Inertia Modeling -> (Inertia Matrix subsystem block)

- Inertial Model from the spacecraft and main body going into solar panels, etc.
- Mass comes at the end considering all the sensors included
- 3 axis are fixed. a is the shortest side (x-axis), b is the length and h is the thickness.
- All dimensions are considered for the solar panels even if for the SRP we do not consider the y-direction.
- Times 2 at the end because we have 2 solar panels
- Calculate distance between the center of the main body and the center of the solar panels, specifically for the transport theorem.
- I_SC = I_SP + I_MB
- We still need to do the CAT model for plots (MISSING)

Kinematics -> (Kinematics subsystem block)

- w_BN as input enters and we create matrix W (from slides -> NEED REFERENCE)
- we create the quaternion by multiplying this matrix W by the quaternion
- Check the initial conditions for the quaternion creation on the kinematics subsystem block
- From the quaternion we transform into the matrix A_BN
    - Change the type of concatenation from matrix to column for simplification
- Relative attitude includes the orbit propagation but we might need to move it outside for better controllability
- A_BL comes from A_BN and A_LN (BL DCM between body frame and the LVLH frame)

Environment -> (Environment subsystem block)

- Gravity gradient formulas modeled and applied, not much to add
    - Outputs the torque
- Solar Radiation Perturbation
- Magnetic torque in body frame
    - Lab 7 slides
    - Check the R on magnetic torque against slides or lecture notes, we have a little confusion on the meaning

        - We changed the magnetic torque into the body frame (also from slides, the formula relating bn)
            - We might consider using more than 3 terms on the h_i
        - Definition of the m_hat vector = direction of the magnetic field (from slides)
        - Inclusion of the parasitic torque in order to compute the total magnetic torque
    - Solar Radiation Perturbation
        - Calculate position of the sun from formula in slides (inertial frame)
            - We could do it in MATLAB to simplify since it's all constants
        - Multiplied by A_BN matrix to get the solar position in the body frame
        - We normalize it and then use it on the force subsystem block
            - S_B is defined as global (which may slow down the performance)
        - Force subsystem block
            - Formula relating F_i which is a vector
            - only 2 faces of the solar panels are considered since the smaller side has a negligible effect
            - We compute a dot product between S_B and N_MB
                - N_MB is the directions normal to the solar panels
                - The solar panel old version still needs to be fixed/updated
                    - Remove selector box
        - Sum Forces
            - Fix the determinant highlighted area block
            - We should obtain the distance vectors between the center of mass (geometrically and gravity) and the point of application of the force
            - Compute the cross product in order to obtain the torque from the radii cross forces

Sensor

- Gyro
    - define sig_b, sig_n, t_sample
    - Check initial conditions and make sure the integrator can be used as thought out
    - Check which of the variances needs to be squared for units

- Magnetometer
    - Input A_BN, the matrix from Euler equations
    - The sensor considers
        - Non-orthogonality
        - Scale factor
        - Noise errors

- All angles on the slides for the error are epsilon on the Simulink block
- Check gain of filter on the Magnetometer and how it works
- Modified A_BN* can be used to find the magnetic field on the body frame b_B

- Sun Sensor (field of view +- 64 degrees)
  - Sun sensor accuracy may or not need not be checked

Actuator

- Reaction Wheels
  - Saturation conditions and maximum allowable torque checks

Attitude Determination

- QUEST Method which is a solution to the Wabha's problem
- Connect the outputs of the sensors to the values we get from the environment
- We need the DCM matrix A_BN, which is the goal of the method
- We only have 2 directions in space so the algebraic method was not an option for us, the solution would not be accurate enough. It doesn't weight the different sensors in different ways, they have different errors and impact.
- In this method we take the accuracies of both sensors (magnetometer and sun sensor) and we normalize them. Then the error can be computed by subtracting the real values from the sensor values.
- Alpha refers to the weight of the contribution of each error
- We can write the A_BN matrix as quaternions
- Create more subsystems on the Attitude determination

Kalman Filter for Detumbling

- Properly define Matrices C, Q and R
- Re-explain everything later

Sun Pointing

- Create the quaternions from each DCM matrices used as input using the Simulink block that considers ALL cases
- The desired attitude matrix was built within this subsystem
- Solar panel is aligned with the sun in the y-axis so S_N goes into the second row of the Desired A_B_N
- We consider the difference between DCM from Euler and the desired one. Also we consider the difference between the desired and the DCM coming from attitude determination

Miscellaneous

- DCM to Quaternion Simulink block needs to be updated

*** General check of products in general (cross vs element wise vs matrix, etc) ***
*** Replace unused values with terminators ***