Institute for Dynamic Systems and Control

Shady Elshater, Philip Wolf

Satellite ADCS

Lab Practice Control Systems
Experiment Script

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology Zurich

February 19, 2024

Contents

1 Introduction

1.1 Background

The aerospace industry is seeing a transition to small-satellites in order to meet demand for increased access to space. The most common type of small satellites are cubesats, as the one shown in this picture.

Figure 1: CubETH – "small is smart" [1]

A 1 unit (1U) cubesat, is a $10 \times 10 \times 10$cm form factor, which can be put together for up to a total of 24 units, and which can carry any kind of sensor, scientific experiment, communication network, as well as thousands of other applications.
When cubesats are deployed, they spin and tumble. Many cubesats can work while spinning, but some need a precise attitude control which allows them to point at specific stars, or areas of the earth. For this reason, attitude determination and control systems (ADCS) are developed. Furthermore, propulsion systems are often useful to avoid collisions while covering larger areas, since cubesats are usually deployed in large groups, and to allow for de-orbiting at the end of their lifecycle, which is fundamental in the fight against space debris.
In general, to control the attitude, you either need a set of chemical propulsion thrusters, or devices such as reaction wheels or control moment gyroscopes, whereas the latter two do not allow for translational motions. Different types of chemical propulsion systems exist: solid, hybrid, liquid bipropellant, liquid monopropellant, and cold gas. For attitude control of a spacecraft, systems with a quick reaction time are needed, which means that most solid, hybrid and liquid bipropellant systems are not suited for this application. Finally, the advantages of a cold gas propulsion system over a liquid monopropellant system are that no combustion takes place and that the fluid supply system is very simple: The idea is to use nozzles to convert the high pressure gas stored in a tank, to a low pressure and high velocity stream which generates thrust.
In this laboratory practice you will get an introduction to a simple one-axis attitude control system for a 4U cubesat, based on a cold gas propulsion system.

[1]CubETH, *Small is smart*, https://espace.epfl.ch/research/past-projects/cubeth/

1.2 Goals

During this laboratory practice you will learn the basics behind a satellite ADCS (attitude determination and control system). The lab consists, in addition to a short homework, of four main parts:

1. PD Controller: You are given a PD controller and you have to change the K_p and K_d parameters, based on the different performance metrics provided. Due to the non-linearity of the valve controller, tuning methods such as Ziegler-Nichols will not work, and you will rather have to rely on your knowledge of how increasing a certain gain affects the response.
2. Pulse-Width Pulse-Frequency (PWPF): The focus then moves to the valve controller. As explained in Chapter 2, the solenoid valves used are on-off valves, which means that the continuous value of the PD controller needs to be converted into a discrete on-off signal. The so called PWPF method based on Schmitt-Triggers is one way to do so. In a similar fashion, you will be able to play with the different parameters of the Schmitt-Trigger and change the modulation of the signal.
3. Feedforward: In this part you will see how a simple feedforward control could be used for such a system. By knowing the inertia, torque, and target velocity, the system can be rotated with two single impulses.
4. Feedforward and feedback: Finally both strategies will be combined.

2 Setup

The structure consists of a round bottom plate on which the air bearing sits. Fixed to the rotor of the bearing there is the 4U cubesat. Most of the space inside the cubesat is taken up by the black storage tank which has a volume of 1L, and where the compressed air is stored at a maximum of around 12bar (tank certified for 200bar).
At the top of the tank there is the connection to the fill line as well as a pressure regulator, which brings whatever pressure (in the range $4 - 12$bar) is in the tank down to about 2.4bar. The outlet of the regulator is then split into 4 pipes connected to the 4 valves, placed at the top of structure. In between there is an electronic rack, where the microcontroller (MCU), the IMU, the bluetooth module, two voltage regulators and two MOSFET switch circuits are located.

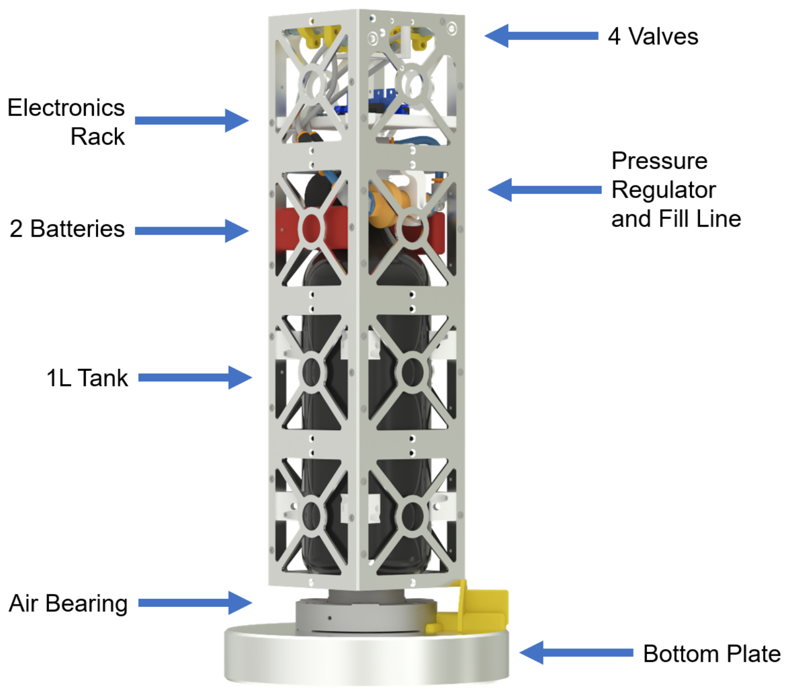

Figure 2: Overall Structure

The electronics scheme is shown in Figure 3. The ground station, on which the MATLAB® interface is running, is connected to the rotary encoder placed below the bearing.
The electronics of the satellite consist of 2 batteries and 2 voltage regulators powering a microcontroller (MCU), a bluetooth module, 4 solenoid valves that are switched by two MOSFETs, and an IMU (Inertial Measurement Unit). Most IMUs simply provide the angular acceleration and velocity around 3 axes, while **this type of IMU measures the angular velocity ω_m and estimates the angular position θ_m** by integrating the velocity (together with an EKF, or Extended Kalman Filter).
The encoder is not connected to the control loop, but only used to verify the angular position provided by the IMU after the end of each test.

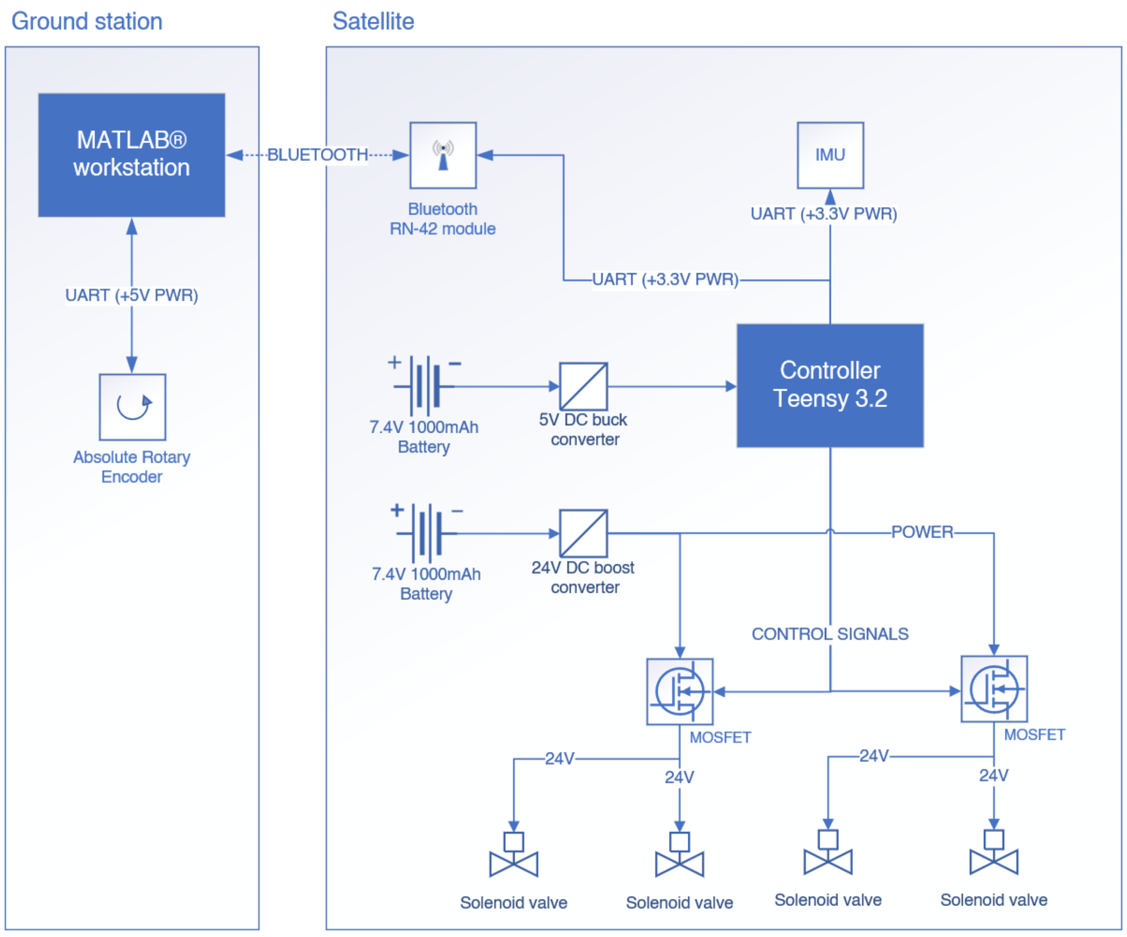

Figure 3: Electronics Overview

3 Homework

In this section you will be given a few exercises that you should solve at home before participating to the actual lab.
Start by downloading the *.zip* file. Open the *SatelliteADCS_Homework.mlx* in MATLAB®. All exercises can be solved directly using that livescript.
At the end of the homework, either save the entire livescript as PDF (Save -> Export as PDF) or bring your laptop with MATLAB® to show the livescript itself.

4 Simulink Modeling

4.1 Simulink Intro

The first part of this lab aims to show students some basic functionalities of Matlab Simulink.

Open the live script `"MATLAB_intro.mlx"` and work through it:

1. Open the Simulink model by clicking on the button `"open file"`
2. Solve the exercise.

   (If you don't know what blocks you need, go to Simulation -> Library Browser)
3. Save the File and close it
4. Check your answer by clicking on the button `"Test input"`
5. Once you have finished all tasks, click on `"Finish exercise"` at the bottom of the script and update your teaching assistant on your progress.

4.2 Simulation

1. Run the first section of the file `Simulation.mlx` in order to load the parameters that will be used in our simulation later.
2. Open the Simulink file `Model_Coded_Schmitt.slx` and double-click on the *PD-block*. Complete the PD-controller by using the variables Kp and Kd from the workspace.
3. Now, return to the overview and double-click on the *Dynamics-block* and complete the connections for the plant.
4. Return again to the overview and double-click on the *IMU-block* to complete the IMU functionality.
5. Finally, we want to code the Schmitt-Trigger block. Head over to the `Schmitt.m` file and implement the Schmitt-Trigger there. In the end, you will be able to run the entire simulation and you should obtain the same result as in the homework simulation.

5 Experiment

5.1 Warning

Air Bearing

The air bearing is a very sensitive and expensive part. It can break easily when operated wrongly. To avoid any damage keep the following points in mind:

- The bearing must not be turned if it's not pressurized. The air pressure on the bearing should always be at 4 ± 0.1 bar.
- The maximal load on the bearing in axial direction is 6 kg. **The system already weighs close to 3 kg, this means that only additional 3 kg can be applied.**
- All torques which are not in the direction of rotation of the bearing shall be avoided.

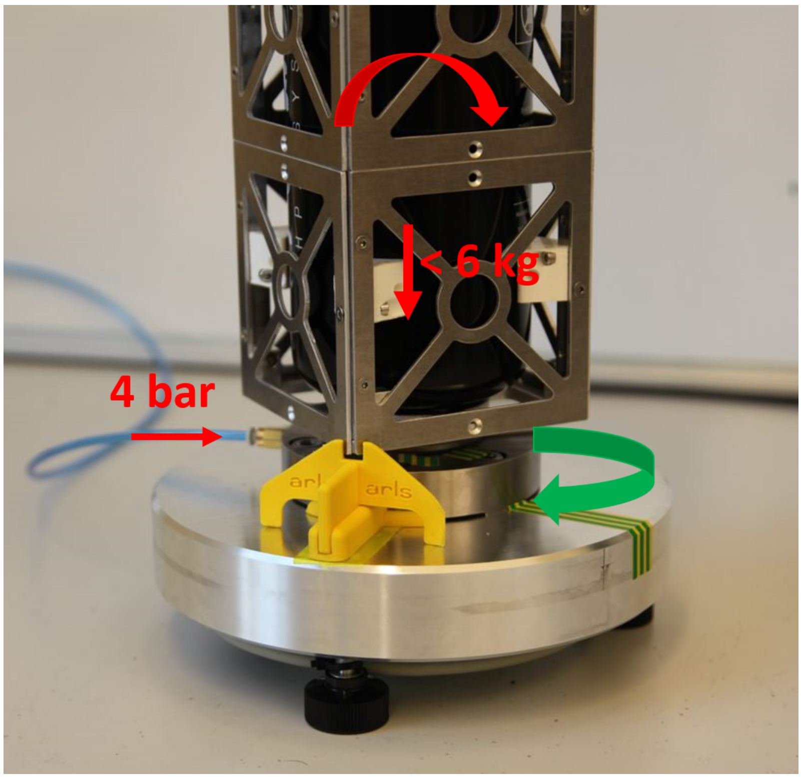

Figure 4: Air bearing warning

Pressure Tank

- In order to ensure the repeatability of your experiments, make sure that the tank pressure is > 4 bar before starting a test. (Although, the satellite has a pressure regulator with a constant downstream pressure of 2.5 bar (according to manufacturer), experiments have shown that the downstream pressure is not completely independent of the upstream pressure.)

5.2 PD Controller

Check the run tank pressure: $p_{\text{min}} > 4\,\text{bar}$**.**

1. In order to get a feeling for the system, run the system with the sinusoidal-sequence (*Testmode 4*). Everything can be managed from the `GroundStation.m` script There is no need to enter the PD/Schmitt gains, as this sequence works with predefined gains. What do you observe when comparing the IMU with the encoder measurement? How do you explain your observation?

2. Now you can test your own PD gains. To do so, choose the $0° - 30° - 0°$ sequence (*Testmode 1*). First, enter your P-gain while keeping the D-gain at zero. Run the system. What do you observe?
   Recall: $r(t) = k_P \cdot (\theta_{\text{ref}}(t) - \theta_m(t)) + k_D \cdot (-\omega_m(t))$

3. In the next step you can set the D-gain to the values found during the homework and run the system. Do you get a satisfactory result?

4. Try to find the optimal gains. For this application optimal means that the fuel consumption is as low as possible, as this is the limiting factor for the satellite's lifetime.
   Save the plot of the optimal response in the folder `Put plot here` as `YourName_PD.png`.

5. Open the live script `Simulation.mlx` and run the simulation of section 1 with the optimal PD-gains. Compare the simulation result to the response of the real system. What do you observe?

6. Now we want to find out about the effect of a P-gain that is too low. For that, return to the script `GroundStation.m`, set $K_P = 10$ and run the system. How can you explain your observation.
   Recall:
   $$r_{DB} = \frac{U_{\text{on}}}{K_m \cdot K_p}$$

7. Run the $0° - 30°$ sequence (*Testmode 5*) and and save the plot in the folder `Put your plot here` as `YourName_FB.png` in order to compare it later to the feedforward performance.

5.3 PWPF Modulator

Check the run tank pressure: $p_{\text{min}} > 4\,\text{bar}$**.**
Based on the formulas calculated in the homework, we now want to verify the effect that extreme values of the Schmitt-Trigger have on the physical system. Use the $0° - 30° - 0°$ sequence *(Testmode 1)* for the following exercises.

1. Set $K_m = 6, U_{\text{on}} = 0.5, U_{\text{off}} = 0.3$. What do you observe? Why?
   Recall:
$$r_{DB} = \frac{U_{\text{on}}}{K_m \cdot K_p}$$
2. Set $K_m = 4, U_{\text{on}} = 0.5, U_{\text{off}} = 0.02$. What do you observe?
3. Go to the live script `Simulation.mlx` and run the simulation of section 2 with the parameters $K_m = 4, U_{\text{on}} = 0.5, U_{\text{off}} = 0.02$. Try to find an explanation for the behaviour with low U_{off}.
   Hint: take a look at the valve torque response of the simulation. And recall that
$$\Delta_{\text{min}} = -\tau_m \cdot \ln\left(1 - \frac{h}{K_m}\right)$$
   gives the minimum impulse width.
4. Switch back to the `GroundStation.m` script. Set $K_m = 4, U_{\text{on}} = 2, U_{\text{off}} = 1.8$. What do you observe? Why?
5. Set the parameters back to the default values: $K_m = 4,\ U_{\text{on}} = 0.5,\ U_{\text{off}} = 0.3$.

5.4 Feedforward

Check the run tank pressure: $p_{\text{min}} > 4\,\text{bar}$**.**

1. First let's validate the values for the feed forward calculated during the homework. To do so, go to section 3 of the `Simulation.mlx` script, change the parameter for $\Delta t_{\text{imp}}, \Delta t_{\text{del}}$ and run the section. Does the valve opening time correspond to what you would expect?
2. Now it's time to check the feedforward parameters with the real system. Go back to the `GroundStation.m` script and choose the feedforward mode (*Testmode 2*) and enter $\Delta t_{\text{imp}}, \Delta t_{\text{del}}$. How close does the satellite get to the target position of 30°?
3. Try to adjust your parameters in order to get as close to the target position as possible. Hint: One possible approach is to find the error and adjust Δt_{del} by changing θ_2 accordingly. How close can you get to the target? How does the fuel consumption compare to the one in feedback mode? Compare your result with the feedback response.
4. In reality, a combination of feedforward and feedback would be used to have a trade-off between low fuel consumption while still reaching the target position accurately enough. Choose the feedforward+feedback mode (*Testmode 3*) and enter your best parameters for both controllers. Run the system and observe how it behaves? How is the performance of the combined controller?

The following pseudocode explains how the feedforward+feedback mode works:

Algorithm 1: Feedforward+Feedback Mode

```
$FFstate = true$;
read $K_p, K_D, K_m, U_{\text{on}}, U_{\text{off}}, \Delta t_{\text{imp}}, \Delta t_{\text{del}}$;
while $true$ do
    get $\theta_m(t)$ and $\omega_m(t)$ from IMU;
    send $\theta_m(t)$ to ground station;
    if $FFstate == true$ then
        call $FFfun(\Delta t_{\text{imp}}, \Delta t_{\text{del}})$;
    else
        get $\theta_{\text{ref}}(t)$ from ground station;
        calculate $r(t)$ with PD;
        give actuation signal via Schmitt;
```

Function FFfun

```
update $timer$
if $timer < \Delta t_{\text{imp}}$ then
    give impulse
else if $timer < (\Delta t_{\text{del}} + \Delta t_{\text{imp}})$ then
    close all valves
else if $timer < (2 \cdot \Delta t_{\text{imp}} + \Delta t_{\text{del}})$ then
    give counter impulse
else
    $FFstate = false$
```