensures that the mission control team has a comprehensive understanding of the satellite's status and can make necessary adjustments to maintain optimal performance. Without this interface, ground operators would be unable to monitor or troubleshoot the CubeSat's operations effectively.

3.1.5. Power command interface

The **power command interface** is responsible for sending commands to the EPS to control the allocation and distribution of power across the satellite. Based on the EPS status data and telecommands, the C&DH subsystem determines the appropriate power settings and transmits them through this interface. For example, it might direct the EPS to prioritize powering critical systems during high-demand periods or to initiate a low-power mode to conserve energy.

3.1.6. ACS command interface

The **ACS command interface** connects the C&DH subsystem to the Attitude Control System (ACS). Using input from telecommands and sensor data, the C&DH subsystem calculates the necessary adjustments to the CubeSat's orientation and sends corresponding commands to the ACS. These commands ensure that the satellite maintains the correct orientation for tasks such as pointing its antennas toward Earth or aligning its solar panels with the Sun.

3.1.7. EPS housekeeping interface

Finally, the **EPS housekeeping interface** facilitates routine data exchange and control between the C&DH subsystem and the EPS. This interface is used for tasks such as toggling between different power modes, performing diagnostics, or conducting maintenance operations. It ensures that the EPS operates efficiently and is prepared to meet the CubeSat's power demands.

The operation of these interfaces is underpinned by a robust data exchange mechanism within the C&DH subsystem. Incoming data from sensors, telecommands, and the EPS status interface is processed and prioritized based on mission requirements and real-time system conditions. The subsystem's control logic ensures that critical operations, such as power management or orientation adjustments, are handled promptly. Meanwhile, output interfaces relay processed data and commands to their respective subsystems, completing the communication loop. Timing and synchronization are critical for these operations, as even minor delays could disrupt the CubeSat's functionality.

3.2. Software Internal layer Interfaces

The **C&DH Software** is composed of several interconnected blocks that work together to process commands, manage data, and communicate with the satellite's other subsystems. These internal components are responsible for ensuring that the CubeSat operates efficiently by executing commands, managing telemetry, and controlling subsystems like the EPS and ACS.

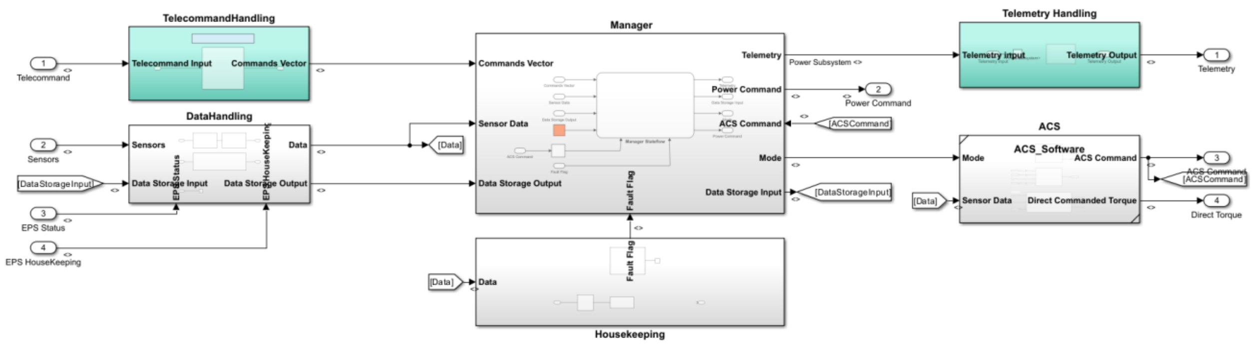

Figure 6- C&DH Software

3.2.1. Telecommand Handling

The **Telecommand Handling** block is responsible for processing incoming commands from the ground station. It receives raw telecommands via the **Telecommand Input** and decodes them into actionable instructions, represented as a **Commands Vector**. These instructions are forwarded to the **Manager** block for further processing. This component ensures that the CubeSat interprets ground station commands correctly, enabling precise control over mission operations.

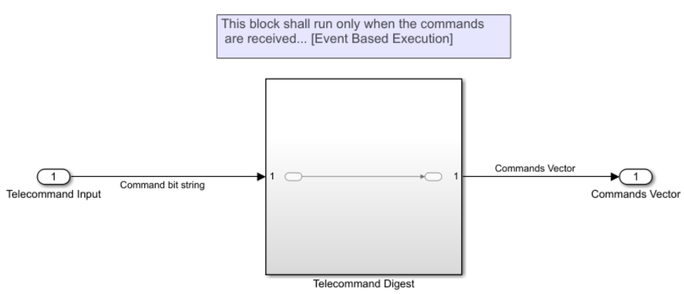

Figure 7- Telecommand Handling block

The "Telecommand Digest" block in this Simulink model is designed to process incoming telecommand signals. It receives telecommands as a bit string through its input port, which originate from a higher-level communication or telemetry system. The purpose of this block is to decode and interpret the received bit strings, converting them into a structured command vector format that can be understood and utilized by other subsystems. The block operates based on event-driven execution, meaning it only activates when new telecommands are received, rather than running continuously. This event-based approach is efficient, as it conserves computational resources by processing data only when necessary.

Internally, the block likely includes logic for parsing and validating the telecommands. It ensures that the incoming data is accurate and actionable before converting it into a command vector. This output vector serves as a structured representation of the telecommands, mapping them to specific actions or functions within the system. Given its critical role, the block might also include error-handling mechanisms to deal with invalid or corrupted inputs. This functionality would be essential in applications such as satellites or spacecraft, where reliable telecommand interpretation is vital for mission success.

3.2.2. Data Handling

The **Data Handling** block serves as the core data management unit within the C&DH subsystem. It aggregates sensor data, EPS status updates, and other inputs to ensure data is available for decision-making and telemetry. Inputs from the **Sensors** and **EPS Status** are routed here, and the block also interacts with a **Data Storage Input/Output** mechanism. This storage component allows the CubeSat to save critical data for later use, especially during communication blackouts or high-priority mission tasks. The processed data is sent to the **Manager** block for integration into telemetry or command decision-making.

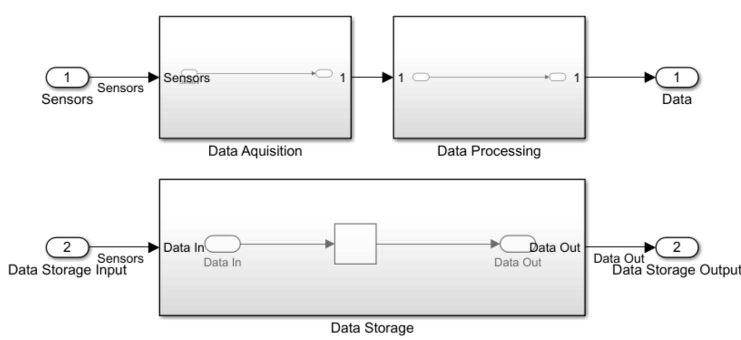

Figure 8- Data handling Block

The Simulink diagram represents a system for acquiring, processing, and storing data, in the context of a sensor-driven application. The flow starts with the "Data Acquisition" block, which takes input from sensors. This block is responsible for collecting raw data from the sensors and preparing it for subsequent stages, such as ensuring the data is in a format suitable for processing. Once the data is acquired, it flows into the "Data Processing" block, where the raw sensor data undergoes transformations, calculations, or filtering to produce meaningful and usable output. The processed data is then outputted as a structured result, labeled "Data," for use in further operations or systems.

Below these two blocks lies the "Data Storage" block, which appears to function independently but serves a complementary role in the system. This block accepts data as input, and stores it for future use. The storage system may include mechanisms for organizing and retrieving the stored data efficiently, ensuring long-term availability and integrity. The stored data can then be accessed through the "Data Out" port, which provides the output . This configuration suggests a system designed for applications requiring both real-time data processing and historical data storage, such as monitoring systems or data logging for analysis and diagnostics.

3.2.3. Manager

The **Manager** block is the central coordination unit of the C&DH subsystem. It integrates data and commands from all other blocks, processes them, and determines the appropriate outputs for the ACS, EPS, and telemetry. For example, the **Commands Vector** from the Telecommand Handling block is combined with sensor data and EPS status to generate control signals such as **ACS Command**, **Power Command**, and **Telemetry**. The Manager also manages the data storage flow, ensuring that critical data is stored or retrieved from the Data Handling block as needed.

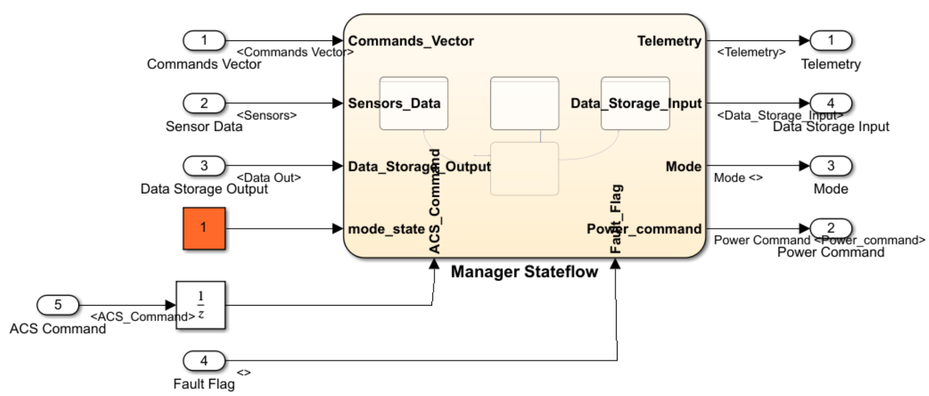

Figure 9- Manager Block

The "Manager Stateflow" block shown in this Simulink diagram acts as the **Mode Manager** for a CubeSat, playing a central role in determining the satellite's operational mode based on input commands and system states. This block integrates various inputs, such as commands, sensor data, and storage outputs, and outputs critical signals that govern the CubeSat's behavior.

The **Commands Vector** input represents a set of high-level commands sent to the satellite, which guide its operations and mode transitions. These commands are processed within the

Stateflow block, where state machine logic likely governs the transition between different modes of operation. Examples of modes could include standby, science operations, fault recovery, or safe mode.

Sensor Data provides real-time information about the CubeSat's environment or internal health, which can influence mode transitions. Similarly, the **Data Storage Output** input represents previously stored data that might be used for decision-making or telemetry.

The **Fault Flag** input is critical for safety; it signals issues in the system, such as hardware malfunctions or adverse environmental conditions. In response, the Stateflow logic may force a transition to a safe mode to protect the satellite. Additionally, the **ACS Command** (likely Attitude Control System) input helps ensure that the CubeSat aligns its operations with positional requirements, which could also influence mode selection.

The block outputs several key signals:

1. **Telemetry**: This includes data sent back to ground control, such as the CubeSat's current mode, status, and any relevant information from other subsystems.
2. **Data Storage Input**: Processed data ready for storage, ensuring efficient logging of operations and events.
3. **Mode**: This defines the CubeSat's current operational mode, actively controlled by the logic within the Manager Stateflow.
4. **Power Command**: A signal managing power allocation to various subsystems based on the current mode, ensuring optimal energy usage.

The Stateflow logic within the Manager implements a decision-making framework that dynamically adjusts the CubeSat's operations. For instance, if the Commands Vector specifies "science mode" and no faults are detected, the CubeSat will transition into its already dictated flow. However, if a fault is flagged, the system may override the command and shift to safe mode, prioritizing the satellite's integrity.

3.2.4. Telemetry Handling

The **Telemetry Handling** block is responsible for formatting and transmitting processed data back to the ground station. It receives telemetry inputs from the **Manager** and packages them into the appropriate format for the **Telemetry Output** interface. This block ensures that mission-critical information, such as system health and performance metrics, is relayed to Earth in a timely and organized manner.

3.2.5. ACS

The **ACS (Attitude Control System)** block interfaces with the Manager to receive commands for adjusting the CubeSat's orientation. The Manager provides the **ACS Command** signal, which is calculated based on sensor data and telecommands. The ACS block interprets these commands and adjusts the satellite's orientation accordingly, ensuring proper alignment with mission goals such as maintaining communication or optimizing solar power collection.

3.2.6. Housekeeping

The **Housekeeping** block is responsible for monitoring and managing routine CubeSat functions. It interacts with the Manager to handle data storage and system diagnostics. The block also processes inputs related to fuel levels or system status and ensures this data is accessible for telemetry or internal decision-making. This component helps maintain the CubeSat's operational health by monitoring its resources and systems.

3.2.7. Data Flow and Inter-Subsystem Communication

The C&DH subsystem operates through a series of interconnected data flows between its internal components:

- **Telecommand Handling** processes incoming commands and forwards them to the Manager as a **Commands Vector**.
- **Data Handling** aggregates sensor data and EPS status, stores critical information, and provides it to the Manager for decision-making.
- The **Manager** processes all incoming data and generates outputs such as telemetry, ACS commands, and power commands.
- **Telemetry Handling** packages processed data for transmission to the ground station.
- **ACS** adjusts the satellite's orientation based on commands from the Manager.
- **Housekeeping** ensures routine operations are monitored and that system diagnostics are available.

The system's modular design allows for efficient processing of commands and data, ensuring the CubeSat performs its mission objectives reliably.

4. Electrical Power System (EPS)

The EPS subsystem comprises two main sections;

1. software responsible for communication with the C&DH and other EPS sections
2. hardware simulation, which includes solar panels, a battery, a Power Conditioning and Distribution Unit (PCDU), and a DC-DC converter.

we will describe the hardware model of the EPS and the EPS software model, followed by the external interfaces of the EPS.

4.1. EPS Physical Model

The figure below illustrates the physical model of the EPS and its interfaces.

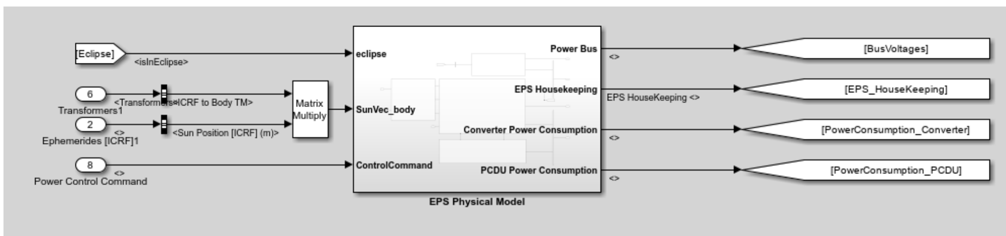

Figure 10- EPS Physical Model

The figure below also shows the hardware components of the EPS, which are included in the physical model of the EPS.

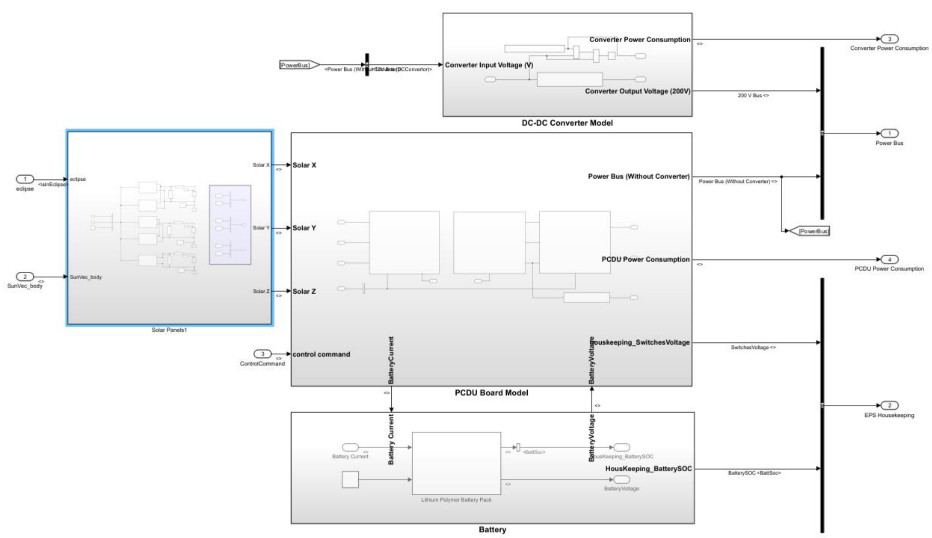

Figure 11- EPS Hardware Model

4.1.1. Solar Panel Model

The solar panel section simulates energy generation by CubeSat-mounted solar arrays. Each solar panel model considers incident irradiance, eclipse conditions, and the satellite's orientation (via the Sun vector). As shown in the simulation image, the model incorporates multiple solar panels (e.g., +X, -X, +Y, -Y, and Z panels) that convert solar energy into electrical power. The current-voltage (I-V) characteristics are modeled, with resistive loads representing the operational circuitry. The outputs from each panel include voltage and current parameters, which are fed into the PCDU board for further processing.

Key Features:

- Sun vector calculations ensure accurate modeling of incident irradiance on the solar panels.
- Real-time adjustment for orbital eclipse conditions, enhancing simulation fidelity.
- Voltage and current outputs are monitored to simulate realistic operational conditions.

Interfaces: Each solar panel interfaces with the PCDU by providing real-time voltage and current outputs. These signals are used for energy management and battery charging. Telemetry data, such as panel efficiency and output power, are also sent to the OBC for monitoring and optimization.

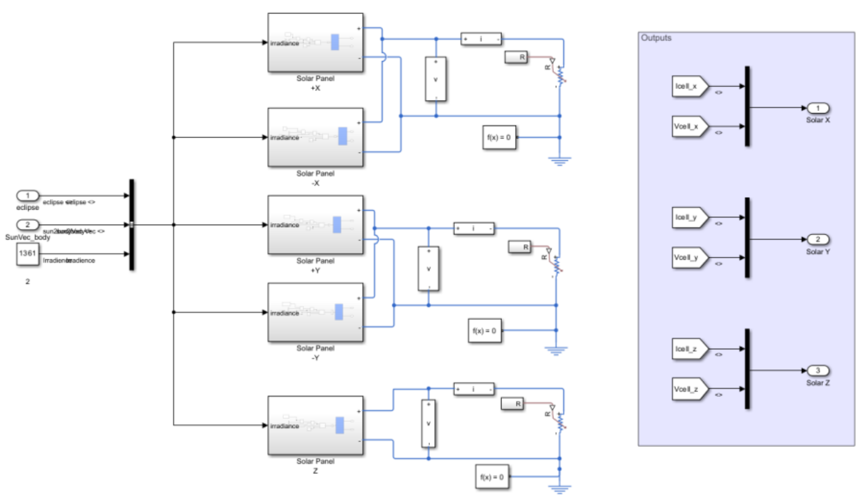

Figure 12- Solar Panels Model

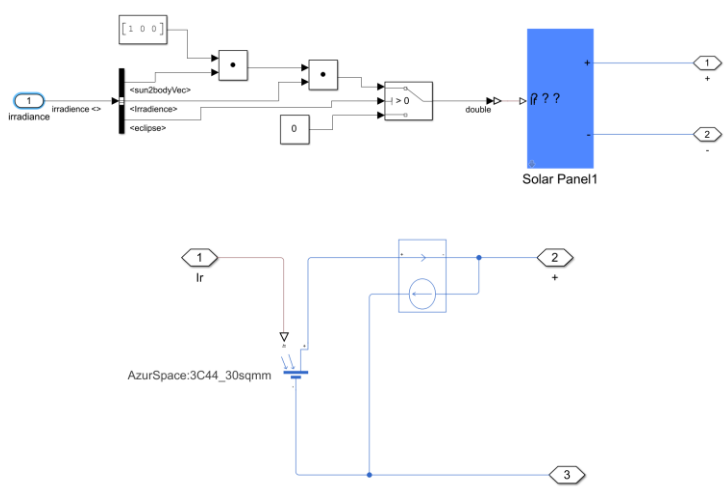

Figure 13- Each solar panel model

4.1.2. PCDU Board Model

The PCDU board serves as the central hub for power regulation and distribution. The simulated PCDU integrates inputs from all solar panels and manages battery charging and voltage regulation. As illustrated in the model, it consists of three primary modules.

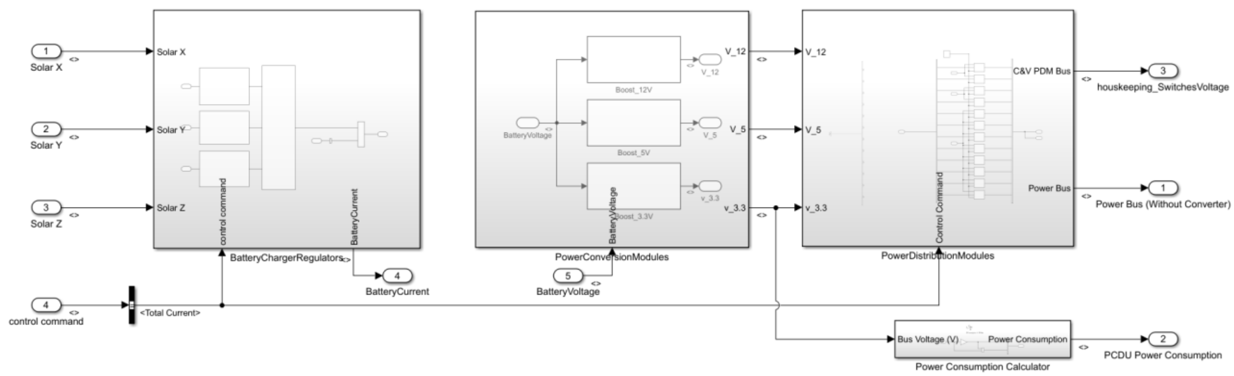

Figure 14- PCDU board model

- Battery Charger Regulators: These regulate current from the solar panels to charge the onboard battery efficiently while protecting it from overcharging. Protection circuits include maximum current limits and temperature-based adjustments.

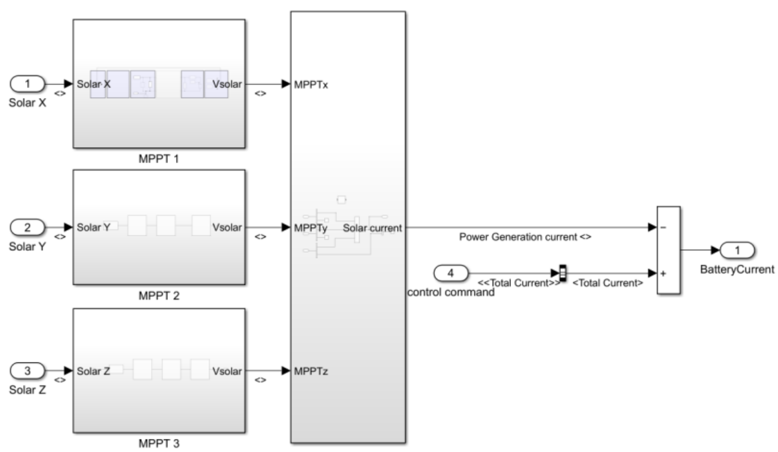

Figure 15- BCR Model

- Power Conversion Modules: These modules generate standard voltage levels required by the CubeSat's subsystems, including 3.3V, 5V, and 12V buses. The simulation includes voltage stability analyses and load adaptability tests.

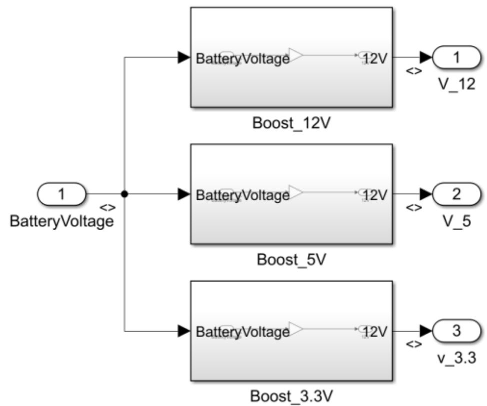

Figure 16- PCM Model

- Power Distribution Modules: These manage the distribution of power to various loads and subsystems via switches and control buses. The PCDU also monitors and communicates voltage and current parameters to the EPS microcontroller for housekeeping purposes. Fault detection mechanisms are incorporated to identify and isolate defective lines or loads.

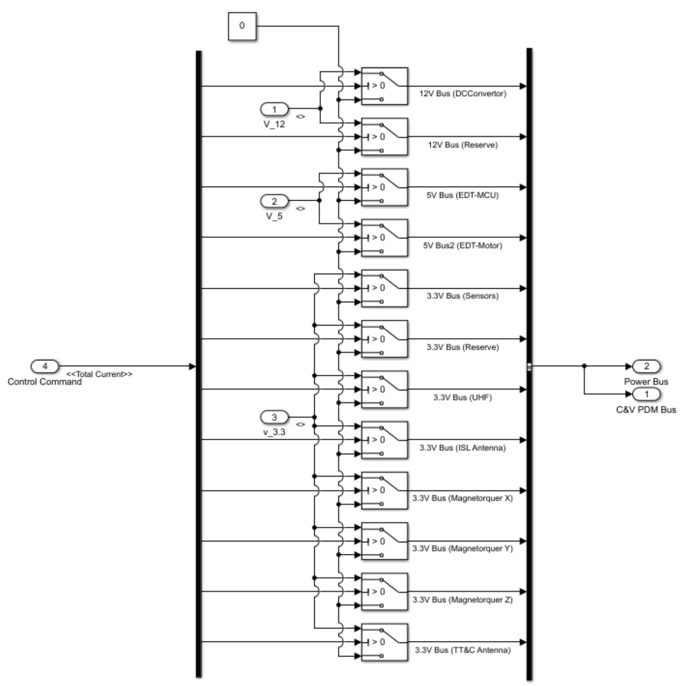

Figure 17- PDM Model

Interfaces: The PCDU interfaces with the battery to regulate charging and discharging. It also communicates with the OBC, providing status updates on voltage, current, and fault conditions. Additionally, it connects to other subsystems, ensuring power allocation according to mission priorities.

4.1.3. Battery Model

The battery model simulates the storage and discharge of electrical energy under varying loads. Key parameters such as state-of-charge (SOC), charge/discharge efficiency, and thermal effects are included. The battery interacts dynamically with the PCDU board to maintain stable power delivery during eclipse phases or peak load conditions.

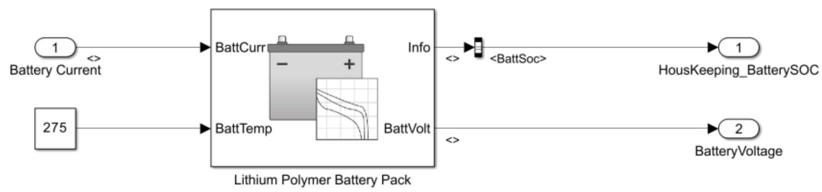

Figure 18- Battery pack Model

Key Features:

- Dynamic SOC tracking, which adjusts based on real-time charging and discharging cycles.
- Efficiency modeling accounts for internal resistance and thermal impacts.
- Emergency modes simulate responses to critical SOC levels, prioritizing essential subsystems.

Interfaces: The battery connects to the PCDU, receiving regulated charging currents and supplying power during low-generation periods. SOC data is continuously sent to the OBC for monitoring. Fault signals, such as over-temperature or under-voltage conditions, are flagged for immediate response.

4.1.4. DC-DC Converter Model

The DC-DC converter is a critical component that steps up the 12V supply to 200V, required for high-voltage payloads. The simulation includes:

- Input Regulation: Ensuring consistent input voltage despite variations from the solar panels or battery.
- Output Voltage Control: Maintaining a steady 200V output with minimal ripple. Control algorithms dynamically adjust switching frequencies to optimize efficiency.

Interfaces: The DC-DC converter receives input power from the PCDU and delivers high-voltage output to payload systems. Feedback loops ensure precise control and stability. Monitoring outputs are shared with the OBC for fault analysis and performance optimization.

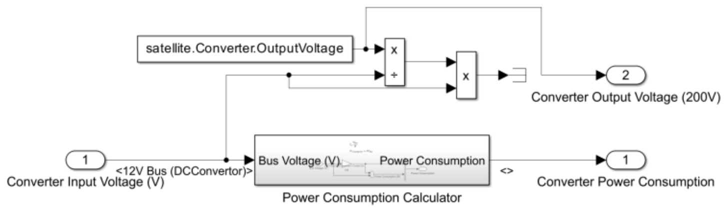

Figure 19- DC-DC converter model

4.2. Software and Control Integration

The software section of the EPS simulation ensures seamless communication between the OBC and hardware components. Control commands from the OBC regulate the PCDU's switching mechanisms, manage battery charging cycles, and activate/deactivate the DC-DC converter as needed. Feedback loops in the simulation verify the accuracy of command execution and fault tolerance.

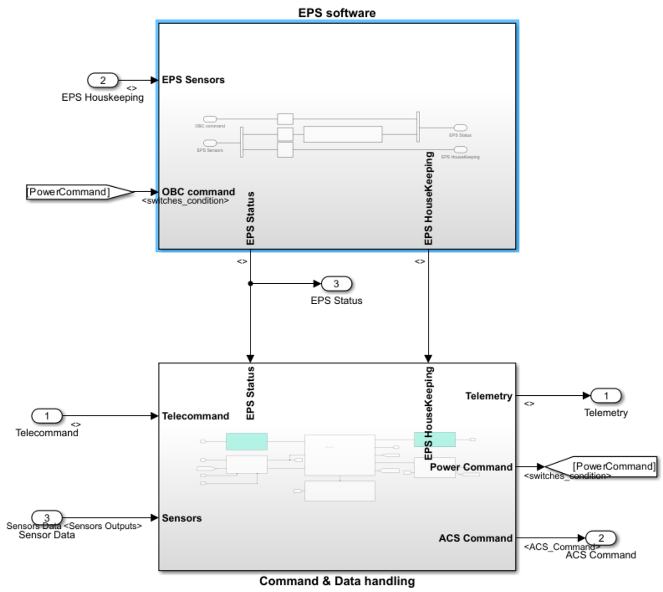

Figure 20- Software model

Interfaces: The software interfaces include:

- Command lines from the OBC to the EPS for power management.
- Data feedback to the OBC, including voltage levels, SOC, and fault indicators.
- Status updates shared with the thermal control and ADCS subsystems to align power availability with operational demands.

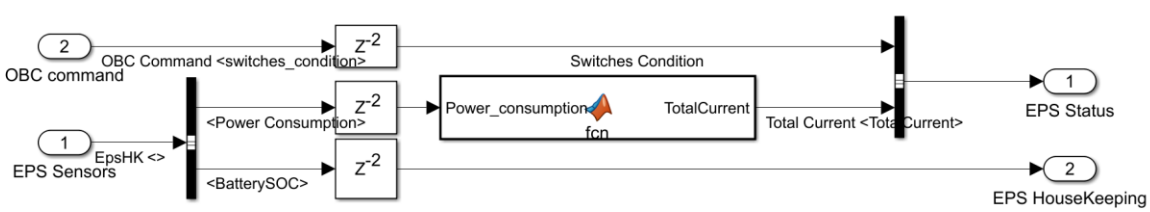

Figure 21- EPS Software Model

4.3. External Interfaces with other Subsystems

The EPS subsystem interfaces with other CubeSat subsystems, ensuring uninterrupted power delivery for critical operations. Key integrations include:

1. Attitude Determination and Control System (ADCS): The EPS supplies power to ADCS components like reaction wheels and magnetorquers, ensuring stability and orientation control. In return, the ADCS provides sun vector data, optimizing solar panel alignment.
2. Communication Subsystem: Stable voltage and current levels are provided to communication modules for consistent uplink and downlink operations. The EPS also prioritizes power for high-demand communication windows.
3. Payload: High-voltage and low-voltage buses from the EPS power scientific instruments and mission-specific payloads. Power budgeting ensures that payload operations do not compromise essential subsystems.
4. Thermal Control: Data from the EPS, such as power consumption and thermal outputs, are shared with the thermal control subsystem to optimize CubeSat temperature management. The thermal subsystem, in turn, prevents overheating of EPS components.

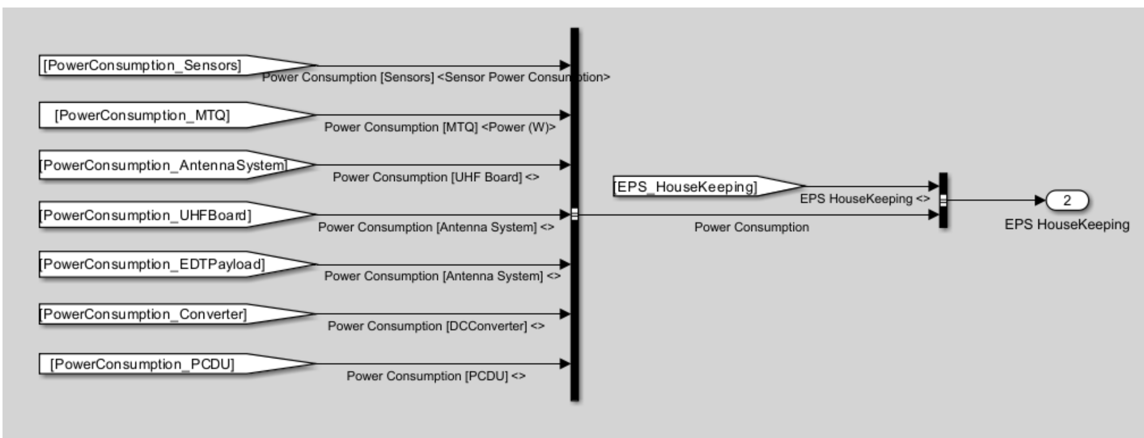

Figure 22- EPS HouseKeeping Model

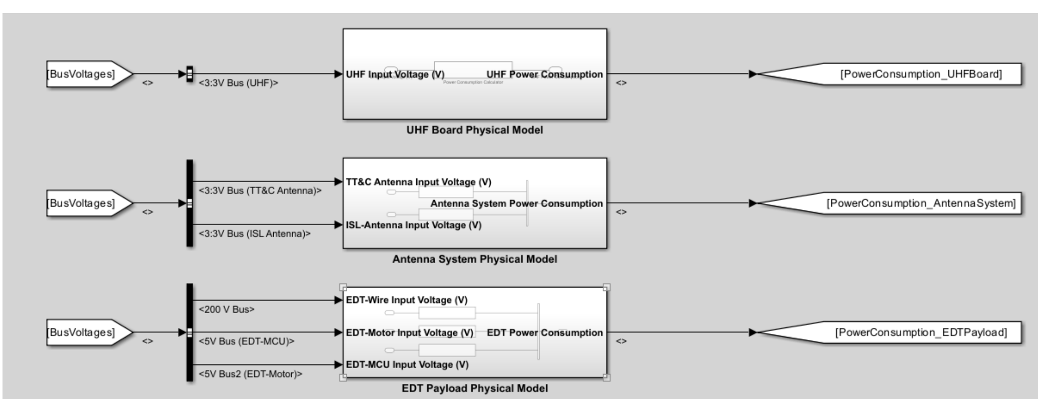

Figure 23- Bus voltages and power consumption models

4.4. Simulation Results

The MIL simulation validates the performance of the EPS subsystem under various scenarios, including sunlight, eclipse, and dynamic load conditions. The figures below show the total current and Battery SOC during the first 150 seconds of the Detumbling scenario.

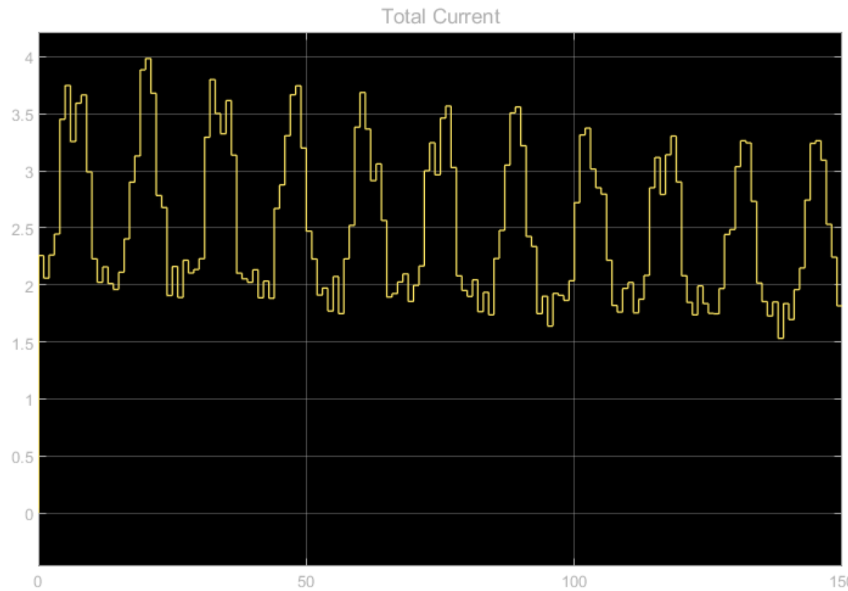

Figure 24- total consumption current results

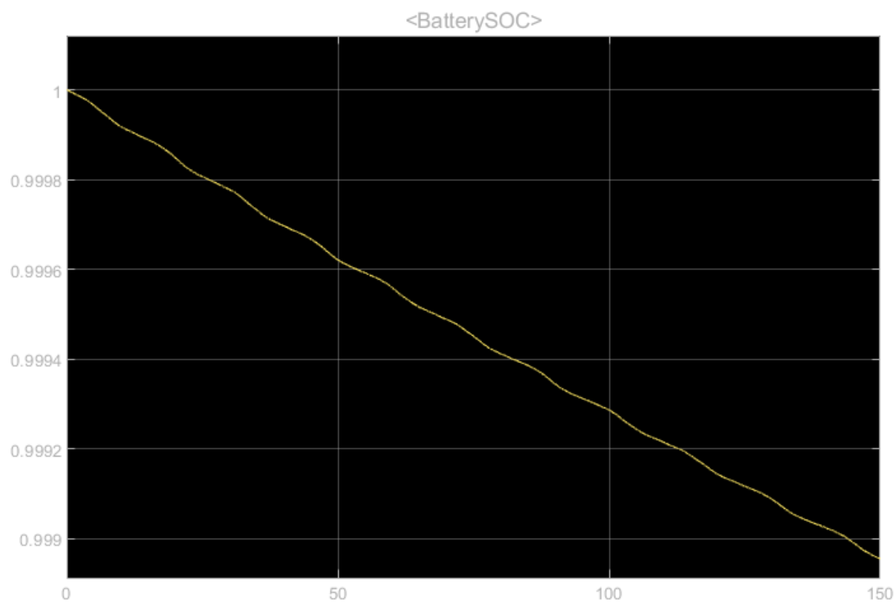

Figure 25- SOC of battery packs results

5. Attitude Determination and Control Subsystem (ADCS)

The ADCS subsystem model consists of components in both the **Physical Domain** and the **Software Domain**. In the **Physical Domain**, the models for sensors and actuators are included. In the **Software Domain**, the attitude determination and control software is implemented.

Next, we will first discuss these two sections of the ADCS model and their intefaces with other subsystem, at the end, review the simulation results.

5.1. Physical Model of ADCS

The Physical Domain of the ADCS includes models for the sensors and actuators.

5.1.1. Sensors Model

The figure below shows the models for the ADCS sensors, including the sun sensor, magnetometer, gyroscope, and GNSS receiver.

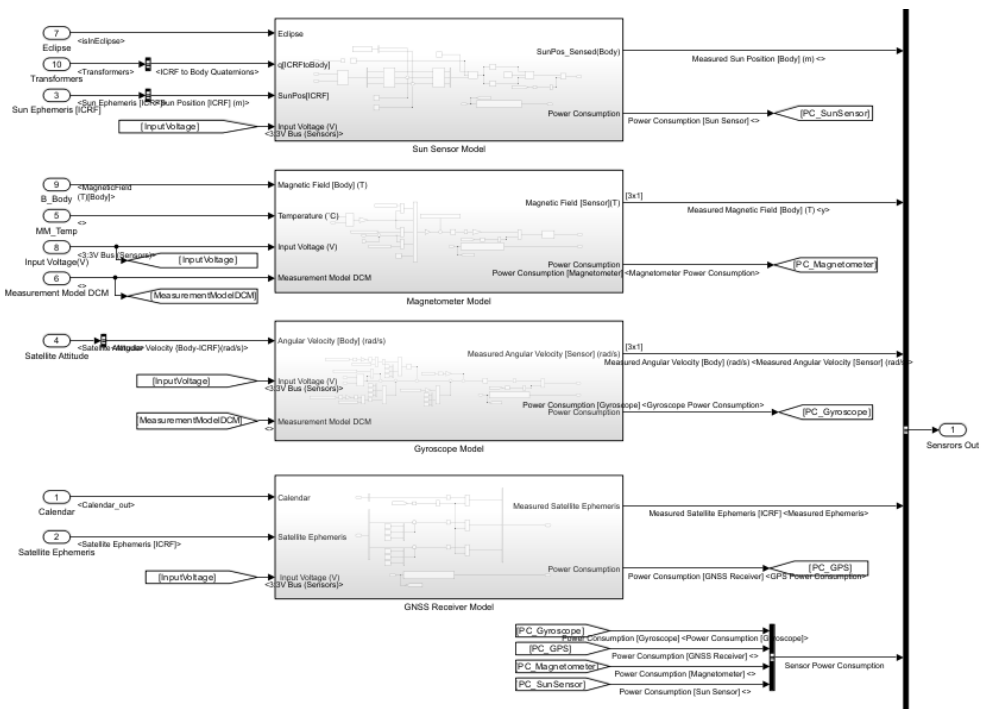

Figure 26- Sensors Model

All sensor inputs come from the environmental model, except for the **input voltage** of each sensor, which is provided by the **EPS physical model**. Therefore, the outputs from the sensors include the measured sun vector in the body frame, the measured magnetic field in the body frame, the measured angular velocity of the body frame relative to the inertial frame, and the measured position and velocity in the inertial frame. Additionally, the **power consumption and current** drawn by each sensor are sent as outputs to the **EPS software.**

The images below show the model of each sensor.

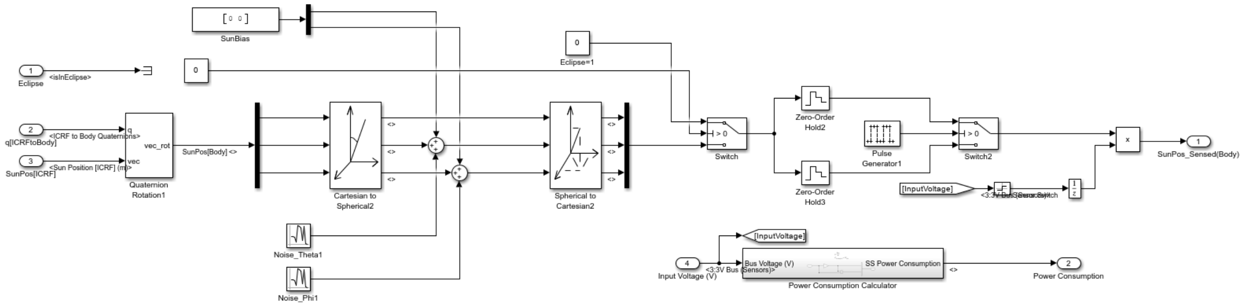

Figure 27- Sun Sensor Model

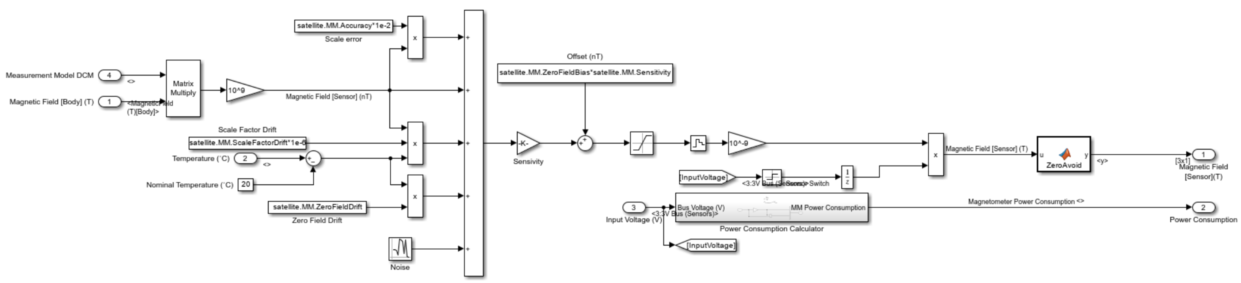

Figure 28- Magnetometer Model

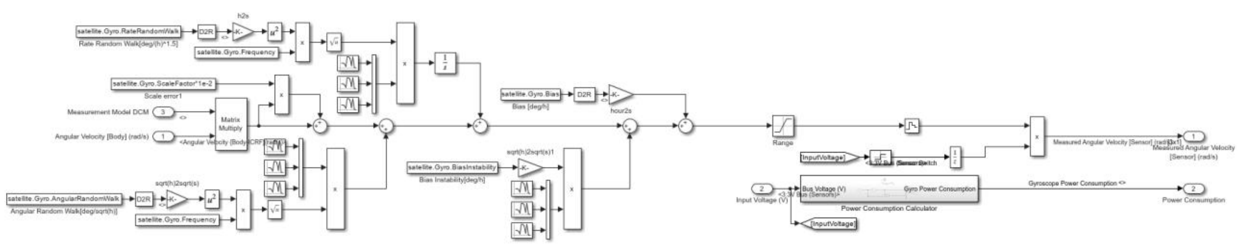

Figure 29- Gyroscope Model

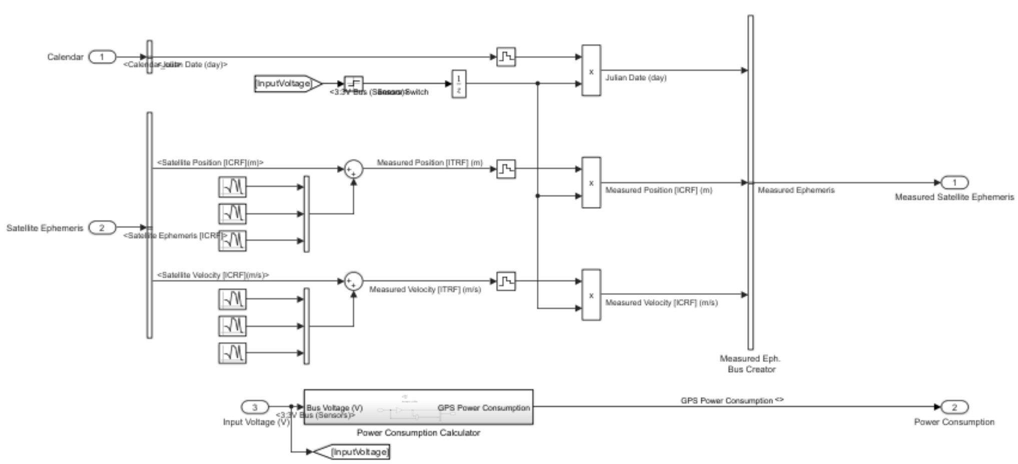

Figure 30- GNSS Receiver Model

The specifications of the sensors are provided in the code below.

```
%% Sensor Specifications

% GPS Specs
satellite.GPS.Frequency = 1;                    % Hz
satellite.GPS.Position.Bias = 5;                % m
satellite.GPS.Position.NoiseSTDDEV = 2.5;       % m
satellite.GPS.Velocity.Bias = 0.1;              % m/s
satellite.GPS.NominalVoltage = 3.3;             % V
satellite.GPS.NominalPower = 0.125;             % W

% Gyro Specs
satellite.Gyro.Frequency = 1;                   % 1Hz
satellite.Gyro.ScaleFactor = 0.5;               % precent (%)
satellite.Gyro.Range = [-150, 150] ;            % deg/s
satellite.Gyro.Bias = 7;                        % deg/hr
satellite.Gyro.AngularRandomWalk = 0.16;        % deg/sqrt(hr)
satellite.Gyro.RateRandomWalk = 6;              % deg/sqrt(hr^3)
satellite.Gyro.BiasInstability = 0.5;           % deg/s
satellite.Gyro.NominalVoltage = 3.3;            % V
satellite.Gyro.NominalPower = 0.14;              % W

% Magnetometer (MM) Specs
satellite.MM.Frequency = 1;                     % Hz
satellite.MM.Accuracy = 1;                      % precent (%)
satellite.MM.Sensitivity = 1;                   % mV/nT
satellite.MM.Range = [-1, 1].*2e5;              % nT
satellite.MM.ScaleFactorDrift = 100;            % ppm/˚C
satellite.MM.ZeroFieldDrift = 5;                % nT/˚C
satellite.MM.ZeroFieldBias = 300;               % nT
satellite.MM.NoiseVariance = [200; 100; 150]; % nT^2
satellite.MM.NominalPower = 0.1;                % W
satellite.MM.NominalVoltage = 3.3;              % V

% Sun Sensor (SS) Specs
satellite.SS.NominalPower = 0.01;               % W
satellite.SS.NominalVoltage = 3.3;              % V
```

5.1.2. Actuators Model

The figure below shows the model of the SPARCS-B actuator, which is a Triaxial Torquer. The input to the actuator is a control command in the form of a magnetic dipole, which is provided by the ADCS software in the C&DH. The output of the actuator includes the torque generated in the body frame, the output dipole, and the power consumption of each torquer.

The maximum dipole of the torquer, which is a design parameter, is applied to the torquer model via the constant **maximum allowable magnetic dipole moment**. Additionally, the generated torque output is fed into the **Environment model**, and more specifically, into the **satellite dynamics model**.

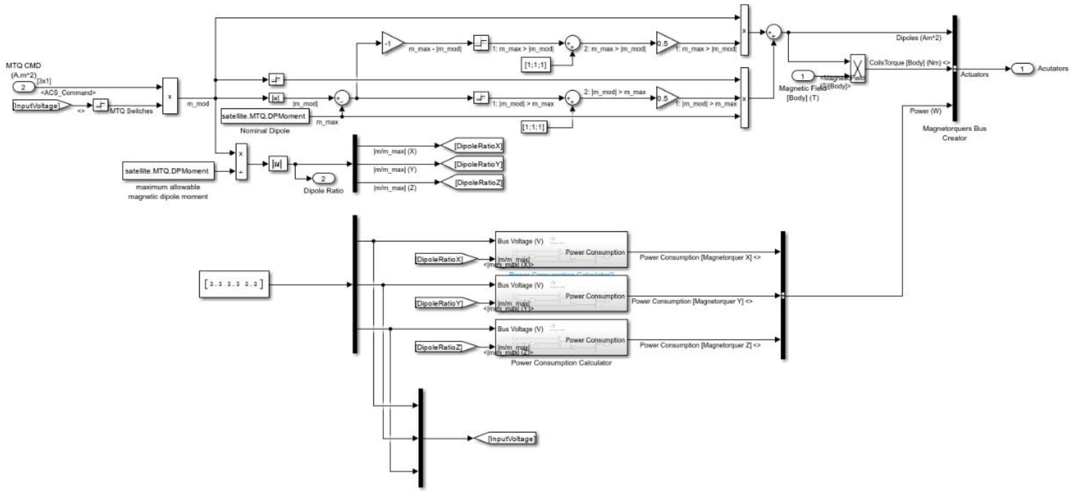

Figure 31- Magnetorquer Model

5.2. Software Model of ADCS

As shown in the figure below, the ADCS software, which is part of the C&DH software, consists of three main blocks:

1. The Onboard Environmental Models
2. The Attitude Determination Algorithms Block
3. The Attitude Control Algorithms Block.

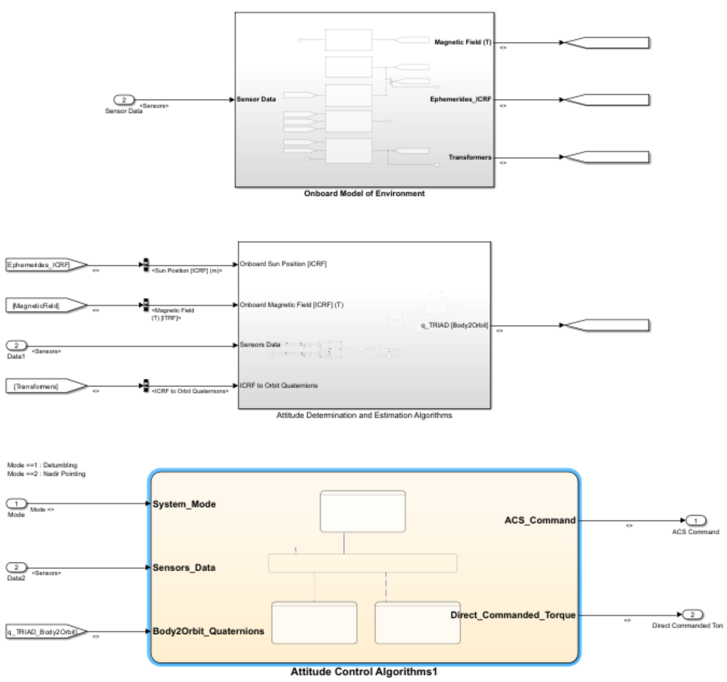

Figure 32- Software Model of ADCS

5.2.1. Onboard Environmental Model

The Onboard Environmental Model includes environmental models such as the calendar, magnetic field, orbit propagator, and the sun ephemeris model, all of which are implemented on the OBC software. These models are used for tasks such as attitude determination with static algorithms.

5.2.2. Attitude Determination and Estimation Algorithms

In SPARCS-B, we use a combination of TRIAD and EKF to determine the attitude for performing attitude control. Currently, only TRIAD is being used for attitude determination, and it will be combined with EKF in the next version.

The figure below shows the TRIAD model. This algorithm receives inputs from the sun sensor and magnetometer data in the Physical Domain, along with the magnetic field, sun vector, and transformers modeled in the Onboard Environmental Model. The output of this algorithm is the Body-to-Orbit quaternion, which is then passed into the attitude control algorithm.

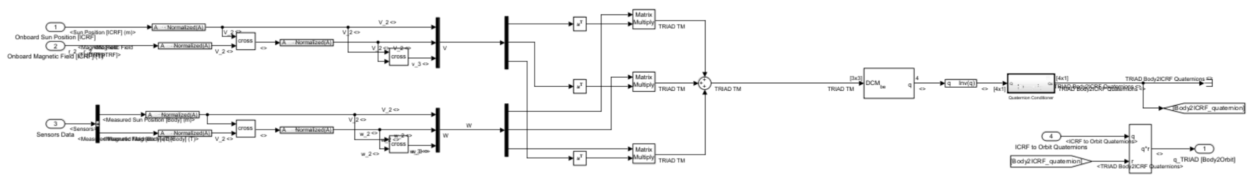

Figure 33- TRIAD Algorithm

5.2.3. Attitude Determination Algorithms

The figure below shows the block for the attitude control algorithms. According to the mission, SPARCS-B has two control modes: Detumbling and Nadir Pointing. Additionally, in the current model, an Off mode is also included.

The inputs to this block are the sensor data, the quaternion obtained from the attitude determination algorithm, and the system mode number. The output of this block is the control command in the form of a dipole. Furthermore, the torque command output from the Nadir Pointing algorithm is also an output; however, in the current model, the conversion of torque to dipole has not been performed due to changes in the magnetic field. We are working on implementing the conversion of torque to dipole for new MIL version.

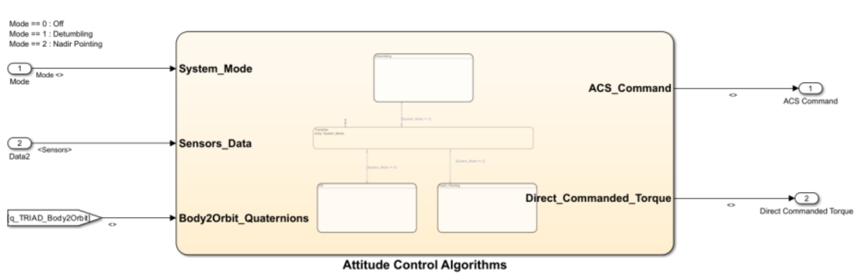

Figure 34- Attitude Control Algorithms

The figures below illustrate the models of the Detumbling and Nadir Pointing algorithms. For Detumbling, we have used the B-dot algorithm utilizing angular velocity measurements, and for Nadir Pointing, we have employed Quaternion Feedback with a PD controller.

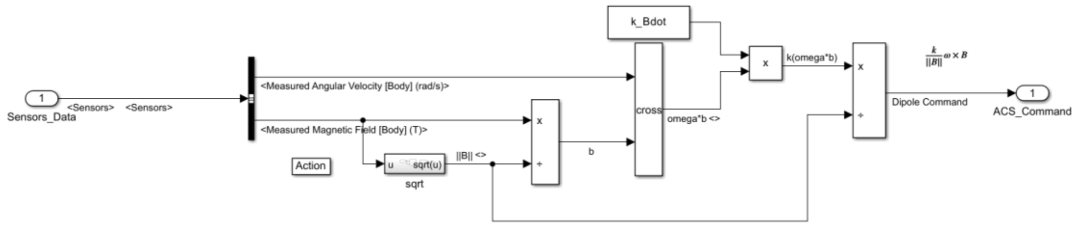

Figure 35- Detumbling Algorithm Model

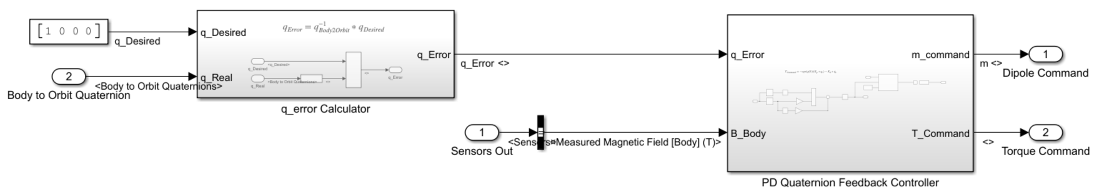

Figure 36- Nadir Pointing Algorithm Model

5.3. Results

The simulation results for the Detumbling and Nadir Pointing modes are presented below.

5.3.1. Detumbling

For the Detumbling mode, we assumed a Nominal Dipole of 0.25 and an initial angular velocity norm of 60 degrees per second.

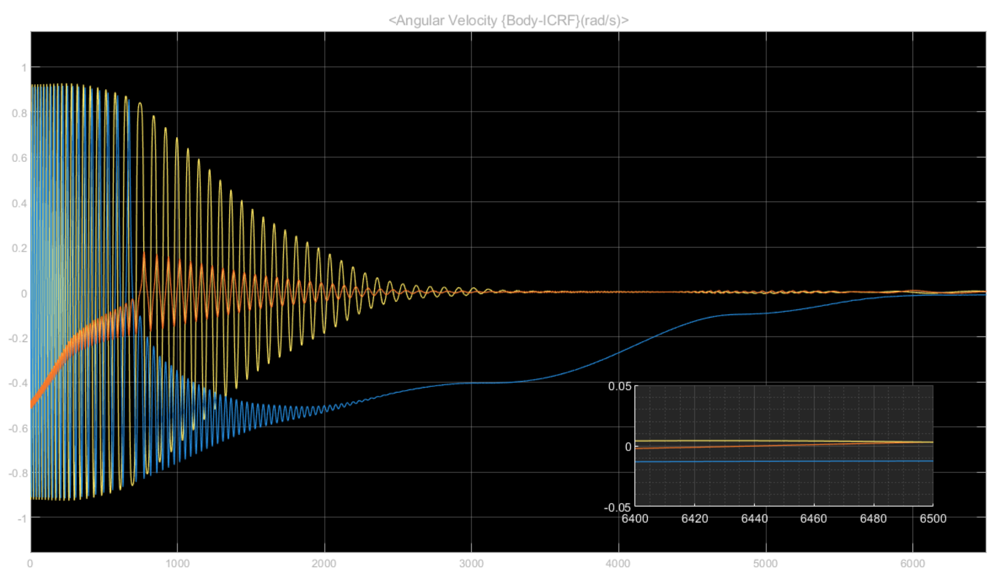

Figure 37- Angular Velocity Vector (rad/s)

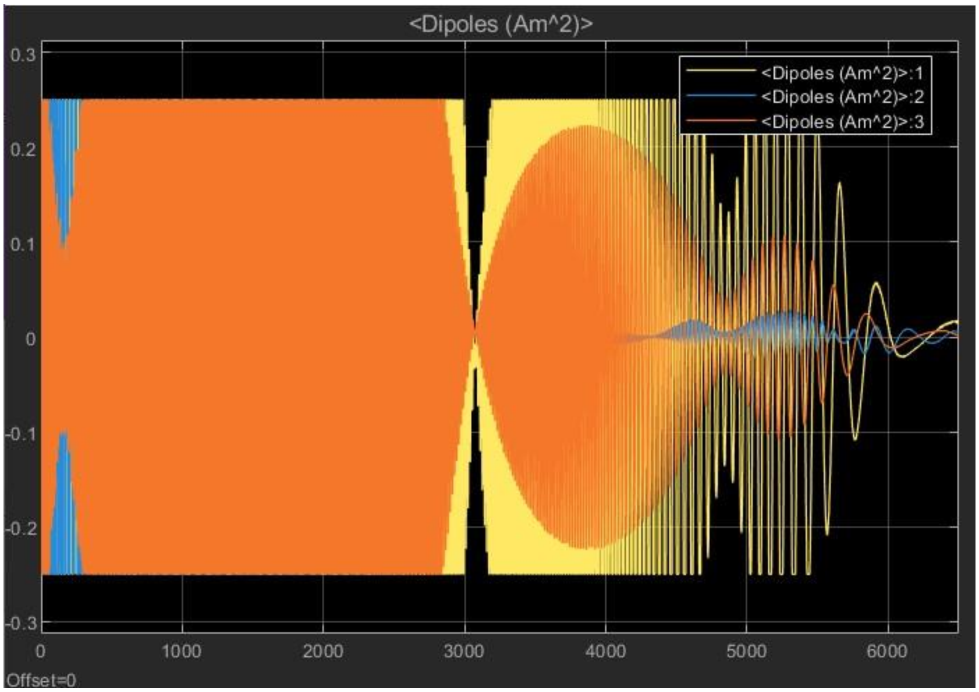

Figure 38- Command Dipole

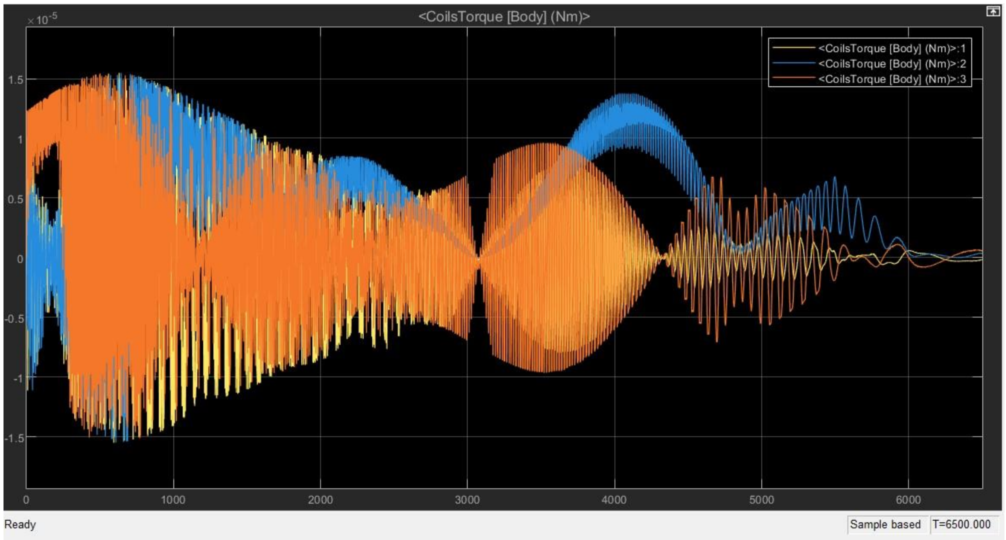

Figure 39- Command Torque

5.3.2. Nadir Pointing

Since the torque-to-dipole conversion in a variable magnetic field has not yet been implemented for **Nadir Pointing**, we limited the torque command to a range that ensures the torque produced under the maximum magnetic field corresponds to a maximum dipole of **0.25**. The norm of the initial Euler angle vector is assumed to be **80 degrees**.

Figure 40- Body to Orbit Quaternions

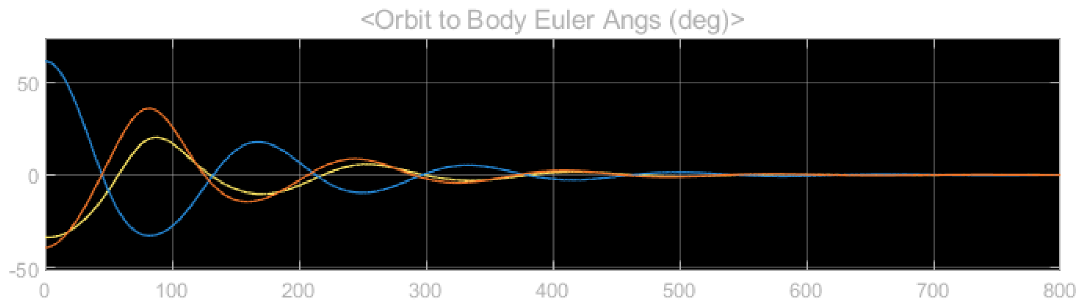

Figure 41- Euler Angles during Nadir pointing

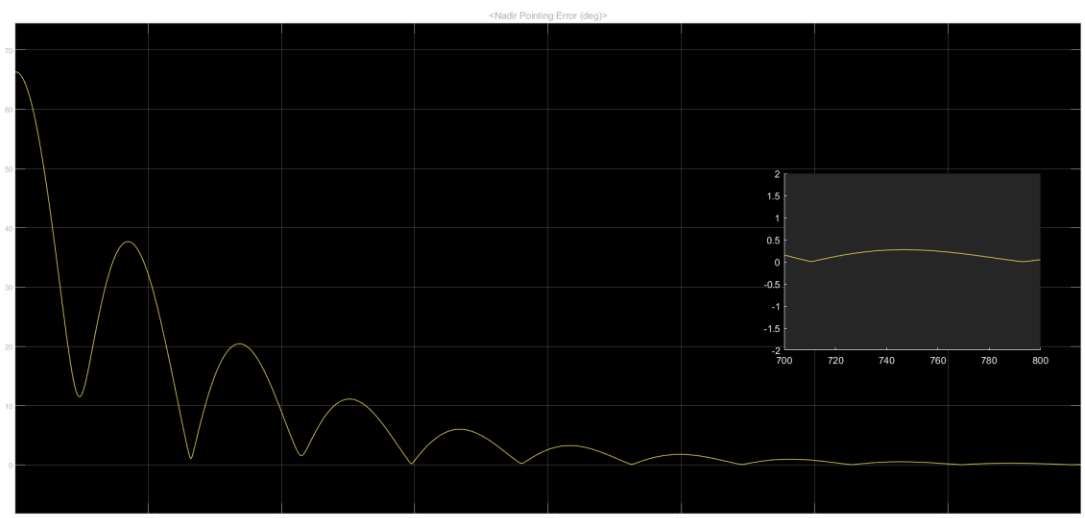

Figure 42- Nadir Pointing Error

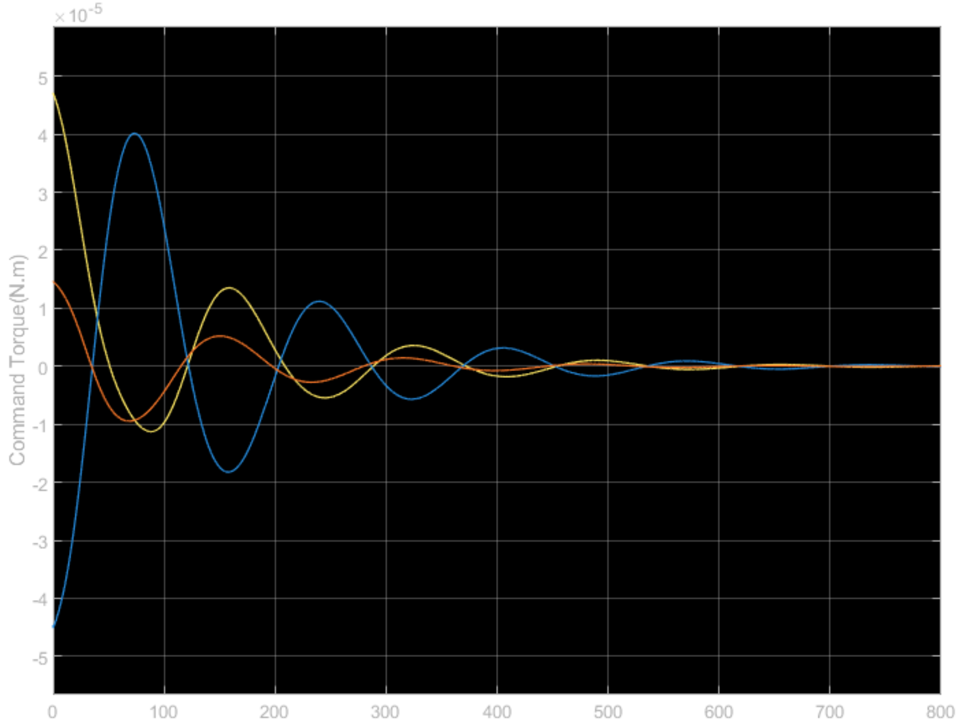

Figure 43- Command Torque

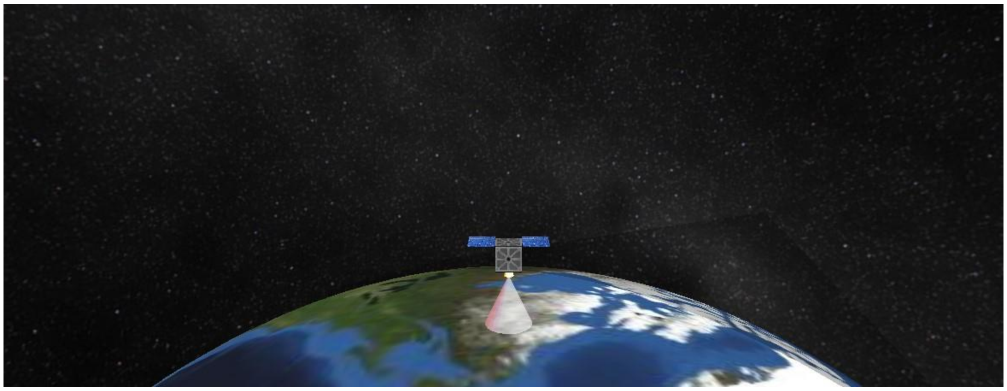

Figure 44- Satellite attitude after Nadir Pointing (animation frame)

6. Communication Subsystem

The communication subsystem has been simulated with respect to digital communication standards. The main resource and inspiration came from Matlab's "Satellite RF Link" example which is a very sophisticated and useful model. The model has been altered to answer the needs of SPARCS mission.

6.1. Model Philosophy

The model, being the communication subsystem of each satellite, sees the receivers, the channel and the other transmitter which can be the TT&C Communication or the other satellite (through ISL) in the following two blocks.

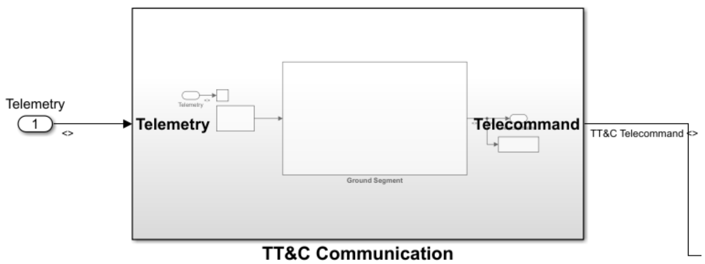

Figure 45- TT&C Communication Model

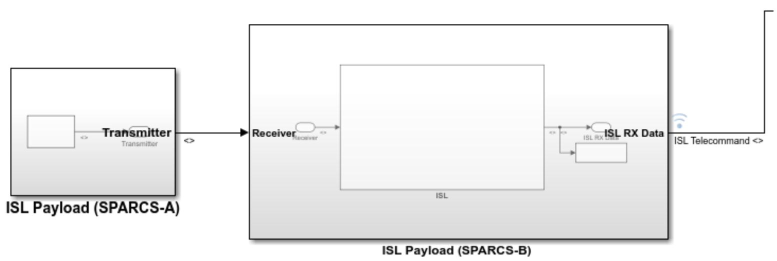

Figure 46- ISL Payload Model

The two data links of each model will be connected to C&DH. Inside of each RF link is similar, except for the gain and losses that are initialized in the "commrfsatlink_mask.m" file as followed. Be aware that the first number in each receiver antenna gain, is the gain calculated using HFSS or information given by Iranian Space Agency. The following numbers are the losses except free-space loss which is calculated elsewhere using a block.

```
paramRFSatLink.TXAntGain_Downlink = 10^((1.7) / 10);
paramRFSatLink.RXAntGain_Downlink = 10^((11 -(10+4+1+4)) / 10);

paramRFSatLink.TXAntGain_Uplink = 10^((11) / 10);
paramRFSatLink.RXAntGain_Uplink = 10^((1.7 -(10+4+1+4)) / 10);

paramRFSatLink.TXAntGain_ISL = 10^((1.7) / 10);
paramRFSatLink.RXAntGain_ISL = 10^((1.7 -(10+4+1+4)) / 10);
```

Figure 47- Antenna Gains & Miscellaneous Losses

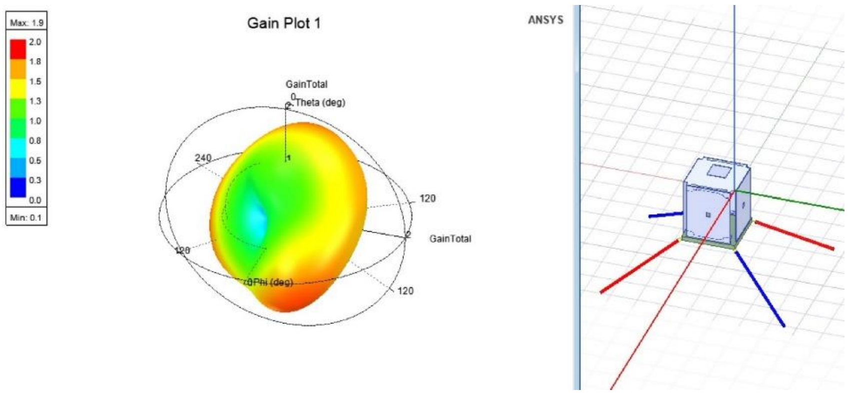

Figure 48- Antenna Gain Plot

So knowing that the inside of the blocks is the same, the TT&C Communication block will be explained below.

6.2. TT&C Communication Model

The model has 3 blocks; One accounting for downlink transmitter, One accounting for downlink receiver and mission control center and uplink transmitter, and one accounting for uplink receiver. The channel is simulated using an FSPL (free space power loss) block in each transmitter and miscellaneous losses in each receiver.

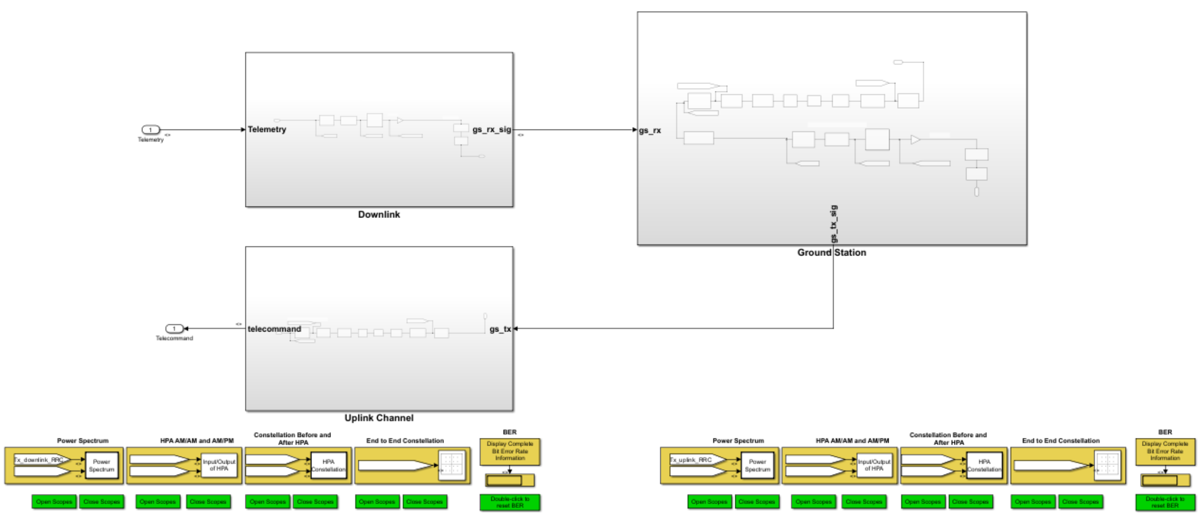

Figure 49- TT&C Communication Model

6.2.1. Satellite Downlink Transmitter (telemetry receiver)

This block takes the binary telemetry data from OBC and transmits to GS. You can the inside of this block as followed.

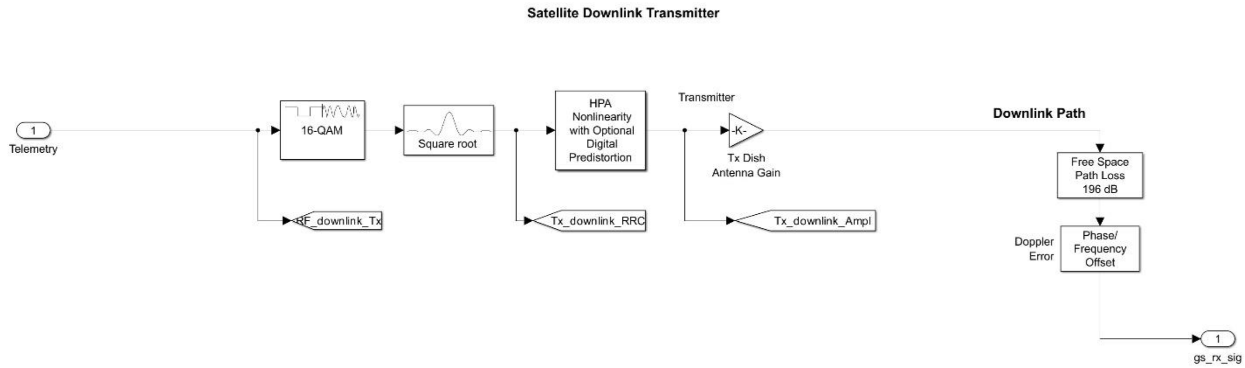

Figure 50- Satellite Downlink Transmitter

The data first gets modulated using a 16-QAM modulator block and then, after stabilization goes into the antenna to be transmitted. The modulator uses 2 watts of power at max. The free space loss has also been calculated using the altitude and frequency.

6.2.2. Ground Station Transceiver System & Mission Control Center (MCC)

The signal is received in the GS receiver, given to a mission control system which generates a bitstream of telecommand (which for now is a Bernoulli binary) and is transmitted.

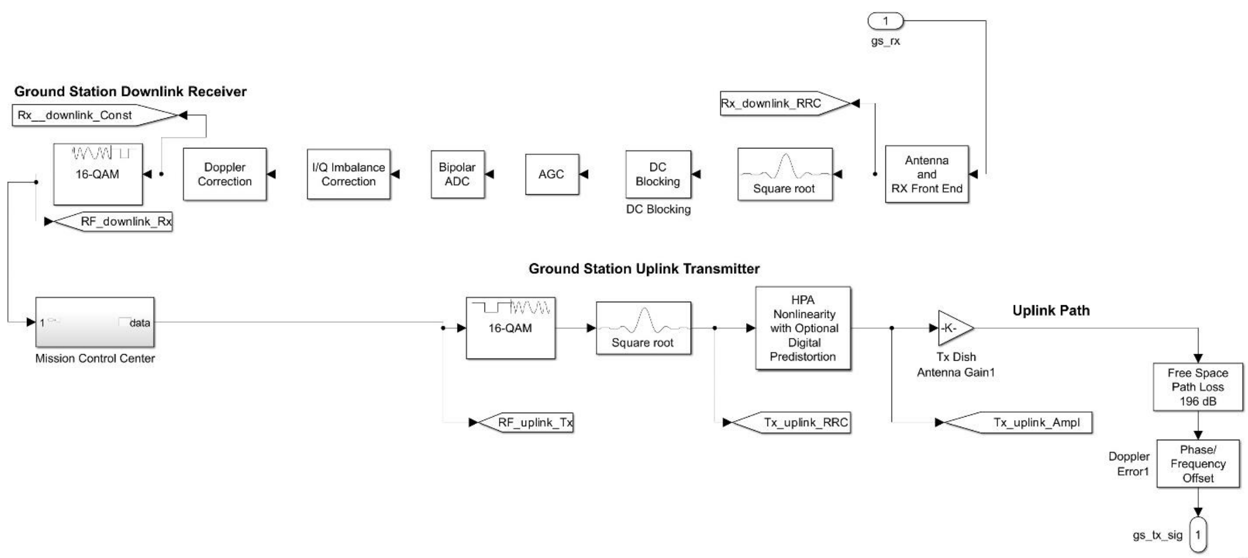

Figure 51- GS Transceiver & MCC

6.2.3. Satellite Uplink Receiver (telecommand transponder)

The uplink signal is received, demodulated and given to OBC as telecommand.

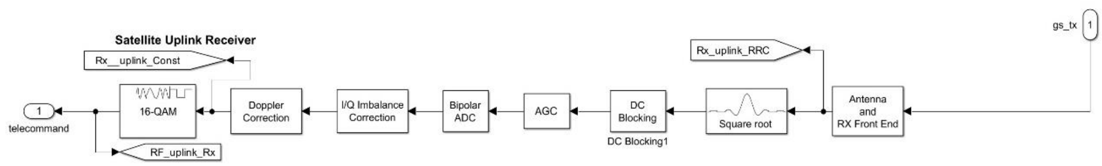

Figure 52- Satellite Uplink Receiver (telecommand transponder)

6.3. ISL Link

As stated before, the inside of this block is similar to the TT&C link.

6.4. Simulation Results

We have 4 communication links:

1. Downlink TX
2. Downlink RX
3. ISL TX
4. ISL RX

Each link is characterized with a constellation diagram which compares the transmitted binary data and the received one. The constellation diagrams of the 4 links are shown below and ensure a very reliable communication link.

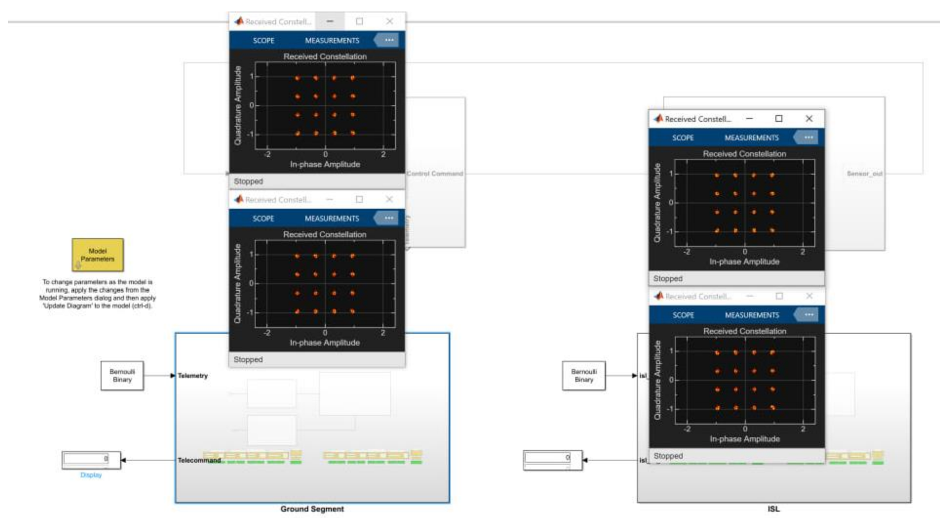

Figure 53- Constellation Diagrams