THE UNIVERSITY OF TEXAS AT AUSTIN

Final Project

ASE 372K ATTITUDE DYNAMICS

Due on December 6th

ATTITUDE DETERMINATION AND CONTROL SYSTEM

The purpose of the final project is to simulate, estimate, and control the attitude of a satellite via a .

A satellite's ADCS (Attitude Determination and Control System) is composed of software and hardware. In your computer code, you must clearly divide the portions of the code pertaining to the software part, which is what we, as engineers, write and upload to the satellite to have it operate as we wish; and the hardware parts (sensors and actuators), which are simulated versions of the mechanical operation of the real hardware mounted on the satellite. Your computer code must also have a clearly divided portion to simulate the dynamics/kinematics the satellite experiences; these represent the true space environment, and are not part of the flight software uploaded to the satellite. Therefore, the project must be compartmentalized, either thorough different functions/routines (if you use a scripted programming language) or different blocks (if you use a graphic programming language such as Simulink).

The student must:

- Write a project report that describes the work performed.
- Upload the code with clear instructions on how to run it and how to duplicate the results. The code must be self contained in a single compressed folder uploaded to canvas. A random subset of projects' codes will be tested by the TAs. You are allowed to use Matlab toolboxes. If you use programming languages other than matlab/simulink, you must provide library-functions you use in your code. If you use software other than matlab/simulink, please make sure to visit the TAs to insure they will be able to run it.

- Choose a reference low-earth orbit (LEO) trajectory for the satellite and a three-axis reference attitude profile. The report must clearly state which orbit and attitude profile are used. The report must specify a realistic inertial matrix of the satellite that is in line with existing satellites orbiting LEO. Example of attitude profiles are:
  - Inertial pointing (easiest choice, you specify which inertial direction each of the 3 body axis must be aligned with)
  - Earth pointing (you specify which body-fixed axis must point towards earth, and which one must be aligned with the velocity (assuming a circular orbit)
- Simulate the attitude dynamics/kinematics including at least two types of torque perturbations (e.g. gravity gradient, solar radiation pressure, atmospheric drag, magnetic torques). The report must contain a plot of the evolution of these perturbations and a plot of the evolution of the satellite's attitude when the control system is off. The purpose of this plot is to make sure that the system indeed needs active control. The simulation must be for at least one full orbit and no less than 90 minutes. If you use code from the internet or not written by you in order to simulate perturbations (e.g. a Earth's magnetic field model) you must clearly cite each and every source you are relying on.
- Choose an actuator of reasonable size and performance. Example of actuators are:
  - Control Moment Gyros (CMGs)
  - Reaction Wheels (RWs)
  - Reaction Control System (RCS)

Hardware components (sensors and actuators) must be able to provide the system with the desired torque and measurement accuracy. Thus, it is important to provide performance specification of each selected hardware component to demonstrate its realistic performance. You must include math models for all hardware selected and clearly specify all values for the models, e.g. max torque, measurement accuracy, etc.

At a minimum, the report must contain the following sections:

1. **Introduction** This section contains a brief description and overview of the work.
2. **Reference Mission** This section contains the reference orbit of the satellite together with the reference attitude. In this section you must specify the inertia matrix of the spacecraft which must be realistic and inline with real satellites performing your selected type of mission.
3. **Dynamics Model** This section must contain the dynamic/kinematic equations of attitude motion (you can choose whichever attitude parameterization you prefer) together with the mathematical description of the model you use to simulate at least two torque perturbations. This section contains the time evolution of the uncontrolled attitude due to the perturbing torques when no active control is included. The uncontrolled attitude must diverge from the reference attitude.

4. **Sensors and Actuators** This section must contain a description of the sensors and actuators chosen as well as their performance specifications. Realistic and reasonable numbers must be provided. A mathematical model of the equations you use to simulate the sensors and actuators in your code must be included.

5. **Attitude Determination and Control System (ADCS)** This section must contain a description of your attitude determination system (I strongly recommend TRIAD) and attitude controller (I strongly recommend a PD controller). If the ADCS requires choosing any tunable parameter, you must specify the values you use in this section of the report. The attitude parameterization used by ADCS need not to be the same used to simulate the true dynamics of the satellite.

6. **Simulation Results** This section must include plots showing that the ADCS works and that the satellite is indeed maintaining the reference attitude.

The ADCS cannot in any way receive or pass data directly with the simulated dynamic/kinematic environment. The simulated "true" attitude and angular velocity from the environment cannot be fed to the ADCS system; rather a sensor must take these true quantities and simulate measurements corrupted by random noise. Similarly, the commanded torque from the controller cannot directly affect the true dynamics, rather an actuator must take the control output and simulate torques affected by small actuation errors.
The attitude of a satellite can be represented in many different ways (quaternions, Euler angles, DCM,...): students are free to choose the representation that best fits their project. **You do not need to simulate the orbital motion (unless you want to)**. You can assume the satellite is following exactly its reference orbit. Since the position and velocity of the satellite might be needed to compute perturbations and perhaps to compute the reference attitude; you can assume that at each given time the satellite's position and velocity are exactly those of the nominal orbit at that time.
The hardware selection is free as well. The student can decide which type of sensors/actuators to use and their accuracy/maximum torque. It is strongly suggested to include a gyro as one of the sensors to directly measure angular velocity. The angular velocity is needed to compute the "D" part of the PD controller. The students are not allowed to directly simulate a measurement of attitude, e.g. of a quaternion. Rather, measurement should be directions, e.g. to the sun, earth, stars, local magnetic field, and an attitude determination algorithm such as TRIAD should be used to estimate attitude from directions.

Students are free to choose any programming language to simulate the ADCS. Simulink is strongly recommended but <u>not mandated</u>. Simulink usually achieves a faster simulation w.r.t. scripted Matlab and gives a clear graphical representation of the different conceptual parts of the ADCS system required by this project. If you choose a language other than Simulink, it is <u>your</u> responsibility to ensure the TAs are able to replicate your results by running your code themselves.

The project's grade will be determined by the overall quality of the work and of the report. Meeting all the minimum requirements specified in this document does not guarantee 100 points. Rather, it is the responsibility of the student to write a coherent, professional, and exhaustive report that fully and clearly demonstrates the validity and performance of their ADCS.

If you have any doubts the correct way to proceed is come to office hours and ask the instructor. If anything is unclear about the project requirements please come and ask.

Delivery Instructions

The final project must be turned in electronically through Canvas by the due date. The final upload must consist of two files:

1. A PDF file containing the project reports. If you write in Word, please save it as PDF for delivery.

2. A single compressed folder containing all the source code needed to run your project. The compressed folder must contain a readme file. The readme file must specify:
   - The programming language used to write the simulation
   - Any special libraries and/or toolboxes used by your code
   - (If applicable) the compiler used to compile your code
   - <u>Detailed</u> instructions on how to compile/run the code to exactly reproduce the results you show in the report

The TAs will pick a random set of projects and run it to confirm the results you show in the report. **It is your responsibility to ensure the TAs are able to exactly replicate your results.** You are welcome to go to the TAs office hours and run your code with them.

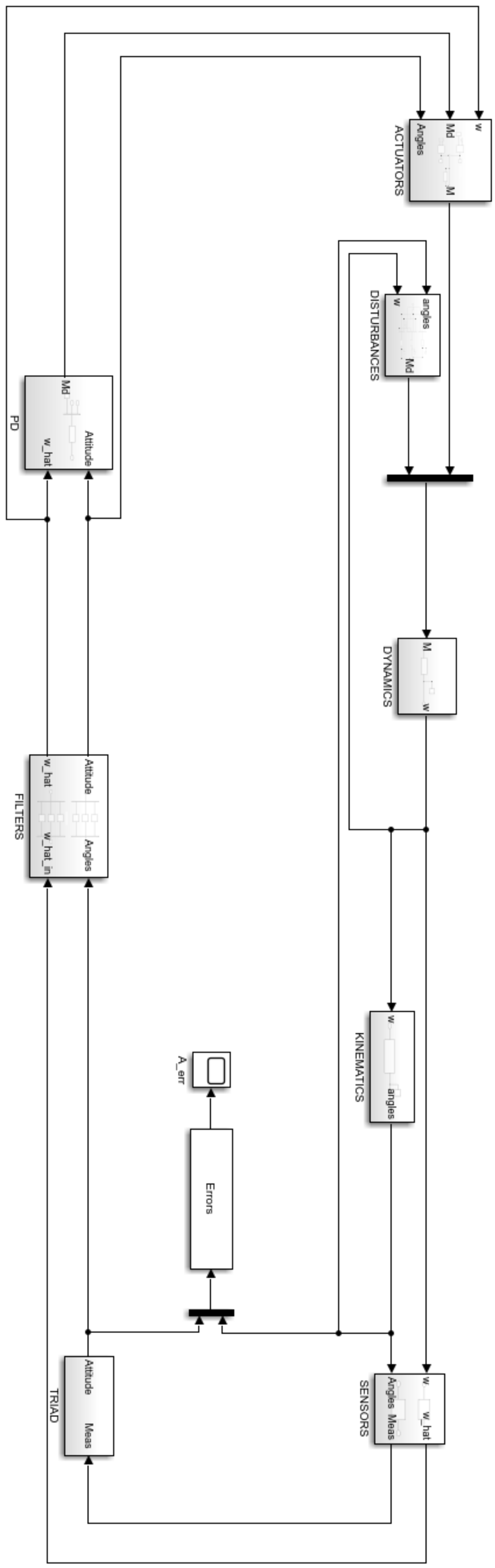

Figure 0.1: ADCS Example