TESTING A REACTION WHEEL SPEED CONTROL LOOP

Cedric Cuypers(1), Tjorven Delabie(1), Dirk Vandepitte(2), Jelle Lanting(2)

(1) arcsec, Blijde Inkomststraat 22 Leuven Belgium, +32 498 817370, cedric@arcsecspace.com,
tjorven@arcsecspace.com
(2) KU Leuven, Celestijnenlaan 300 Leuven Belgium, +32 16 32 24 80,
dirk.vandepitte@kuleuven.be, jelle.lanting@kuleuven.be

ABSTRACT

Reaction wheels are common actuators in the attitude control of satellites. One performance indicator of an Attitude Determination and Control System (ADCS) is its ability to accurately control the spacecraft pointing and to keep the satellite stable. To precisely control the pointing of satellites, the angular velocity of reaction wheels must be precisely controlled. Instabilities in the bearings of reaction wheels can cause sudden changes in friction, changing the speed of the reaction wheels. This deteriorates the stability and performance of the ADCS. To overcome this issue, an internal control loop can be implemented in the reaction wheel system to estimate the current friction level and increase or decrease the control torque accordingly.

Different types of controllers were simulated. The PI controller and Kalman filter showed similar performance. The PI controller was chosen for its simplicity over the Kalman filter. The control loop was then implemented in a hardware set-up and used to control a motor. The motor was controlled as a reaction wheel would be controlled, i.e. by commanding a certain torque from the system, rather than a speed. The controller successfully restores the angular velocity within a few seconds after the friction is introduced.

1 INTRODUCTION

Arcsec is a KU Leuven spin-off company that develops ADCS modules for SmallSats sand CubeSats. In the framework of a GSTP project, arcsec is currently working on a compact, high precision reaction wheel with little microvibrations for small satellites. This project involves several research paths to increase the performance of the reaction wheel, that will be combined in one integrated unit. One of these paths focusses on the control side of the reaction wheels, and more specifically on their behavior during changes in friction conditions. Unpredictable changes in the friction have been observed in the reaction wheels of several past missions such as Cassini, XMM-Newton and Rosetta [1] [2] [3]. These sudden changes in friction will cause changes in the angular velocity of the wheels, which will in turn affect the pointing of the satellite. The work presented in this document focusses on a control loop that monitors the speed for such unpredictable friction anomalies and increases the motor demand to quickly restore the angular velocity in such an event. Different controller designs were first simulated in a Simulink environment and compared with a baseline simulation of a reaction wheel without an internal control loop. One of them was selected and implemented in a hardware set-up to validate its operation on a physical system.

2 DESIGN of the ESTIMATOR CONTROL LOOPS

2.1 No friction estimation

In a first approach, no friction estimation is added to the control loop as shown in Figure 1. The control torque is passed directly to the reaction wheels [4].

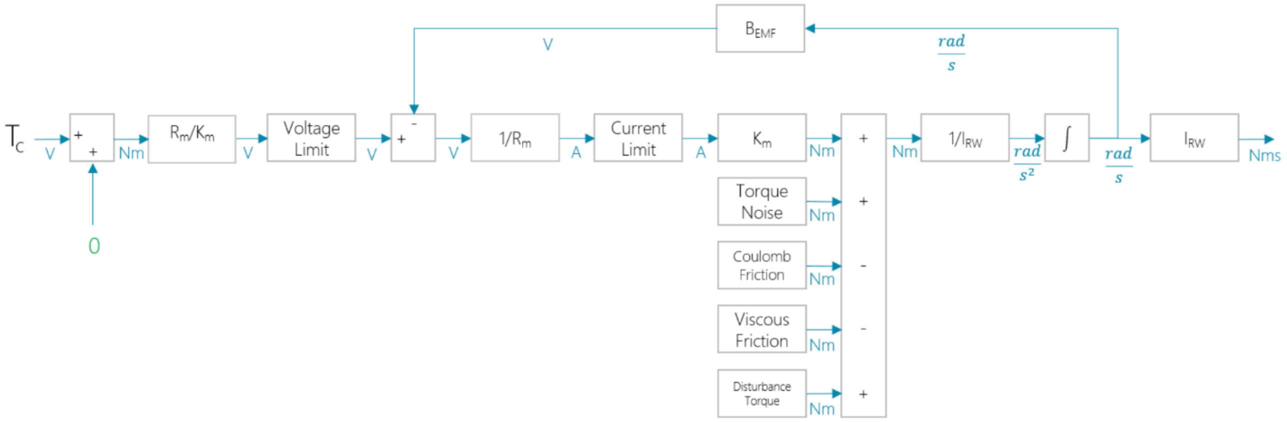


Figure 1: Control strategy without friction estimation

2.2 PI friction estimator

In a second approach, a PI controller was used to estimate the friction level as shown in Figure 2. The estimated friction is then added to the control torque before being passed to the reaction wheels.

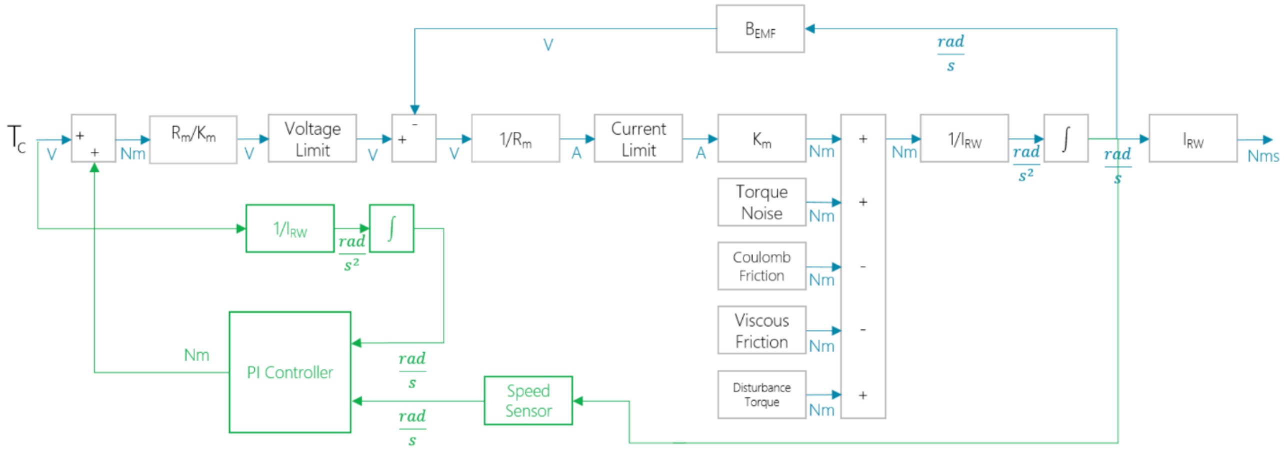


Figure 2: Control strategy PI friction estimation

The PI controller takes the desired speed ω_{des} and measured speed ω_{meas} to estimate the friction in the system. The desired speed is calculated from the control torque and reaction wheel inertia [4] and the following speed error e is obtained:

$$e = \omega_{des} - \omega_{meas}$$

Equation 1

The estimated friction is calculated as

$$T_{fest} = K_p e + K_i \int e$$

Equation 2

with the gains calculated as

$$K_p = 2\zeta\omega_{PI}I_{RW} \text{ and } K_i = \omega_{PI}^2 I_{RW}$$

Equation 3

2.3 Kalman Filter

In a third approach, a Kalman filter is used to estimate the friction. This friction is similarly added to the control torque before passing it to the reaction wheel.

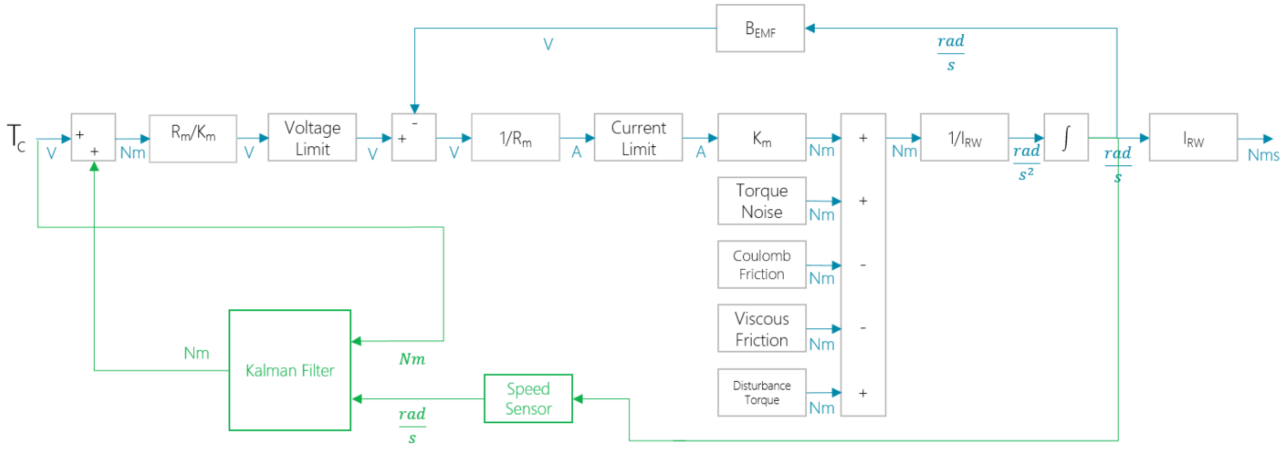


Figure 3: Control strategy KF friction estimation

The Kalman filter implementation estimates the friction torque $T_{f\ est}$ as well as the angular velocity ω_{RWest} of the reaction wheel [4] as one state vector x:

$$x = \begin{bmatrix} \omega_{RWest} \\ T_{f\ est} \end{bmatrix}$$

Equation 4

The control law implemented is based on a linear friction model

$$T_c - T_f = T_{RW}\dot{\omega}_{RW}$$

Equation 5

where T_f is replaced by the estimated friction $T_{f\ est}$.

3 SIMULATION and ESTIMATOR SELCTION

3.1 Simulation environment

An overview of the simulation environment for the comparison of the different friction estimator strategies is shown in Figure 4.

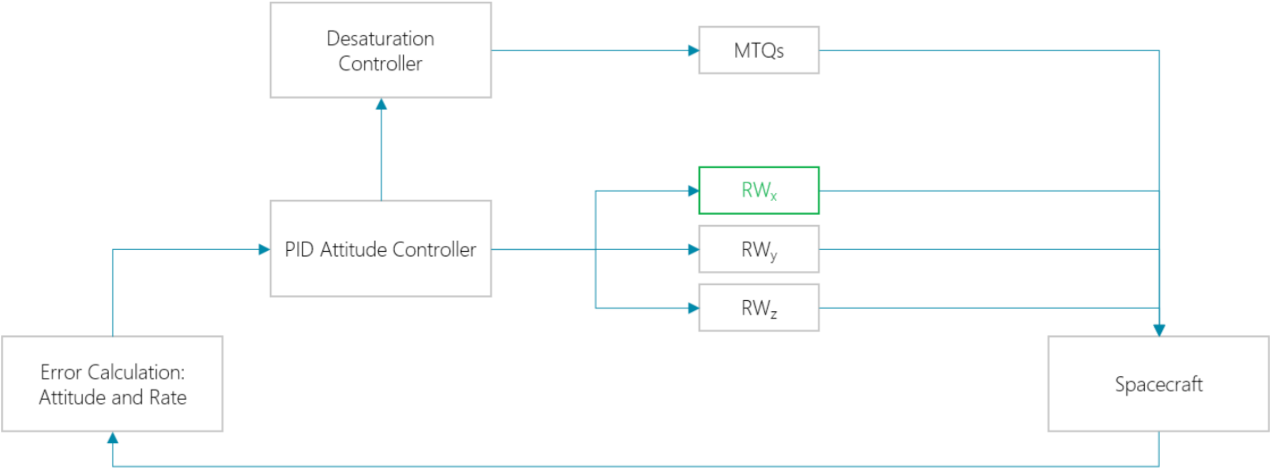


Figure 4: Simulation environment overview

The ‘Spacecraft’ block contains the attitude propagator of the spacecraft. The true quaternion and rotational rate are passed to the ‘Error Calculation’ block. This latter block calculates the difference between the desired and true quaternion and rotational rate and provides input to the ‘PID Attitude Controller’. This block calculates the desired torque for each reaction wheel and also sends a corresponding input to the ‘Desaturation Controller’. The ‘Desaturation Controller’ determines the required magnetic moment of the magnetorquers to keep the reaction wheels around a nominal rotational rate. The reaction wheel and magnetorquer output are inputs to the attitude propagator of the spacecraft, closing the loop. The reaction wheel speed control is performed inside the ‘RW-’ blocks. For the comparison the focus is on the RWx block [5].

3.2 Simulation results

In this section the three control strategies presented in chapter 2 are compared to each other. Two scenarios are investigated: steady-state behavior and performance under the presence of torque spikes.

3.2.1 Steady state

For the steady-state analysis, a slew maneuver was simulated and the angular velocity of the reaction wheels is logged. The spacecraft initial quaternion was [1 0 0 0] and a desired attitude of [120; 60; 30] ([roll; pitch; yaw]) was commanded. The resulting attitude of the spacecraft is plotted below for all three control strategies [6].

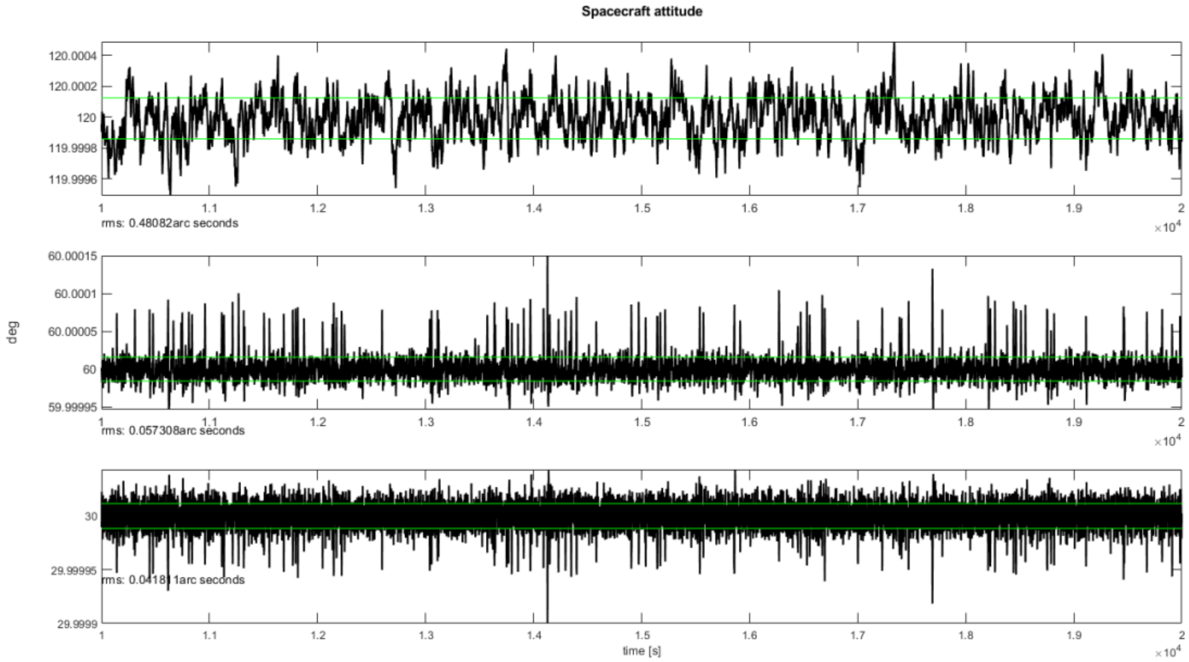


Figure 5: No RW speed control: Spacecraft attitude

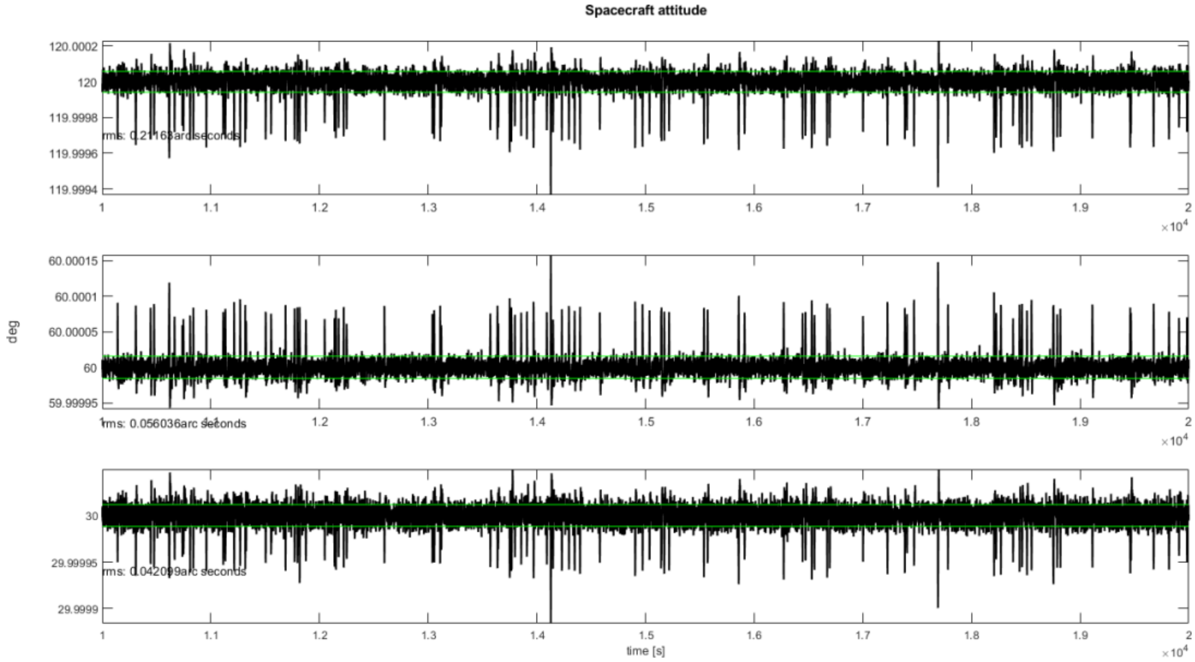


Figure 6: PI Controller: Spacecraft attitude

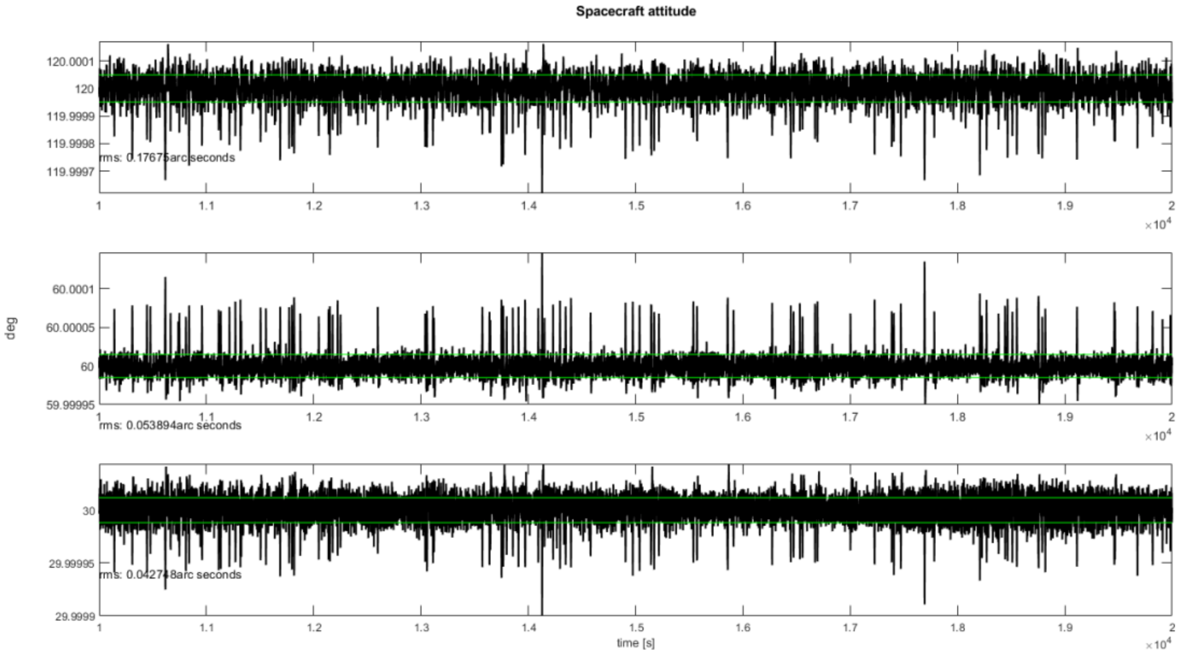


Figure 7: KF Controller: Spacecraft attitude

From Figure 5 to Figure 7 it can be seen that both the PI controller and KF reduce the error on the attitude during steady-state compared to the response without an internal RW speed control loop.

3.2.2 Torque spikes

In this case a similar input is given to the system, but this time a torque spike occurs at time t=9000 and lasts until t=11000. This torque spike represents a sudden increase in friction in the reaction wheel. The resulting spacecraft attitude is shown below for the three cases in Figure 8, Figure 9 and Figure 10.

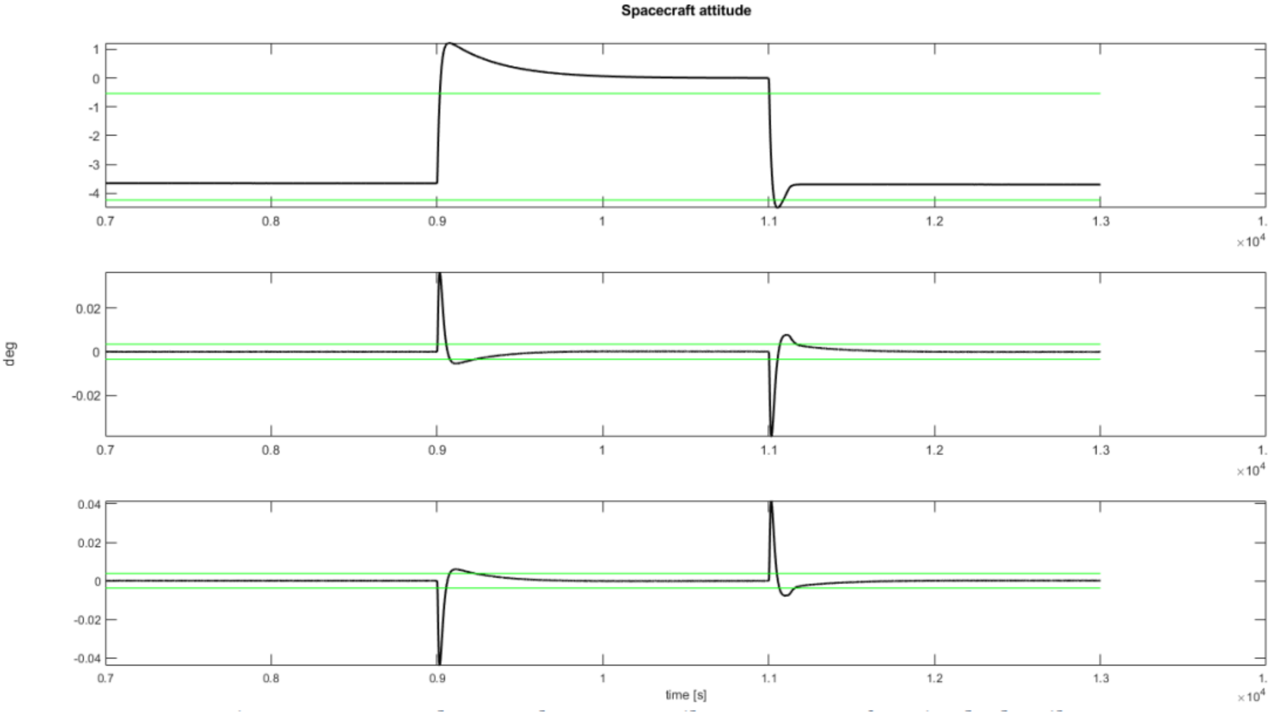


Figure 8: No RW speed control: Spacecraft attitude in response to torque spikes

As the reaction wheel does not have a speed controller, the motor does not immediately react by increasing the motor demand. When the reaction wheel slows down due to the increase in friction, the attitude of the spacecraft changes. The ADCS controller notices this change and controls the RW's to restore the attitude. Due to the higher friction torque and the fact that the integration action of the spacecraft attitude controller is limited to prevent integrator windup, situations can occur where the attitude is not fully recovered [6].

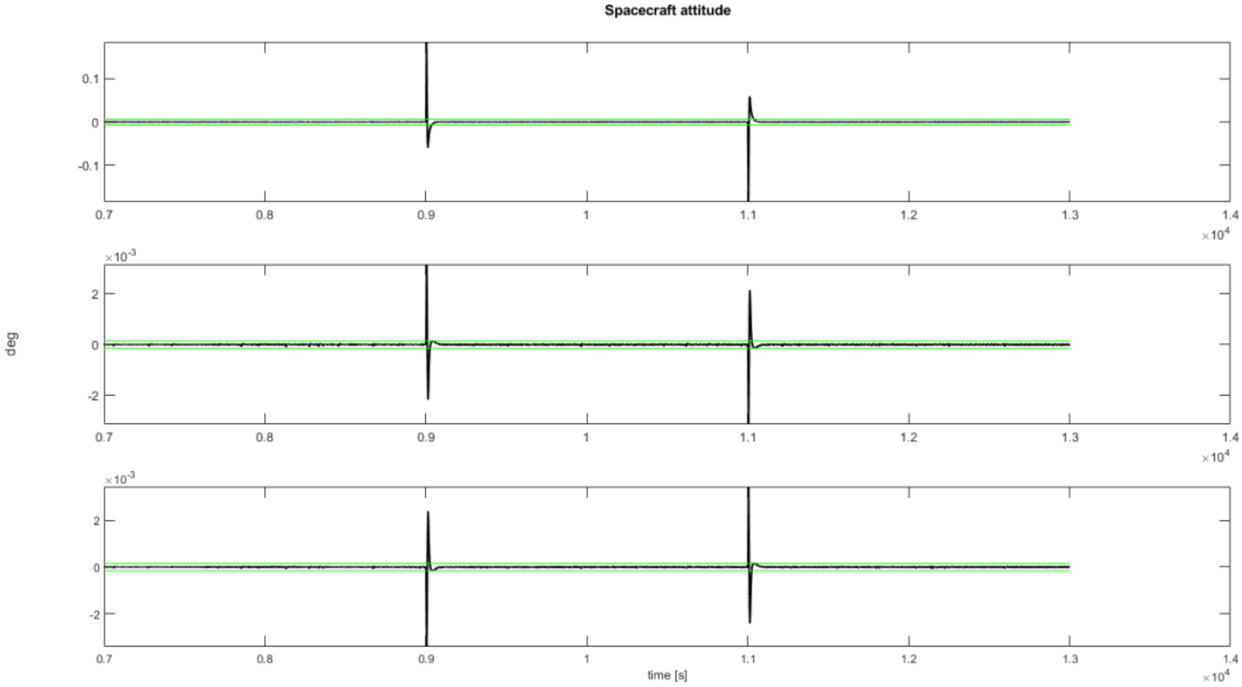


Figure 9: PI Controller: Spacecraft attitude in response to torque spikes

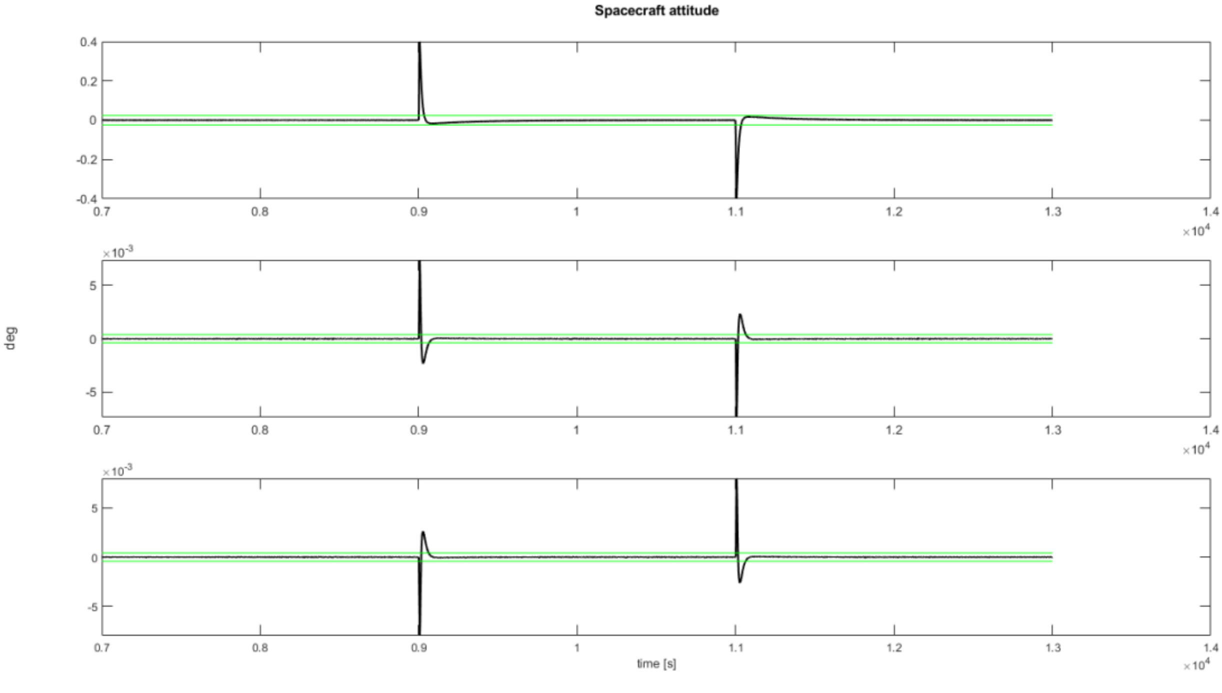


Figure 10: KF Controller: Spacecraft attitude in response to torque spikes

In Figure 9 and Figure 10, the attitude of the spacecraft is quickly restored after an occurrence of a torque spike. Both internal control loops detect the increase in friction and react by adding their estimate to the control torque, quickly restoring the angular velocity of the RW. Figure 11 and Figure 12 show the estimated friction torque (blue) plotted on top of the true added friction (black) for the PI controller and the KF controller respectively [6].

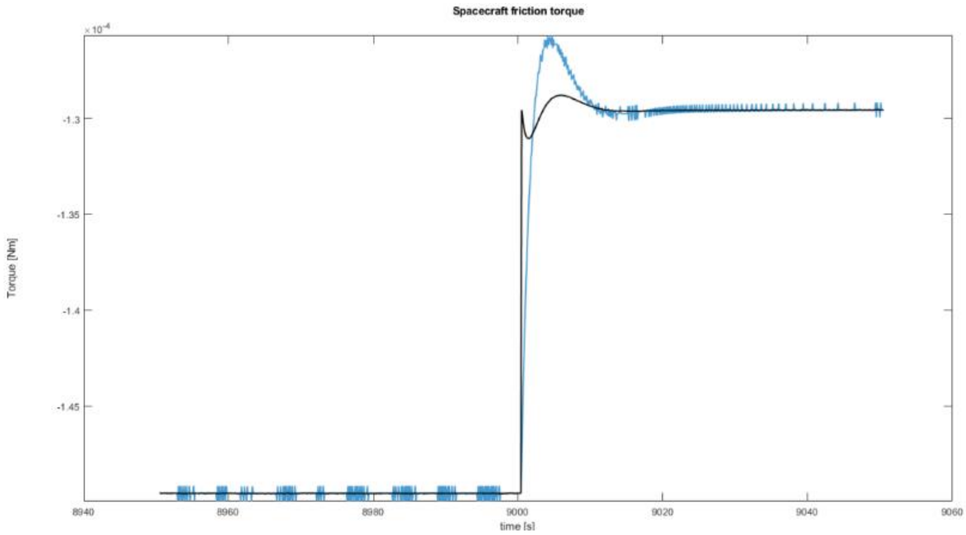


Figure 11: Estimated friction torque by PI controller

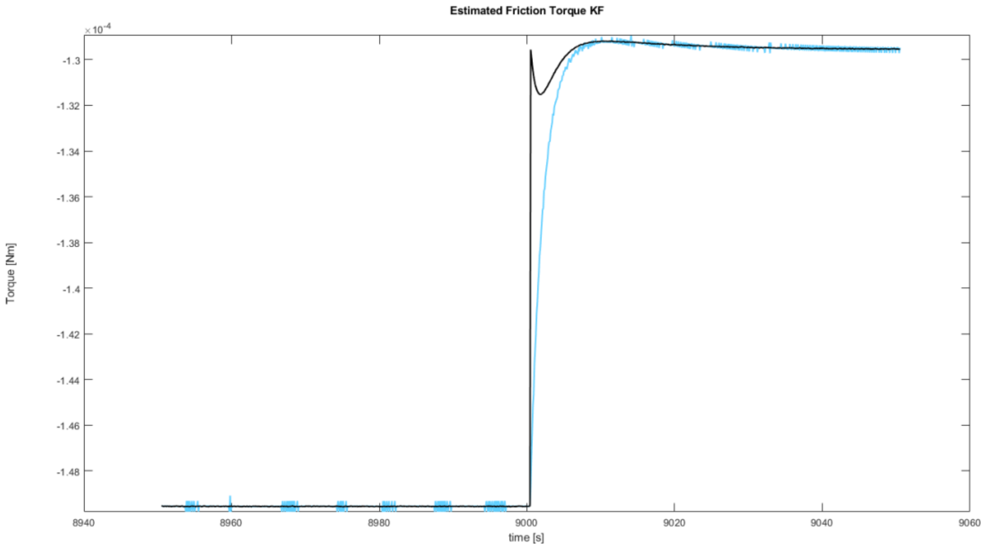


Figure 12: Estimated friction torque by KF

Both control strategies estimate the friction correctly and restore the attitude of the spacecraft. The KF is more complex and has a larger computational demand than the PI controller. Therefore, the PI controller was selected as the most practical and was implemented on a hardware set-up.

4 HARDWARE IMPLEMENTATION

4.1 Set-up and method

The set-up consists of a 12N14P BLDC outrunner, driven by an Electronic Speed Controller (ESC). The speed feedback is done with a magnetic encoder. The pulse train from the magnetic encoder is read and interpreted by a STM microcontroller. This microcontroller runs the control algorithms, which were translated from the original Simulink environment to C code. The output of this algorithm is then converted to a SERVO PWM which can be interpreted by the ESC to drive the motor. A schematic of the set-up is shown in Figure 13.

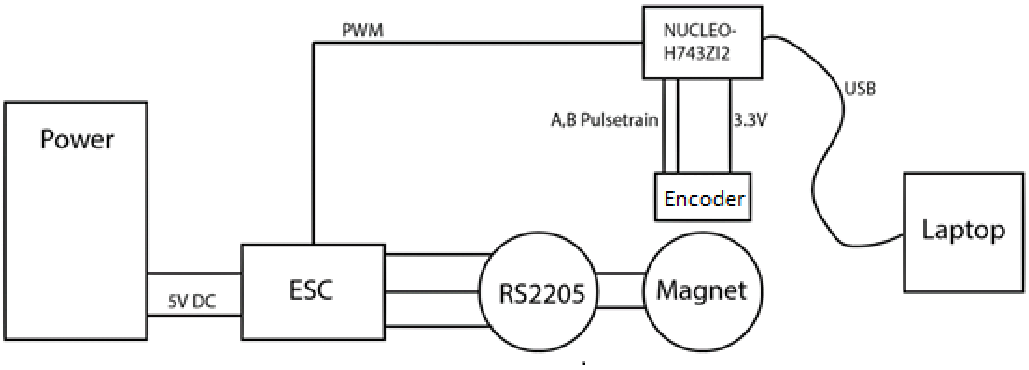
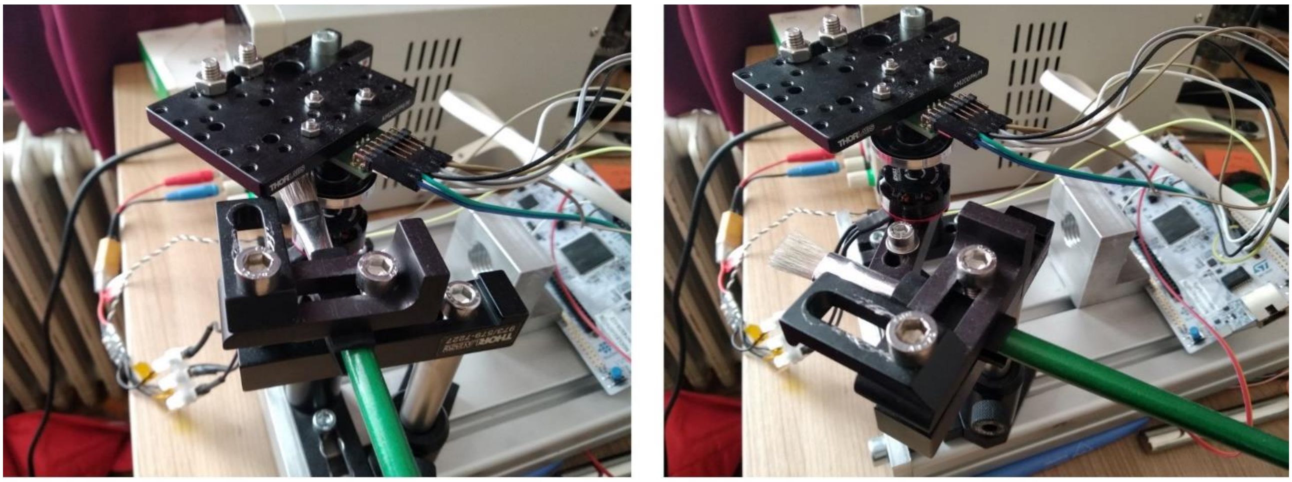


Figure 13: Schematic representation of hardware test set-up

Friction is applied to the system by pushing a paintbrush to the rotor to investigate the effect of sudden friction changes on the RW angular velocity. As shown in the simulations in chapter 3, changes in RW angular velocity have a direct impact on the attitude of the spacecraft. The observed variables are the measured angular velocity, the control signal outputted by the control algorithm, the pulse width fed to the ESC and the estimated friction in the system. The setup is shown in Figure 14 [7].



(a) Friction introduced via paint brush

(b) No friction introduced to the system

Figure 14: Test set-up for hardware validation

The motor was controlled the same way a RW is controlled by the ADCS controller. The ADCS controller sends a torque demand to the RW to change the spacecraft attitude. Similarly, the motor was first accelerated to the nominal RW speed. When no control torque is present, the added RW PI controller should ensure that the RW speed remains stable, regardless of external friction.

4.2 Results

Two scenarios were tested on the hardware set-up. First the torque demand was kept at zero, and the brush pushed against the rotor. This was done for different bandwidths and damping values of the PI controller.

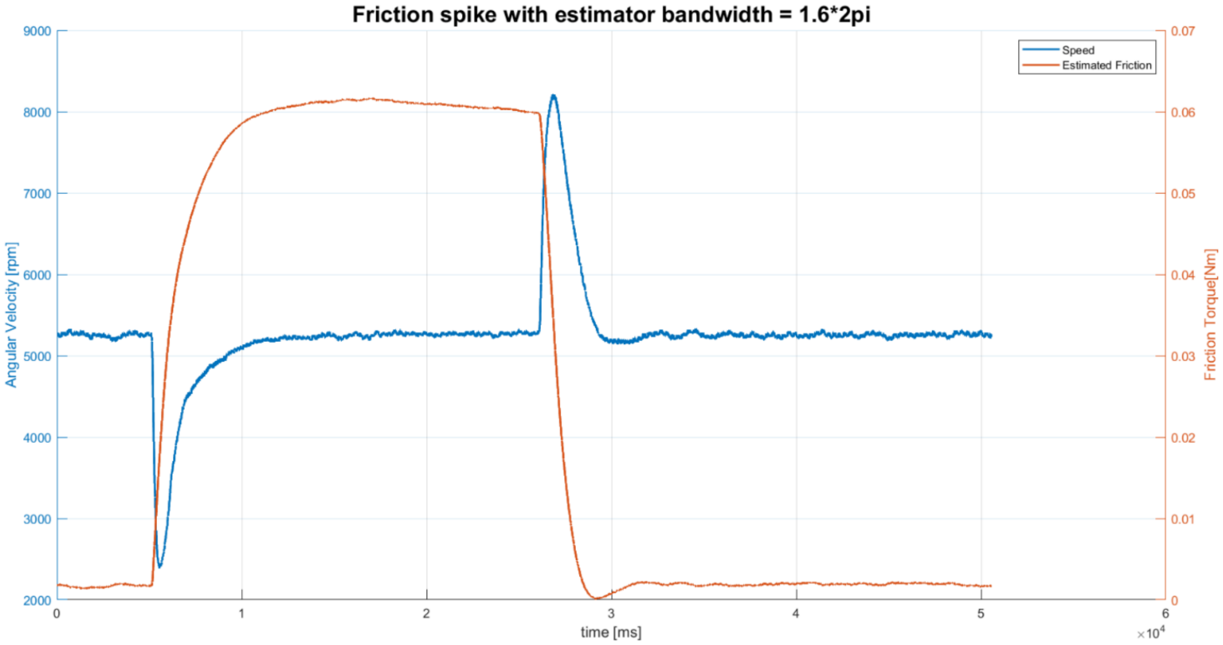


Figure 15: Response of angular velocity to friction increase for low bandwidth

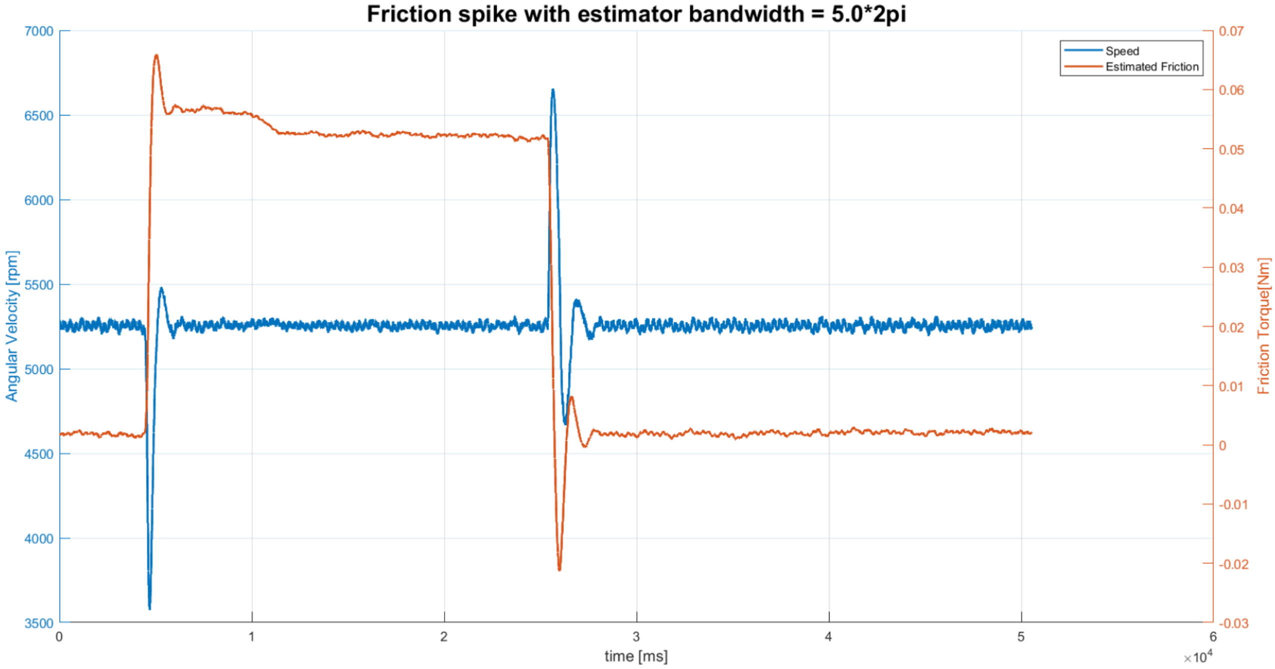


Figure 16: Response of angular velocity to friction increase for high bandwidth

From Figure 15 and Figure 16 it can be seen how the PI estimator responds similar as during the simulations. The sudden increase in friction, now applied physically by means of a brush rather than a disturbance signal as in the simulation, is estimated and added to the control torque to restore the angular velocity of the RW. A higher bandwidth results in a more aggressive controller. Settling time decreases and the size of the dip/spike in angular velocity decreases with increasing bandwidth. Overshoot increases with increasing bandwidth. Adding a higher damping to the system results in less aggressive response, larger settling times and less overshoot.

The second scenario that was tested is the response of the system, with the internal control loop, to a torque demand coming from the ADCS controller. The responses for a low and a high bandwidth are shown below in Figure 17 and Figure 18.

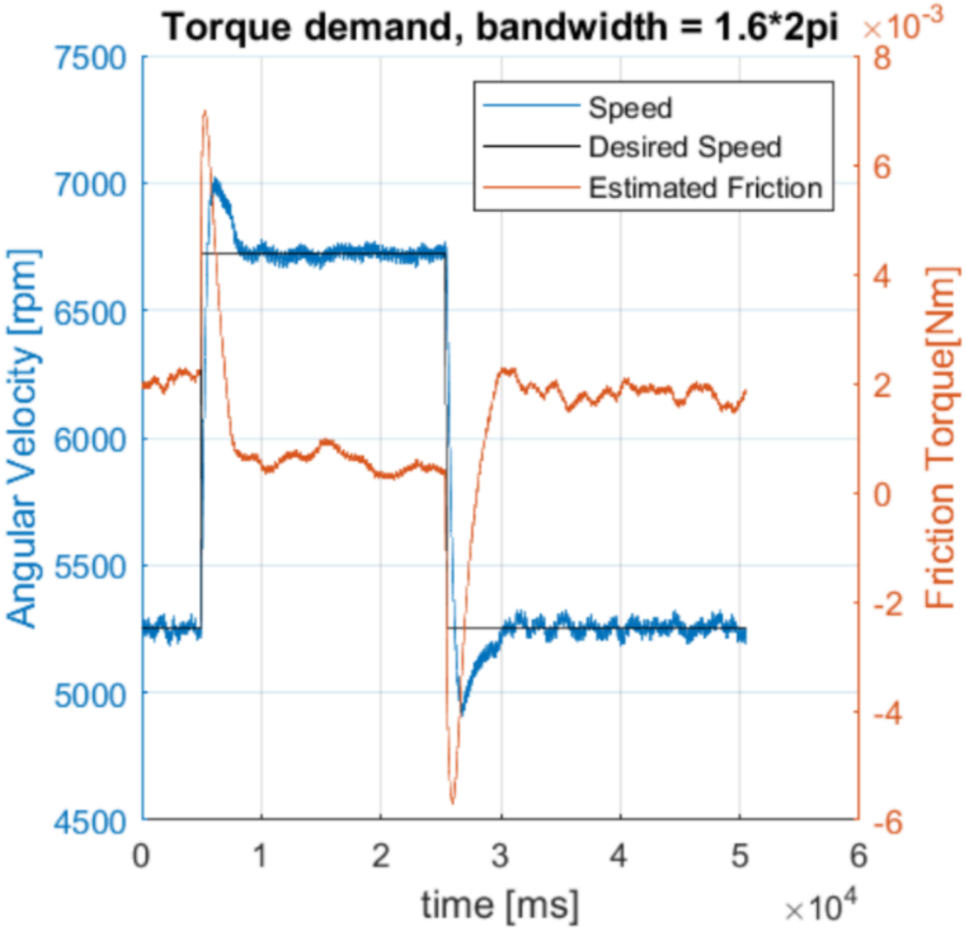


Figure 17: Response of angular velocity to control torque for low bandwidth

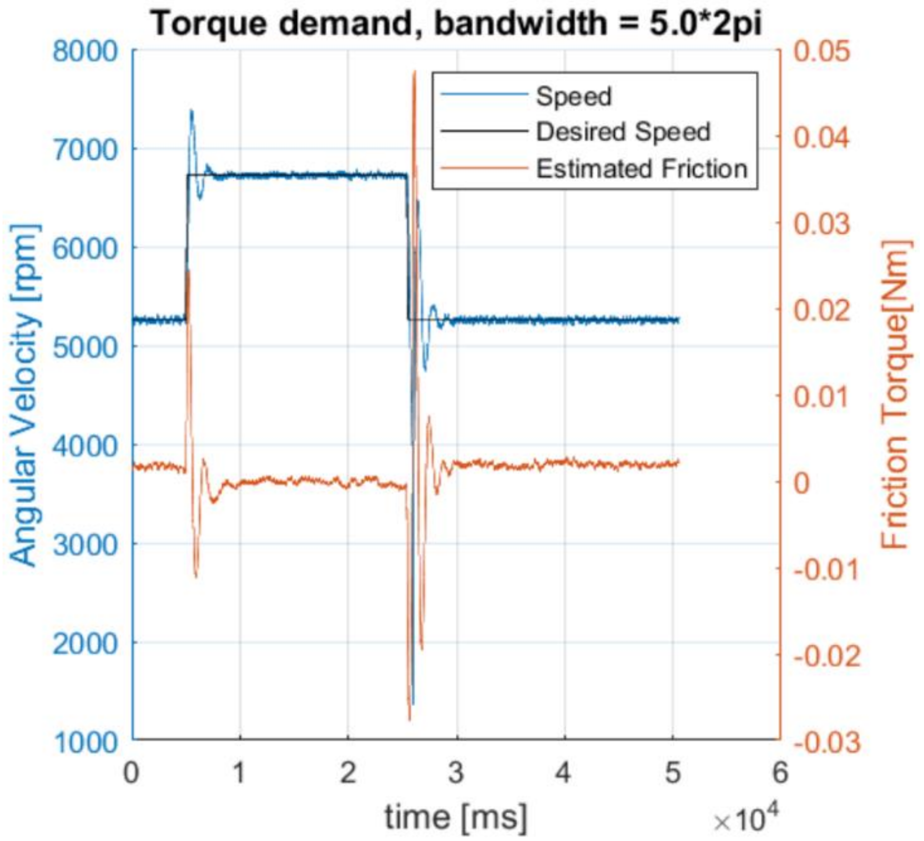


Figure 18: Response of angular velocity to control torque for high bandwidth

With the internal control loop, the RW still responds to control torque of the ADCS. Right after the torque demand, the estimated friction increases and overshoots, before settling.

For larger bandwidths within the tested range, the settling time decreases and the overshoot increases, resulting in more aggressive control. Higher damping ratios provide a faster response to a torque demand, while reducing the overshoot, at least in the tested range. Both overshoot and settling time are reduced for the higher damping values [7].

5 CONCLUSION

Changes in angular velocities of the reaction wheels directly impact the attitude of the satellite. Sudden changes in friction in reaction wheels change the angular velocity of these wheels and negatively impact the performance of ADCS system. The ADCS controller will eventually notice this change in attitude and act on it by demanding a control torque from the reaction wheel. This however results in a slow response to the friction increase and the ADCS might not be able to fully restore the attitude error in all situations. Having an internal speed control loop in the reaction wheels can help to solve this problem. The internal speed loop proposed in this paper monitors the speed of the reaction wheel and estimates the friction. Subsequently, the estimated friction is added to the control torque coming from the ADCS for spacecraft attitude control. When a sudden friction increase occurs, the estimated friction increases, increasing the motor demand and restoring the reaction wheel speed, independent of the ADCS controller. This results in a faster response to friction changes as compared to allowing the change in reaction wheel velocity to propagate through the ADCS controller by means of an attitude error. Such an internal control loop in the reaction wheels improves the performance of the ADCS.

6 REFERENCES

[1] A. Y. Lee and E. K. Wang, "In-Flight Performance of Cassini Reaction Wheel Bearing Drag in 1997–2013," *JOURNAL OF SPACECRAFT AND ROCKETS,* vol. 52, no. 2, pp. 470 - 480, 2015.

[2] G. F. M. Kitsch, J. Martin, M. Pantaleoni, R. T. Southworth, F. Schmidt, D. Webert and U. Weissmann, "Cage instability of XMM-Newton's reaction wheels discovered," in *12th Int. Conf. Sp. Oper. SpaceOps,* Stockholm, 2012.

[3] R. Seiler and A. Accomazzo, *"ROSETTA Reaction Wheels High Friction Torque Anomaly," in ESA-CNES Workshop on In-Orbit Anomalies,* 2010.

[4] T. Delabie, "System Design Report of Control Loop Software," arcsec, Leuven, 2020.

[5] T. Delabie, "System Design Report Simulation Environment," arcsec, Leuven, 2019.

[6] T. Delabie, "Test Campaign Procedures and Results of Control Loop," arcsec, Leuven, 2020.

[7] C. Cuypers, *Testing a reaction wheel speed control loop,* Leuven: KU Leuven; Gent University, 2021.