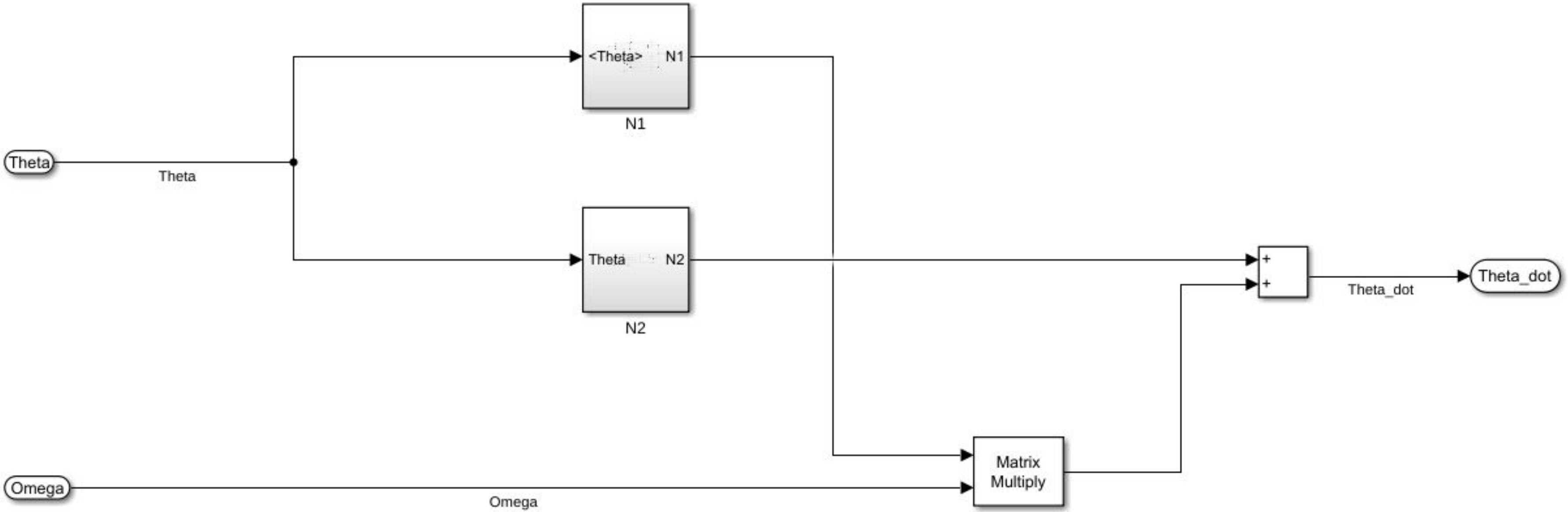
Reaction wheel based Attitude control system

Attitude Dynamics

For spacecraft in a circular orbit with an orbital rate $n = \sqrt{\mu/R^3}$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{\cos\theta_2} \begin{bmatrix} \cos\theta_2 & \sin\theta_1 \sin\theta_2 & \cos\theta_1 \sin\theta_2 \\ 0 & \cos\theta_1 \cos\theta_2 & -\sin\theta_1 \cos\theta_2 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \frac{n}{\cos\theta_2} \begin{bmatrix} \sin\theta_3 \\ \cos\theta_2 \cos\theta_3 \\ \sin\theta_2 \sin\theta_3 \end{bmatrix}$$

$$\implies \boxed{\dot{\bar{\theta}} = N_{1E}(\bar{\theta})\omega + nN_{2E}(\bar{\theta})}$$



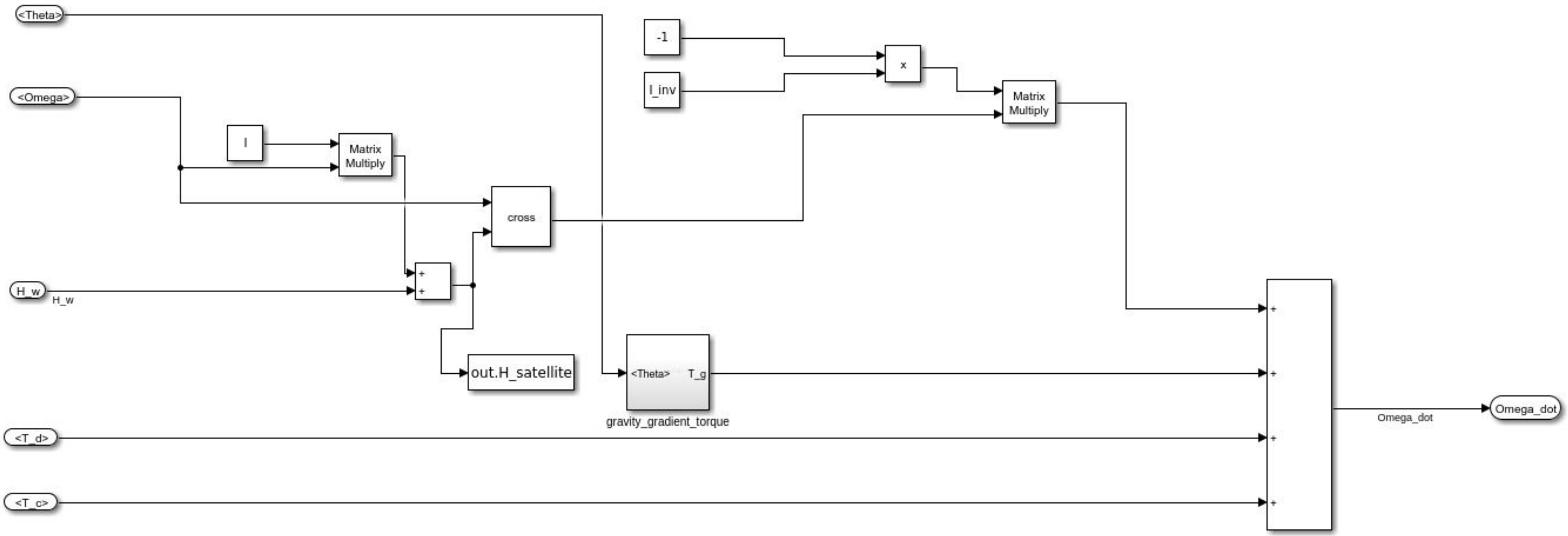
Theta
Theta
<Theta>
N1
N1
Theta
N2
N2
Matrix Multiply
Omega
Omega
+
+
Theta_dot
Theta_dot

$$J\begin{bmatrix}\dot{\omega}_1\\ \dot{\omega}_2\\ \dot{\omega}_3\end{bmatrix} = -\begin{bmatrix}0 & -\omega_1 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ \omega_2 & \omega_1 & 0\end{bmatrix} J \begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} + 3n^2 \begin{bmatrix}0 & -\cos\theta_1\cos\theta_2 & \sin\theta_1\cos\theta_2\\ \cos\theta_1\cos\theta_2 & 0 & \sin\theta_2\\ -\sin\theta_1\cos\theta_2 & -\sin\theta_2 & 0\end{bmatrix} J \begin{bmatrix}-\sin\theta_2\\ \sin\theta_1\cos\theta_2\\ \cos\theta_1\cos\theta_2\end{bmatrix} + \overline{T_d} + \overline{T_c}$$

$$\implies \boxed{J\dot{\overline{\omega}} = -S(\overline{\omega})J\overline{\omega} + \overline{T_g} + \overline{T_d} + \overline{T_c}}$$

To also include the contribution of angular momentum of reaction wheels

$$\boxed{J\dot{\overline{\omega}} = -\overline{\omega} \times (J\overline{\omega} + H_{rw}) + \overline{T_g} + \overline{T_d} + \overline{T_c}}$$



<Theta>
<Omega>
I
Matrix Multiply
cross
H_w
H_w
out.H_satellite
-1
I_inv
x
Matrix Multiply
<Theta>
T_g
gravity_gradient_torque
<T_d>
<T_c>
Omega_dot
Omega_dot

Reaction wheel

4 wheels in Pyramidal configuration.
β - tilt angle of reaction wheel axes

$$\begin{bmatrix} \hat{T}_{cx} \\ \hat{T}_{cy} \\ \hat{T}_{cz} \end{bmatrix} = \begin{bmatrix} T_{cx} / \cos(\beta) \\ T_{cy} / \cos(\beta) \\ T_{cz} / \sin(\beta) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

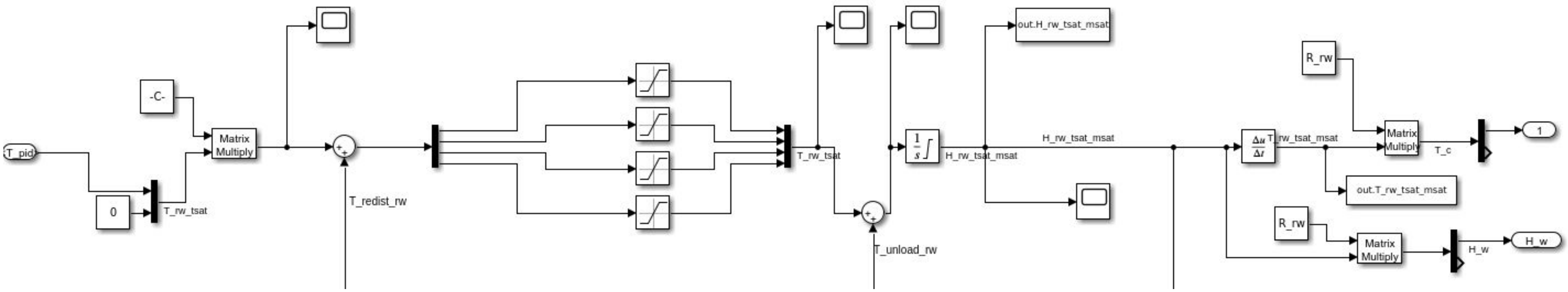
For optimal distribution of momentum among the 4 wheels.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1/2 & 1/2 \\ 0 & 1 & 1/2 & -1/2 \\ -1 & 0 & 1/2 & 1/2 \\ 0 & -1 & 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} \hat{T}_{cx} \\ \hat{T}_{cy} \\ \hat{T}_{cz} \\ 0 \end{bmatrix}$$

Pyramidal reaction wheel arrangement optimization of satellite attitude control subsystem for minimizing power consumption

Abolfazl Shirazi* and Mehran Mirshams**

Space Research Laboratory, Aerospace Engineering Department, K.N. Toosi University of Technology, Tehran, Iran

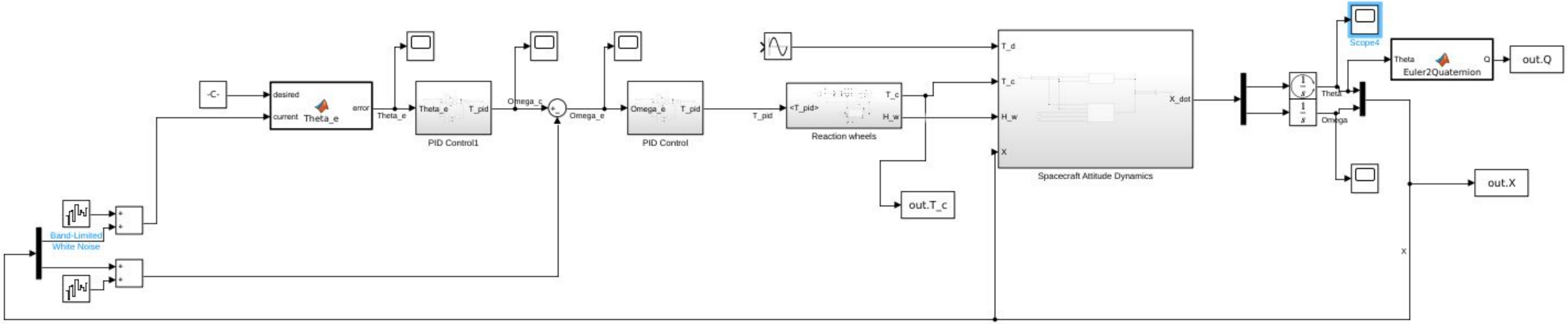


T_pid
-C-
0
T_rw_tsat
Matrix Multiply
T_redist_rw
T_rw_tsat
T_unload_rw
H_rw_tsat_msat
out.H_rw_tsat_msat
H_rw_tsat_msat
T_rw_tsat_msat
out.T_rw_tsat_msat
R_rw
Matrix Multiply
T_c
1
R_rw
Matrix Multiply
H_w
H_w

Cascaded PD control

Outer loop is the primary controller. It converts the error in attitude to a desired angular velocity (Makes it go from initial rates to 0).

The inner loop then takes in the error in angular velocity and commands the reaction wheels. The inner loop must have >10 times faster dynamics



-C-
desired
current
Theta_e
error
Theta_e
Theta_e
T_pid
PID Control1
Omega_c
Omega_e
Omega_e
T_pid
PID Control
T_pid
<T_pid>
T_c
H_w
Reaction wheels
out.T_c
T_d
T_c
H_w
X
X_dot
Spacecraft Attitude Dynamics
Scope4
Theta
Omega
Theta
Q
Euler2Quaternion
out.Q
out.X
X
Band-Limited
White Noise

For the given initial and final conditions, and the restrictions on maximum reaction wheel torques and angular momentum, the reaction wheels saturate very quickly before the desired final conditions are reached.

Hence, a momentum management system was designed to redistribute the momentum among the 4 wheels and also provide an external unloading torque when necessary.

Available Data

Initial Slew Attitude (deg)	[0 0 0]
Final Slew Attitude (deg)	[30 50 70]
Initial Rate (deg/sec)	[15 -15 15]
Desired Rate (deg/sec)	[0.2 0.2 0.2]
MOI of the Reaction Wheel (kg m^2)	0.00042
MOI of the satellite (kg m^2)	[2.10 0.00 0.01; 0.00 2.30 -0.03; 0.01 -0.03 1.72]
Disturbance Torque (Nm) acting on the satellite (sinusoidal in ECI Frame)	[10^-6 10^-6 10^-6]
Maximum Torque generated (Nm) by the Reaction Wheel	0.015
Angular Momentum generated (Nms) by the Reaction wheel	0.035
Reaction Wheel Configuration	Pyramidal Configuration

Momentum Management System

- Redistribution
- Unloading

Redistribution

R_rw is the matrix that converts 4 individual reaction wheel torques into the body torque

$$\begin{bmatrix} T_b \\ 0 \end{bmatrix} = R_{rw} T_{rw}$$

Redistribution is triggered when any of the reaction wheels exceeds 90% of the max momentum capacity

Let's assume wheel 1 reached >0.9 H_c_max
Then we would like to set the desire momentum of wheel 1 to 0, and distribute hat momentum to the other wheels.

momentum to be redistributed to other wheels, del_H = [H_1; 0; 0; 0]

Momentum to be redistributed in 3 axis, H_redist = R_rw del_H

Now we want to know how to distribute the momentum. For that, we make first column of R_rw as 0 (let's call it R_rw_redist), so that no momentum will be added to the saturated wheel

H_redist_rw = inv(R_rw_redist) *H_redist

Hence, the new desired momentum for the wheels become
H_des_rw = [0;h1;h2;h3]+H_redist_rw
H_des = (R_rw H_des_rw)[1:3]

The redistribution is performed on the model by calculating a redistribution torque, and the subjecting it to the saturation limits of the reaction wheels.
T_redist = -K_redist*(H-H_des)
T_redist_rw = inv(R_rw)[T_redist;0]

Momentum unloading

It is triggered when sum of momentum of all reaction wheels are >0.9*4*H_c_max

Once triggered, it continues until the sum of momentums of all reaction wheels are <0.1*4*H_c_max.

Hence, the H_desired_rw = 0.1*H_c_max[1;1;1;1],

H_des = R_rw*H_desired_rw

T_unload = -K_unload*(H-H_des)

T_unload_rw = inv(R_rw) [T_unload; 0]

Assumption - unloading torque is provided by an external source. There is no restriction in the magnitude.

Simulink model results

1000 seconds

With sinusoidal disturbance torque of [10^-6 10^-6 10^-6] Nm but no sensor noise.

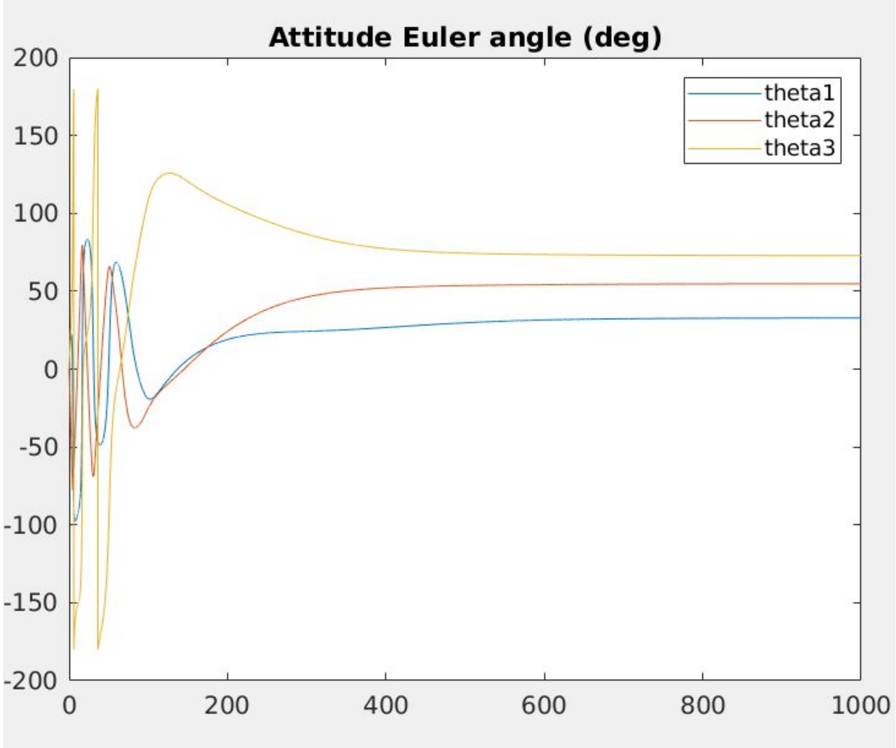
Momentum management system gains

K_redist = [1;1;5]

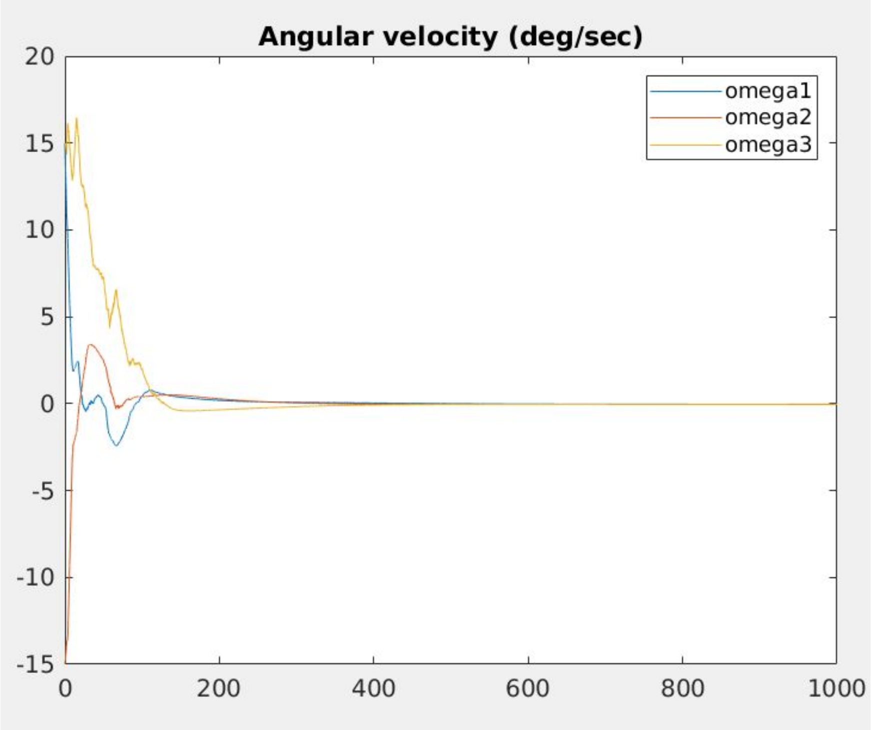
K_unload = 1.5*1e-1

PID gains

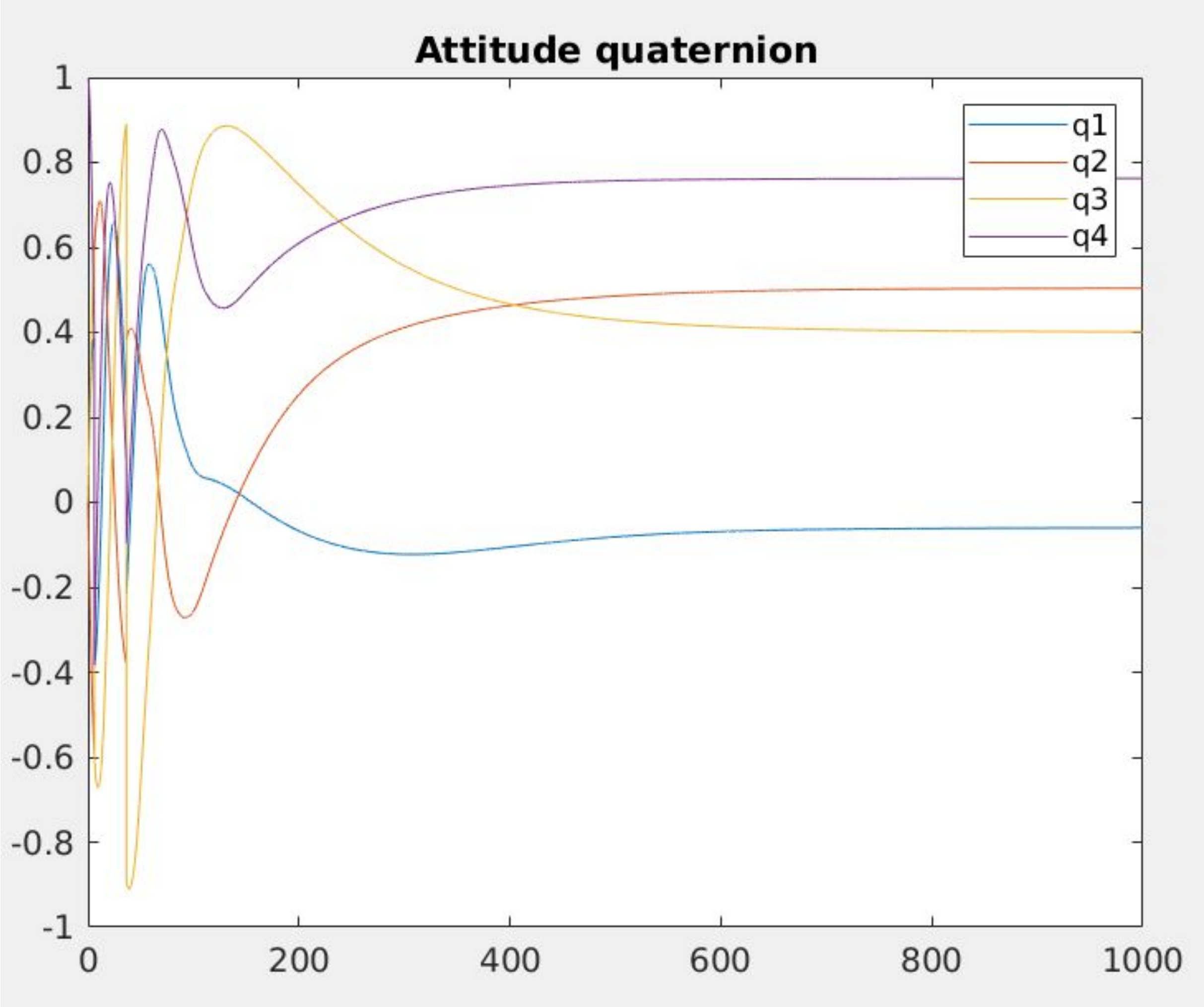
Outer loop	Inner loop
Kp_t=1e-2;	Kp_w=1.1*1e-1;
Kd_t= 1e-4;	Kd_w=1e-1;
Ki_t=0;	Ki_w=0



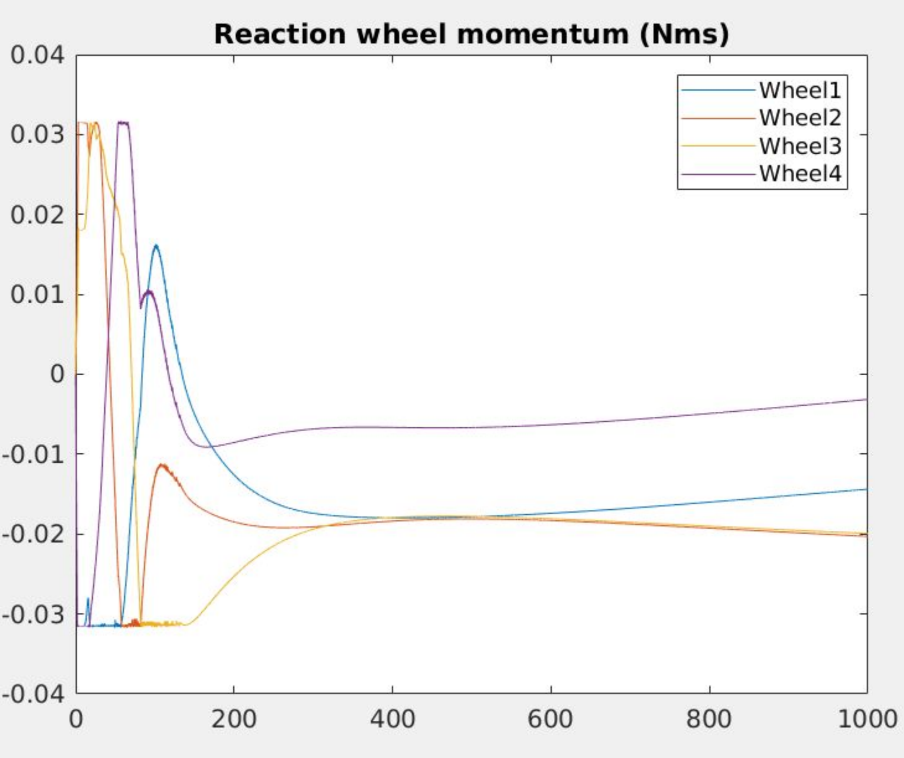
Attitude Euler angle (deg)
theta1
theta2
theta3
200
150
100
50
0
-50
-100
-150
-200
0
200
400
600
800
1000



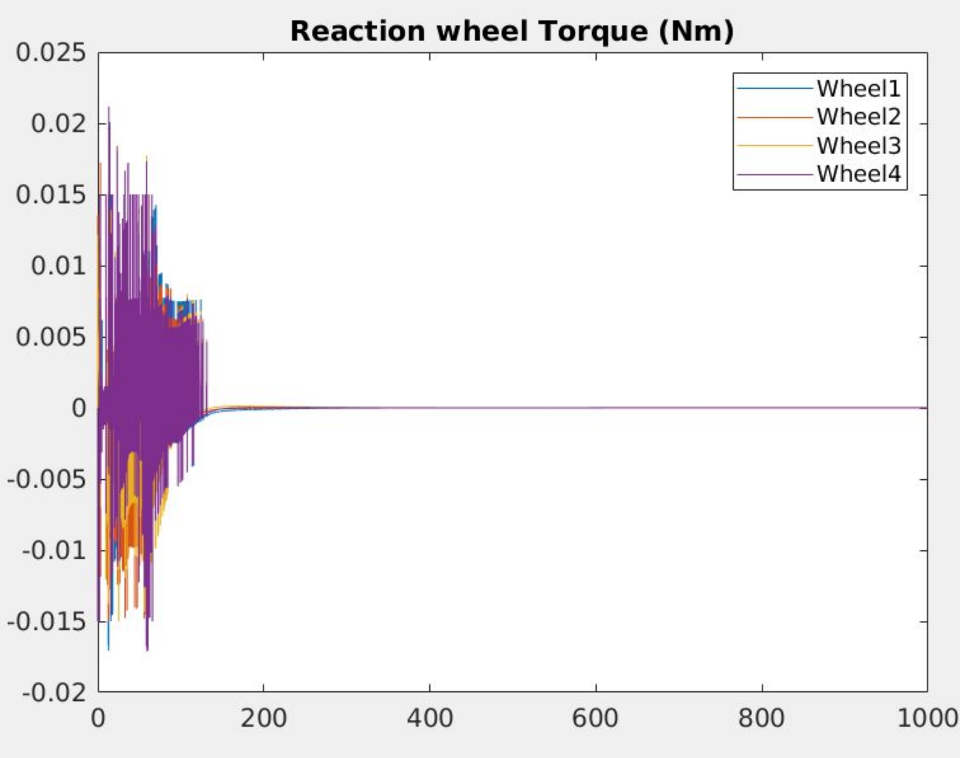
Angular velocity (deg/sec)
omega1
omega2
omega3
20
15
10
5
0
-5
-10
-15
0
200
400
600
800
1000



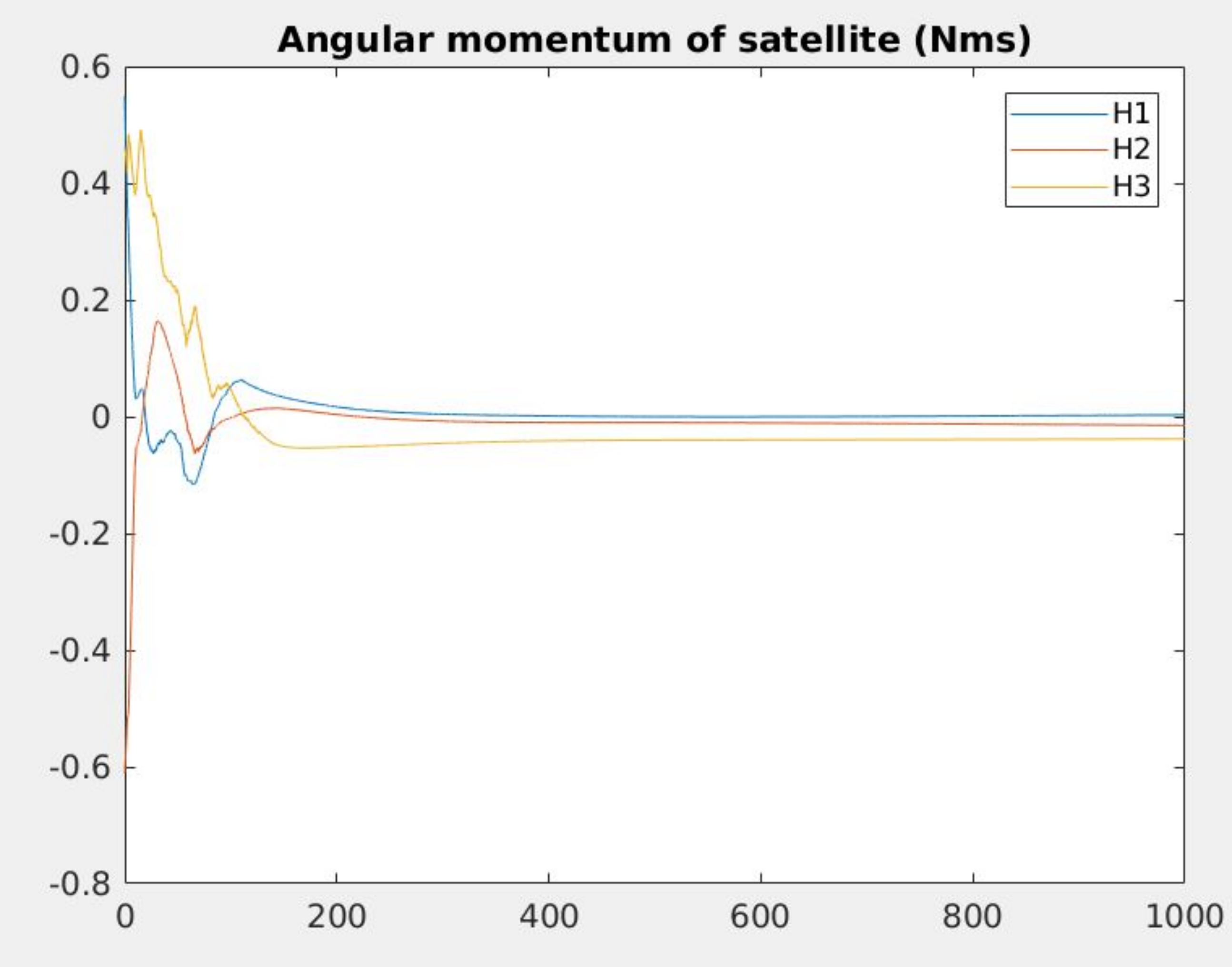
Attitude quaternion
q1
q2
q3
q4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
200
400
600
800
1000



Reaction wheel momentum (Nms)
Wheel1
Wheel2
Wheel3
Wheel4
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
0
200
400
600
800
1000

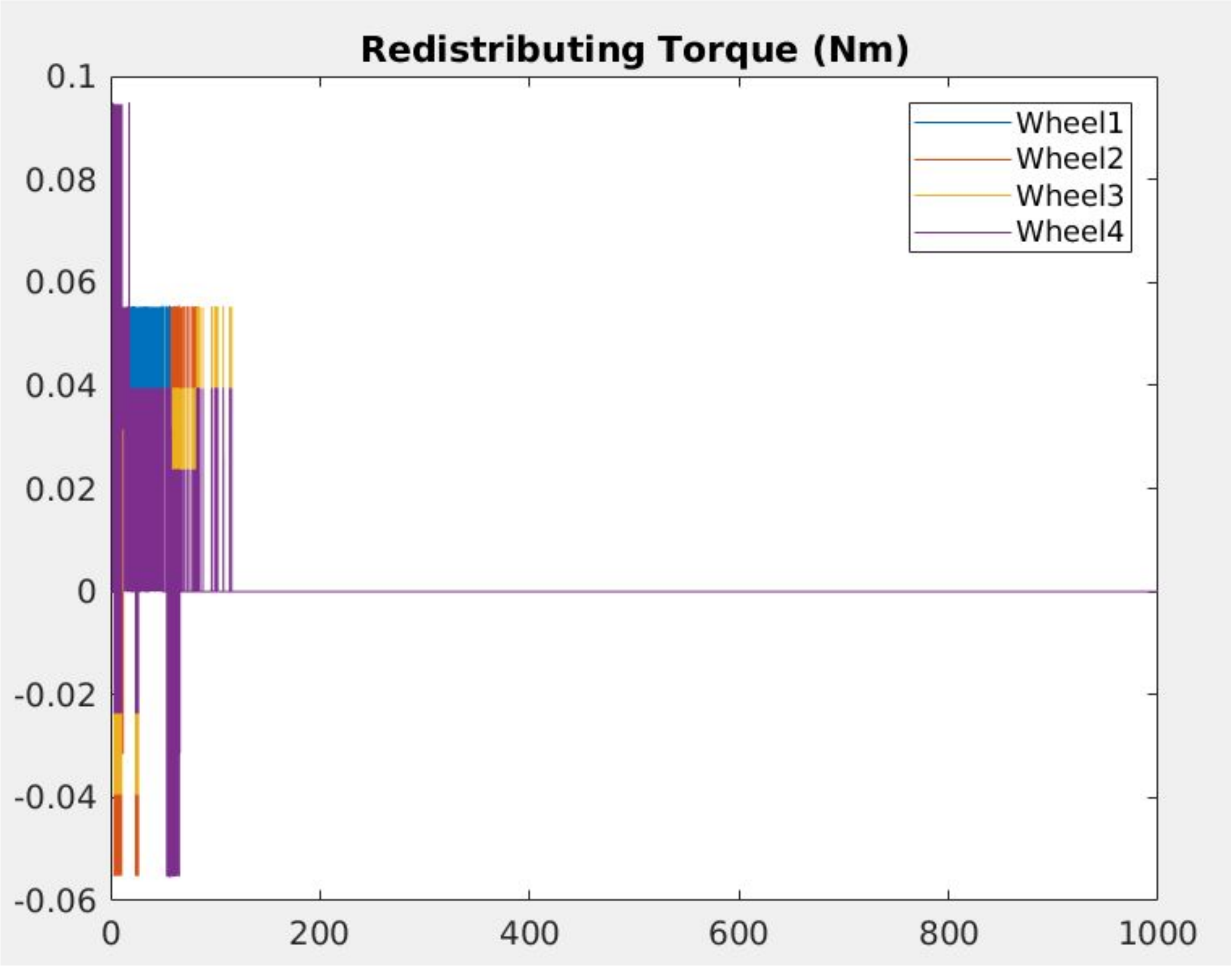


Reaction wheel Torque (Nm)
Wheel1
Wheel2
Wheel3
Wheel4
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
0
200
400
600
800
1000

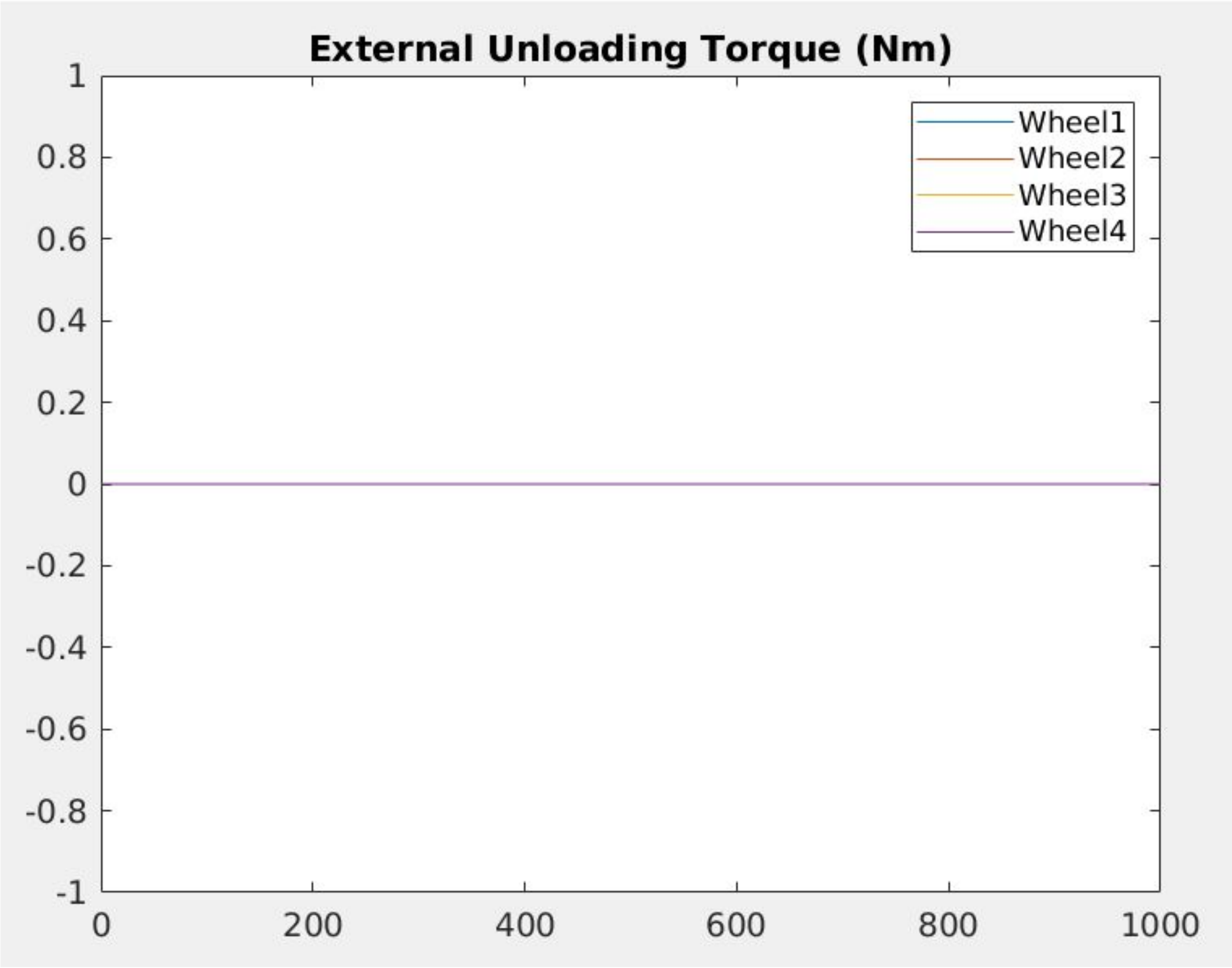


Angular momentum of satellite (Nms)
H1
H2
H3
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
200
400
600
800
1000

Even though redistribution torque command might seem high, this command passes through the torque saturation block of reaction wheels.



Even though the unloading functionality is implemented, it was not necessary for this problem and tuned control.



Settling time (2%)

Theta_settling_time = [667.2000; 500.5000; 504.6000] s

Omega_settling_time = [595.8000; 766.2000; 817.7000] s

At 900 s , the attitude and angular velocities are

Theta=[32.6559;   54.6541;   73.0042] deg

Omega=[-0.0317;   -0.0411;   -0.0321] deg/s

Ways of improving

- Better momentum management system. The redistribution of momentum does not seem like a smooth operation right now from what is seen on the plot of reaction wheels torques.
- Adding additional blocks that can linearize the dynamics. This will make the system more robust, but also prone to hardware modeling issues. Hence the decision should be made based on the expected error in hardware modeling.
- Add different modes for control operation for different errors in attitude and angular velocity. So different modes for when errors are high, compared to when the errors are low and need finer adjustments.

Thank You!