Relative Positioning and Tracking of Tethered Small Spacecraft Using Optical Sensors

AE 8900 MS Special Problems Report
Space Systems Design Lab (SSDL)
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

Author:
Yanjie Guo

Advisor:
Prof. Brian C. Gunter

December 15, 2018

Relative Positioning and Tracking of Tethered Small Spacecraft Using Optical Sensors

Yanjie Guo *
Georgia Institute of Technology, Atlanta, GA, 30313, USA

Advisor: Brian C. Gunter†
Georgia Institute of Technology, Atlanta, GA, 30313, USA

This study analyzes the attitude dynamics and tracking control system for an object attached via tether to a primary spacecraft using optical sensors. Requirements of the tracking include body pointing of the main spacecraft to the tethered target, in addition to optimized sun pointing for maximum power generation from the primary spacecraft's solar panels. This study shows that a modified PD controller with 3 reaction wheels can track the tethered object given only relative position information, i.e. the relative target position and the sun pointing vector in the body frame, generated from an optical camera and sun sensor.

Nomenclature

BBB	=	BeagleBone Black
FoV	=	Field of View
HWIL	=	Hardware-in-the-loop
LEO	=	Low Earth Orbit
PD	=	Proportional-Derivative
SRP	=	Solar Radiation Pressure
SSE	=	Steady-State Error

I. Introduction

In many cases, in particular for small satellites, there is not the mass, volume, or power available to have a separate pointing mechanism for the mission's payload instrument. As such, the targeting of the payload sensor is often done with body pointing of the main spacecraft, but this also needs to be balanced with the need to maintain an optimal sun pointing angle for maximum power generation. For dual CubeSat systems, or formation flight missions in LEO, the desired attitude is constantly changing due to the relative dynamics, which usually tends to be periodic.[1] These

*PhD Student, Daniel Guggenheim School of Aerospace Engineering.
†Assistant Professor, Daniel Guggenheim School of Aerospace Engineering, Senior Member AIAA.

constraints require the use of a robust 3-axis attitude control system, which is assumed here to utilize three reaction wheels. This study investigates a scenario in which a primary spacecraft is tracking a secondary (tethered) object using only relative attitude information obtained from an optical camera and a sun sensor. The results demonstrate that the control system can successfully track the target using a modified PD controller with 3 reaction wheels. The tracking is maintained without the use of absolute attitude information such as horizon sensors or star cameras.

This analysis mainly focuses on the Tethering And Ranging mission of the Georgia Institute of Technology (TARGIT) CubeSat shown in Fig. 1.

Fig. 1 TARGIT CubeSat mission.

The TARGIT CubeSat[*] is chosen by NASA in 2016 to participate in the CubeSat Launch Initiative (CSLI), which guaranteed participating nanosatellites a spot as a secondary payload aboard an upcoming NASA-supported rocket. Once built, the solar-powered 3U TARGIT CubeSat will house a compact miniaturized LiDAR altimetry system capable of delivering accurate cm-level topographic data from as far away as 10 kilometers.

TARGIT will utilize a combination of commercial off-the-shelf (COTS) CubeSat components and custom, in-house developed payloads to assess the performance of the LiDAR system, whose ultimate objective is to enable future small-satellite-based planetary missions to gather valuable topographic, navigation, and reconnaissance data with a low size, weight and power (SWaP) instrument.

The whole TARGIT mission has several phases as shown in Fig. 2. Once TARGIT is launched into the target LEO in space, after detumbling and initial acquisition and system checkout, it will deploy a tethered 1-meter inflatable tetrahedral target at a distance of 6-10m, and create its 3D image using the LiDAR system. Since the target's size and

[*] `https://bgunter.gatech.edu/research/targit/`

reflectivity will already be well-known, the accuracy of the LiDAR camera can be tested.

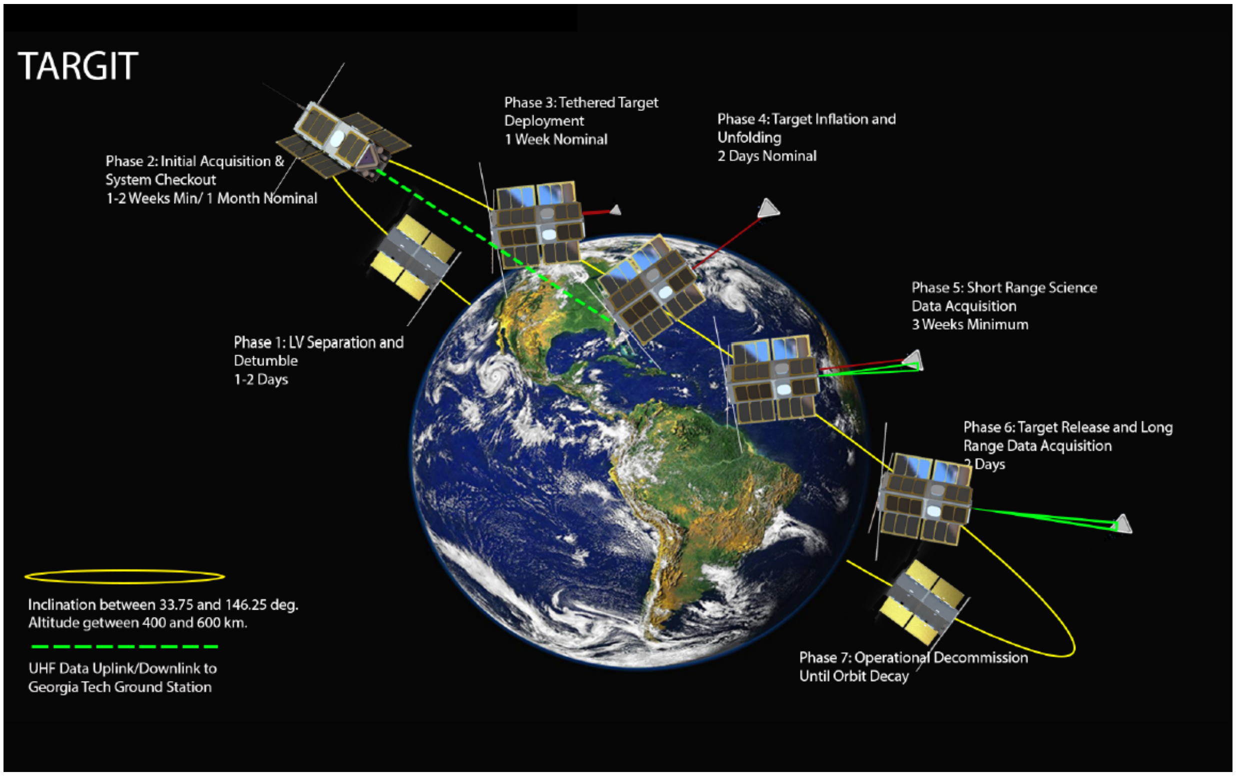


Fig. 2 TARGIT mission phases.

Eventually, the tether will be cut, and the satellite will continue capturing images of the inflated target as it drifts farther and farther from the satellite. This final phase is intended to represent a flyby scenario in a traditional space mission, where the topography of a comet, asteroid, or specific surface location is the desired target.

In order to achieve this, the primary 3U spacecraft needs to continuously point the LiDAR system at the target. Although the depth and the relative angles of the target can be obtained directly from the LiDAR system, this information cannot be used for attitude control because of its relatively narrow Field of View (FoV) and low resolution. Due to the relative dynamics, the target will constantly drift within the FoV of the LiDAR. Once the target is outside its FoV, it would be difficult to reacquire the target using the LiDAR system. Therefore, an optical sensor with a much wider FoV such as an on-board camera that is capable of object tracking is needed.

The camera chosen is a commercial-off-the-shelf (COTS) product with built-in image tracking capabilities[†]. It is located in the front of the CubeSat and can be used to detect and track objects with hues and return their sizes (height and width) and the locations in the image plane. Its FoV is 40 (height) x 60 (width) degrees, which is much larger than the LiDAR system. Also, it can work at 60 Hz, which is more than is needed to determine the relative position. The resolution of the camera is 315 (width) x 207 (height), which is enough for the tracking requirement

[†]https://pixycam.com/pixy2/

within approximately 15 meters. Since the actual size of the target is known, the distance of the target can be determined using the image size information from the camera; however, the target distance is not needed for attitude control. Only the relative direction matters. One item to note is that when the target is first released less than approximately 1.5 meters away, it will be too large in the image plane. The tracking process will therefore only begin when the target is far away enough and necessary to track.

As mentioned above, the tracking camera detects objects based on their hues. The built-in object tracking of the camera works best when in moderate lighting conditions (not overly bright or dark) and when the target has a distinct color, such as green or orange, as opposed to white, gray or black. This places natural conditions on the material selection of the target, as well as when the target can be tracked (i.e., outside of the eclipse period). Workarounds for this may be possible, such as optical filters and active illumination of the target, which are currently under investigation for TARGIT, but are outside the scope of this study, which is focused purely on the tracking of the target.

The tracking camera can provide the relative position of the target in the image plane, which can be converted to a primary pointing vector. In order to determine the desired attitude, a secondary pointing vector is found to maximize the solar panel efficiency, i.e. the normal vector of the solar panel should be pointing at the sun, which can be provided by a sun sensor. The primary pointing vector generates pitch and yaw attitude reference, while the secondary pointing vector determines the roll attitude reference. Then, the controller will output torque commands according to these reference signals.

Since the relative dynamics of two objects in LEO tends to be periodic, so should be the control signals. It doesn't require much control to track the target once it is locked within a small region in the center of FoV as will be shown in the result. Thus, reaction wheels are suitable actuators for the attitude control system because it only uses electricity as power. Ideally the three reaction wheels will be aligned with the principal axes of the CubeSat.

The control system is a modified PD controller. Normally it is preferred to decouple the system using absolute attitude information.[2, 3] But simulation shows that with only relative attitude information (error signals), only two wheels are required to track the target.[4] A third wheel can be added to maintain an optimal solar angle. To account for the unknown disturbance or incorrect system parameters, the controller can be made adaptive.[5] The tether system can even be used to help stabilize the attitude.[6] A series of hardware-in-the-loop (HWIL) experiments were then used to test the controller.

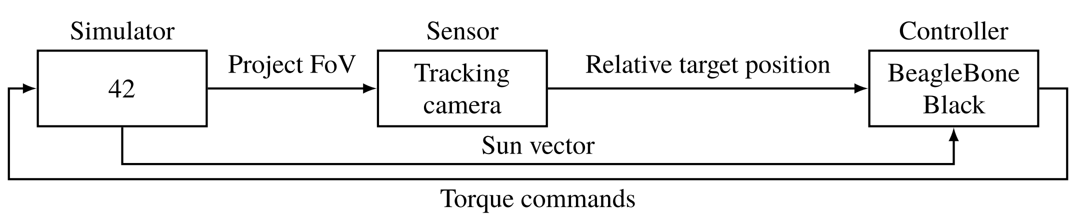


Fig. 3 Hardware-in-the-loop simulation setting

A block diagram of the HWIL experiments is shown in Figure 3. The software package 42[‡] serves as the simulator that propagates the dynamics of the states, generating a frontal FoV of the CubeSat and projecting it with correct size. The tracking camera is fixed in front of it with a correct distance corresponding to the projection size as if it is watching the actual space on board. The BeagleBone Black (BBB) serves as the controller/computer that receives and processes data, the relative position from tracking camera, and the sun vector from 42. Then the torque commands are computed and sent back to 42 for the next propagation, thereby closing the loop. The main challenge for the HWIL simulation is the measurement noise and possible loss of target signals, which can result in instability. The baseline controller now has passed some basic tests, including tracking the target in a stable manner, and reacquiring the target when it gets outside of FoV and reenters. Its performance will be demonstrated under additional testing scenarios, such as when the target experiences high dynamical motion, or when the target/spacecraft enters the eclipse, or when the target is not currently visible, and a search is initiated.

There are several metrics for the performance. The most important one is the steady-state error (SSE), which tells the final offset in the tracking. The settling time is also interesting, because a very slow response is not desired. Some other metrics are overshoot and the L2 norm of the error signal, since it is better to avoid excessively oscillating response to the noise and obtain as smooth a performance as possible.

The main contribution of this work is the buildup of the hardware-in-the-loop simulation and an attitude control system that only use relative attitude information to track certain target that works under the testing scenarios.

II. Methodology

Usually in an attitude tracking and control problem, the objective attitude of the spacecraft is independent on its own attitude dynamics, i.e. controlling the attitude of the spacecraft wouldn't change the objective attitude. In most missions, the objective is to point the scientific equipment at some target. For example, point the camera or telescope at some interesting asteroids or stars; or point the solar panel or solar sail towards the sun for maximum efficiency; or point the thruster in a desired direction to perform some maneuvers. For this type of problems, a general solution is to design a PD controller that decouples the attitude dynamics using absolute attitude information, either in terms of Euler angles or quaternions.

But for a tethered system, the relative motion of the target can be affected by the attitude dynamics of the spacecraft. For example, a constant rate pitch or yaw motion of the spacecraft will drag the target through the tether to also rotate around it. On the positive side, this makes the target cooperative, i.e. there is some way to control at least a subspace of the target states. However, this impact on the target can also lead to undesired complicate relative dynamics. In the worst case, it can cause system instability and result in the tracking failure.

In this specific study, it is assumed that during deployment, the target is released in a slow and smooth manner, and

[‡]`https://software.nasa.gov/software/GSC-16720-1`

the relative motion is mild. In addition, due to the system configuration shown in fig (add a figure of the configuration here), the target is tethered at the center of the frontal surface, next to the camera. Pointing the camera at the target wouldn't draw the tether too much, and hence change the target motion very much. Therefore, the tethered system is treated as a black box that doesn't have a huge effect on the relative dynamics. And the objective becomes to track an arbitrarily moving target.

III. Controller

A. System dynamics & traditional solution

The nonlinear attitude dynamics is shown in Eqn. 1, where I and J_w are the inertial matrices for the satellite and reaction wheels, ω_b and Ω_w are the angular velocity vectors, T_d is the disturbance torque which we assume to be 0.

$$I\dot{\omega}_b + J_w\dot{\Omega}_w + \tilde{\omega}_b(I\omega_b + J_w\Omega_w) = T_d = 0 \tag{1}$$

Since it is assumed that the reaction wheels are aligned with the principal axes of the spacecraft, it is best to describe the rotation as an Euler-like angle sequence, i.e. pitch, yaw, and roll. Define the body frame as x pointing forward, y pointing right, and z pointing down. The pitch, yaw, and roll angles $\phi,\ \theta,\ \psi$ becomes $\theta_X,\ \theta_y,\ \theta_z$.

Assume the reaction wheels take torque commands as the control input, the nonlinear system can be rearranged to separate the reaction wheel torques.

$$I\dot{\omega}_b + \tilde{\omega}_b I\omega_b + \tilde{\omega}_b J_w\Omega_w = -J_w\dot{\Omega}_w \tag{2}$$

Expressing Eqn. 2 in the inertial frame leads to the full nonlinear coupled attitude dynamics system.

$$I\dot{\omega}_{b/N} + \begin{bmatrix} (I_z - I_y)\omega_z\omega_y \\ (I_x - I_z)\omega_x\omega_z \\ (I_y - I_x)\omega_y\omega_x \end{bmatrix} + \begin{bmatrix} h_z\omega_y - h_y\omega_z \\ h_x\omega_z - h_z\omega_x \\ h_y\omega_x - h_x\omega_y \end{bmatrix} = -J_w\dot{\Omega}_w, \tag{3}$$

where

$$I = diag(I_x, I_y, I_z), \quad J_w = diag(J_x, J_y, J_z),$$

$$J_w\Omega_w = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} J_x\Omega_x \\ J_y\Omega_y \\ J_z\Omega_z \end{bmatrix}, \quad \omega_{b/N} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & sin(\theta) \\ 0 & cos(\phi) & cos(\theta)sin(\phi) \\ 0 & -sin(\phi) & cos(\theta)cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} - \omega_0 \end{bmatrix},$$

$$\dot{\omega}_{b/N} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} 0 & cos(\theta)\dot{\theta} \\ -sin(\phi)\dot{\phi} & -sin(\theta)sin(\phi)\dot{\theta} + cos(\theta)cos(\phi)\dot{\phi} \\ -cos(\phi)\dot{\phi} & -sin(\theta)cos(\phi)\dot{\theta} - cos(\theta)sin(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} - \omega_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & sin(\theta) \\ 0 & cos(\phi) & cos(\theta)sin(\phi) \\ 0 & -sin(\phi) & cos(\theta)cos(\phi) \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix},$$

where ω_0 is the angular speed of the Earth.

The system can be linearized using small angle approximation,

$$\begin{aligned} I_x\ddot{\phi} - \omega_0\dot{\theta}(I_x + I_z - I_y) + h_y\omega_0 &= -\dot{h}_x \\ I_y\ddot{\theta} - \omega_0\dot{\phi}(I_y + I_x - I_z) + h_x\omega_0 &= -\dot{h}_y \\ I_z\ddot{\psi} &= -\dot{h}_z \end{aligned} \tag{4}$$

If all states can be measured, it is very easy to decouple the system by cancelling the nonlinear terms, and then apply the simple PD controller on the reference errors.

$$\begin{aligned} \dot{h}_x &= \omega_0\dot{\theta}(I_x + I_z - I_y) - h_y\omega_0 + K_{px}e_\phi + K_{dx}\dot{e}_\phi \\ \dot{h}_y &= \omega_0\dot{\phi}(I_y + I_x - I_z) - h_x\omega_0 + K_{py}e_\theta + K_{dy}\dot{e}_\theta \\ \dot{h}_z &= K_{pz}e_\psi + K_{dz}\dot{e}_\psi \end{aligned} \tag{5}$$

If the target is moving slowly, then the reference signal is approximately constant. The system will be brought back to the ideal second order system.

$$\begin{aligned} I_x\ddot{e}_\phi + K_{dx}\dot{e}_\phi + K_{px}e_\phi &= 0 \\ I_y\ddot{e}_\theta + K_{dy}\dot{e}_\theta + K_{py}e_\theta &= 0 \\ I_z\ddot{e}_\psi + K_{dz}\dot{e}_\psi + K_{pz}e_\psi &= 0 \end{aligned} \tag{6}$$

But this result is built on all the above assumptions and given the absolute information. A new controller needs to be designed that can work on some region as large as possible without the absolute information.

B. Modified PD

Without the absolute attitude information, there is no way to truly decouple the nonlinear attitude dynamics. Thus, the system is linearized around the desired state, where the target is right in front of the CubeSat, under small angle and circular orbit assumption. This naturally leads to a decoupled three-channel PD controller for the pitch, yaw, and roll attitude, which only uses relative information. To be specific, the pitch and yaw channels are for target tracking, and the roll channel is for optimal solar angle.

However, there is a huge problem if this controller is implemented directly. As in the image plane, there is no difference between a rolling CubeSat with the target fixed in front of it with some angular offset and a fixed CubeSat with the target circling (around the central FoV) in front of it. In both situations, the target will appear circling around the FoV center in the image plane. Thus, the dynamics is still coupled for the linear model, even though the controller is already decoupled. Unless the target is already locked in the FoV center, rolling the CubeSat will also change the target position in the image plane, which makes the channels fighting against each other.

One natural way to solve this problem is to separate the two tasks. The CubeSat will track the target first and only roll to the optimal solar angle when the target is within some small ϵ circle at the FoV center. Note that this order cannot be changed, i.e. the optimal solar angle cannot be achieved before the target is locked. Otherwise, tracking the target will also change the sun vector, which results in a different optimal solar angle. A soft rolling activation can be implemented by adding a gaussian radial kernel to the torque command.

$$\phi(r) = e^{-r^2/\sigma^2}, \quad r^2 = \theta_x^2 + \theta_y^2 \tag{7}$$

Where σ is used to controller the activation tolerance.

IV. Camera

The camera is the only hardware in the loop that is tested, which is used to measure the position of the detected target in the image plane. Any camera that can detect and locate the target in real time will suffice the work. There are several popular object detection algorithms in computer vision. For object with complicated features, such as cars and aircrafts, usually a convolutional neural net (CNN) is used to do feature selection, object detection, and return the bounding box for that object. Although CNN can have very good performance, it is relatively computationally expensive due to the deep CNN and not suitable for real time detection. If the object has very distinct features which can be used as a reference, then a feature extractor can match these features and use RANdom SAmple Consensus (RANSAC) to estimate the location of the object. An even faster method that is used in real time is Viola-Jones object detection, which trains several very simple filters that only require a few addition and subtraction to output the confidence level of target object.

All the above algorithms focus on geometric features, the shapes and configurations. The specific COTS camera that is used in this study, on the other hand, uses hues of the object as features. Since the target in this mission is very likely to have a distinct surface color, possibly green or orange, the object detection task becomes hue detection instead, which is much easier and faster to deal with. The hue that is desired to be detected have to be taught beforehand. The normal RGB (red, green, blue) representation of a color image can be converted to HSL (hue, saturation, lightness). It can be seen from fig. 4[§] that when the lighting condition changes intensively, all hues look just bright or dark. Therefore, when the hue is first taught, the lighting condition should match the true space lighting as much as possible. Otherwise, there might be bias and/or bigger error for hue determination.

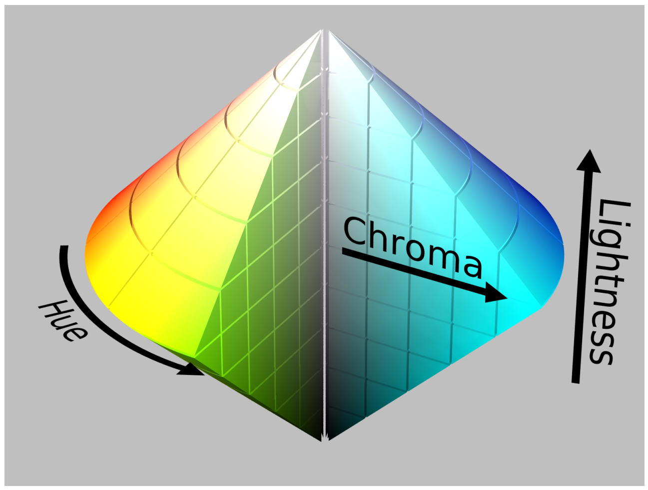


Fig. 4 The HCL cone.

Depending on the true lighting conditions in space, the accuracy of the detection can vary a lot. In order to make the signal smooth in presence of measurement noise and occasional loss of target, the data need to be averaged over a short period of time. In the worst case, potential glares on the target surface or running into eclipse can make the target undetectable. Optical filters can be added on the camera when it is too bright to remove certain wavelength or just dim the whole FoV. Some active illumination equipment can also be used during eclipse. On the other hand, since the target is a 1-meter tetrahedron, most of the time it looks like a triangle in the image plane. This feature can be used to build a triangle detector. But this is out of the scope of this study.

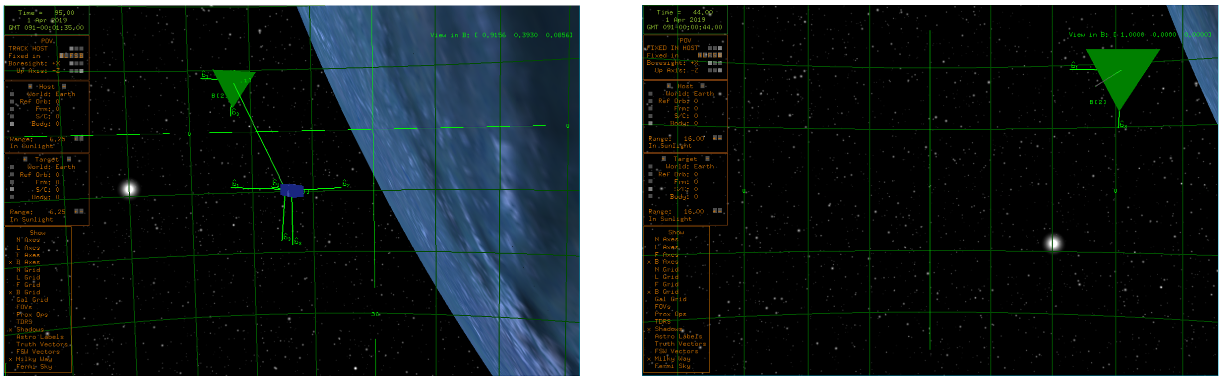
V. HWIL setting

A. 42

The NASA software 42, as described on its website, is a "high-fidelity simulation of attitude and trajectory dynamics and control of multiple spacecraft composed of multiple rigid or flexible bodies". It is used as the propagator that can perform high-fidelity simulation in the HWIL. There are plenty of sensors and actuators that can be simulated in the

§`https://commons.wikimedia.org/w/index.php?curid=9802536` software, such as sun sensors and reaction wheels. It is also capable of simulating multibody dynamics, such as a tethered CubeSat/target system. This powerful software is very suitable for this study and saves a lot of work.

In the HWIL, it is used to visualize and project the frontal FoV of the main spacecraft, which is the CubeSat, making a simulated space environment for the camera, as shown in Fig. 5. In order to test the control algorithm with a true camera, the simulation is running in real time with a 0.25s time step. Note the time step should be a little bit higher than the camera frequency, which is 60Hz, to avoid measurement noise.



(a) General view (b) Frontal FoV of the CubeSat

Fig. 5 42 simulation graphics

Apart from the visual information, the 42 software is also transmitting the simulated sensors and actuators data. At each time step, it sends a sun vector to the controller and waits for torque commands sent back. Then the software computes the true torque based on current wheel states given the wheel limits.

In the tests, it is assumed that the CubeSat will be deployed at a 400km parking LEO. The target is tethered to the CubeSat through a 10-meter rigid rod hinged on both ends. Although this is not the ideal setting for the TARGIT mission, it is quick and easy to setup and gives general ideas of how the dynamics of the actual system would be.

B. Camera calibration

Before the simulation, the camera needs to be calibrated with the projection as if it is watching the actual space on board. Since the true setup is that the 1-meter tetrahedral target is 10 meters away from the CubeSat, a simple relation can be drawn to match the geometric similarity. $\frac{1m}{10m} = \frac{y}{x}$, where y is the size of the target in the projection, and x is the correct distance between the projection and the camera. The next calibration is to align the camera FoV with the projection. This is simple but crucial for the mission. Since the controller takes as input the target position in the image plane, if the camera is tilted, the target will not be locked in front of the CubeSat.

C. Inter-process communication

For the controller on BBB to communicate with the simulation software 42, as well as all the other subsystem onboard such as the actual LiDAR system, power management, multiple sensors and reaction wheels, an inter-process communication (IPC) is built between them. Since BBB is the computer for the CubeSat, which is like the brain to the body, the server is built on it to connect with all the other clients through their sockets. In the HWIL simulation, there is only the 42 client. In real tests, the controller on BBB will transmit the information directly with the sensors and actuators, i.e. it will take the sun vector from the sun sensors, and the target position from the camera, and send the torque commands to the reaction wheels.

D. BeagleBone Black

For this HWIL simulation, there are two main functions in the controller, data preprocessing and torque command computing. Suppose the data is given as the averaged target position in the image plane, E_y, E_z, and the sun vector, $S_v = [S_{vx}, S_{vy}, S_{vz}]^T$. First the target position needs to be converted into angular offset, $\theta_z = \frac{\pi}{6}\frac{E_y}{157}, \theta_y = -\frac{\pi}{9}\frac{E_z}{103}$. Then the optimal solar angle offset θ_x are computed. Let $\hat{n} = [n_y, n_z]$ be the normal vector of the solar panel in the image frame, then $\hat{n} = n_y\hat{b}_2 + n_z\hat{b}_3$ in the body frame. Since the optimal solar angle is found while pointing the camera at the target, only the projection of the sun vector in the image frame is needed. Then

$$\begin{aligned}
\hat{n} \times Proj(S_v) &= (n_y\hat{b}_2 + n_z\hat{b}_3) \times (S_{vy}\hat{b}_2 + S_{vz}\hat{b}_3) \\
&= (n_y S_{vz} - n_z S_{vy})\hat{b}_1, \\
\hat{n} \times Proj(S_v) &= sin(\theta_x)\|Proj(S_v)\|\hat{b}_1 \\
&= sin(\theta_x)\sqrt{S_{vy}^2 + S_{vz}^2}\hat{b}_1.
\end{aligned}$$

Therefore, let $s = sin(\theta_x) = (n_y S_{vz} - n_z S_{vy})/\sqrt{S_{vy}^2 + S_{vz}^2}$. To eliminate ambiguity and keep θ_x within $[-\pi, \pi]$, let

$$\begin{aligned}
c = \hat{n} \cdot Proj(S_v) &= (n_y\hat{b}_2 + n_z\hat{b}_3) \cdot (S_{vy}\hat{b}_2 + S_{vz}\hat{b}_3) \\
c &= (n_y S_{vy} + n_z S_{vz}), \\
\hat{n} \cdot Proj(S_v) &= cos(\theta_x)\|Proj(S_v)\| \\
&= cos(\theta_x)\sqrt{S_{vy}^2 + S_{vz}^2}.
\end{aligned}$$

Then if $c \geq 0$, $\theta_x = asin(s)$. Otherwise, if $s \geq 0$, $\theta_x = 180 - asin(s)$, else $\theta_x = -180 - asin(s)$.

Now all three angular offsets are computed, the modified PD controller with soft rolling activation is as follows,

$$T_y = k_p\theta_y + k_d\dot{\theta}_y, \tag{8}$$

$$T_z = k_p\theta_z + k_d\dot{\theta}_z, \tag{9}$$

$$\begin{aligned} r^2 &= \theta_y^2 + \theta_z^2, \\ \phi(r) &= e^{-r^2/\sigma^2}, \end{aligned}$$

$$T_x = \phi(r)(k_p\theta_x + k_d\dot{\theta}_x). \tag{10}$$

Nonetheless, if the target is lost, all torques are set to be zero.

VI. Testing scenarios & Results

A. Testing scenarios

There are four major tests. The baseline test starts with the target inside the FoV with initial pitch and yaw angle and rate offsets. The advanced test starts with the target outside the FoV but will reenter after some time. The current controller has passed the first two tests. The last two tests are rather experiments. During eclipse, it is assumed that the camera will lose the target and hence no torques will be applied to the CubeSat. It is simulated by covering the camera physically during eclipse. It is desired to see how much the target will drift when the eclipse ends. The last test is not likely to happen, but still it is interesting to show the system motion under high dynamics.

B. Results

All units in the plots are in SI units. The angles are in degrees.

1. Baseline

Fig. 6-8 show the baseline response. In this test, the target is initially placed inside the camera FoV with some pitch and yaw offset. It can be seen from Fig. 6 that the pitch and yaw offsets are stabilized within 40 seconds with a 20% overshoot. The SSE is around 1-2 degrees, which is actually 1 pixel in the image plane. This corresponds to the measurement error of the camera. The roll offset comes from the sub-optimal solar angle. It can be seen that in order to decouple the pitch/yaw and roll attitudes, the rolling torque is only activated when the target is already locked within the center of FoV with a small tolerance. And its response is much slower by design so it doesn't introduce too much effect on the pitch and yaw attitude.

Moreover, a maximum torque limit of $0.6mN \cdot m$ is applied to the reaction wheels, corresponding to the actual equipment that is going to be used on board. The torque commands are usually very large compared with the wheel limit as shown in Fig. 7. With a limit three times larger than the current limit, the response would be much faster.

Nonetheless, the controller can stabilize the system with the current limit. The control signal looks very similar to bang-bang control with singular control arc connecting them while the attitude offset is small. It can also be seen that the rolling torque is zeroed to begin with when the target is not stabilized. Since it is designed to be small, the maximum torque command is way smaller than the wheel limit, which means that the CubeSat might reduce its budget by switching the rolling wheel to a less capable one.

At last, Fig. 8 shows the solar efficiency of the spacecraft. The timing matches the controller logic. The first 20 seconds is for the controller to track the target and bring it back to the center of FoV. After the target is locked at around 30 second, the rolling torque is activated and the solar efficiency is increased by 40% from 0.25 to 0.35.

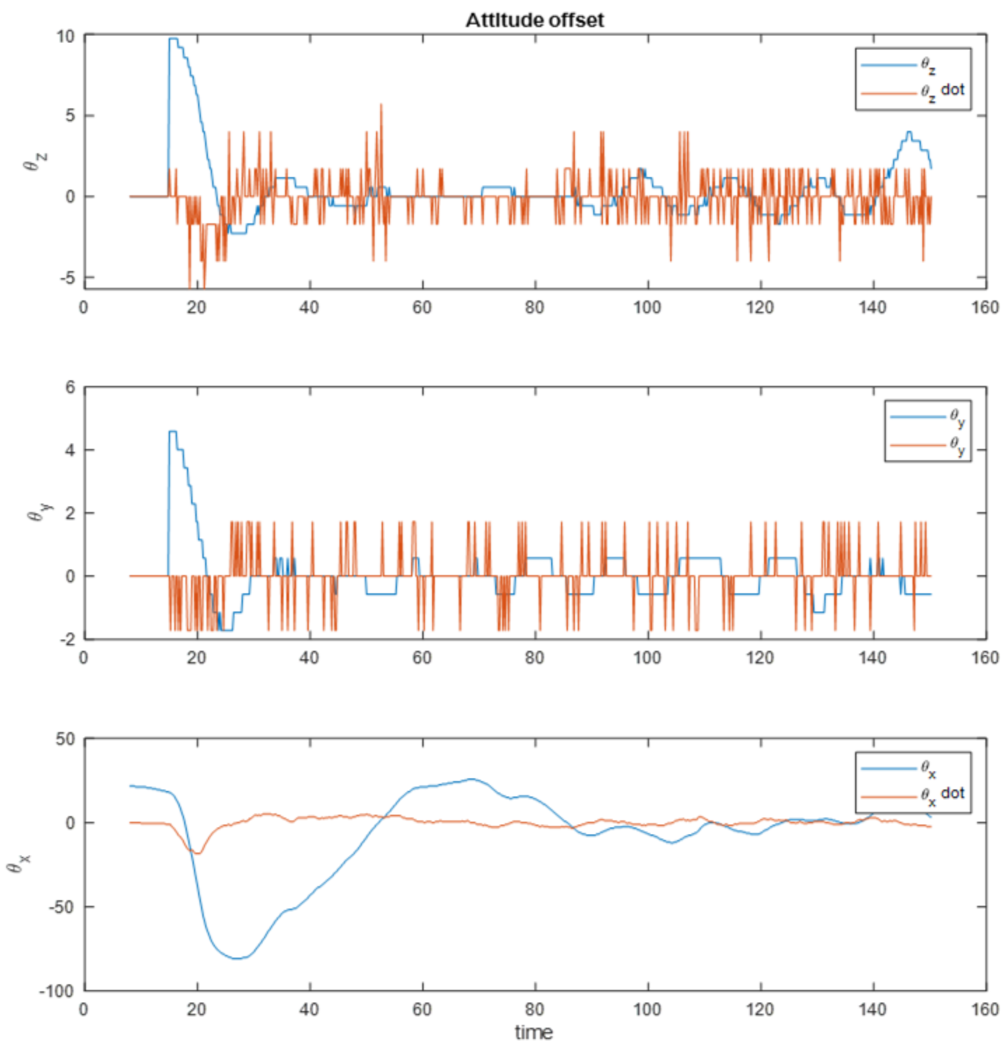


Fig. 6 Relative attitude error response.

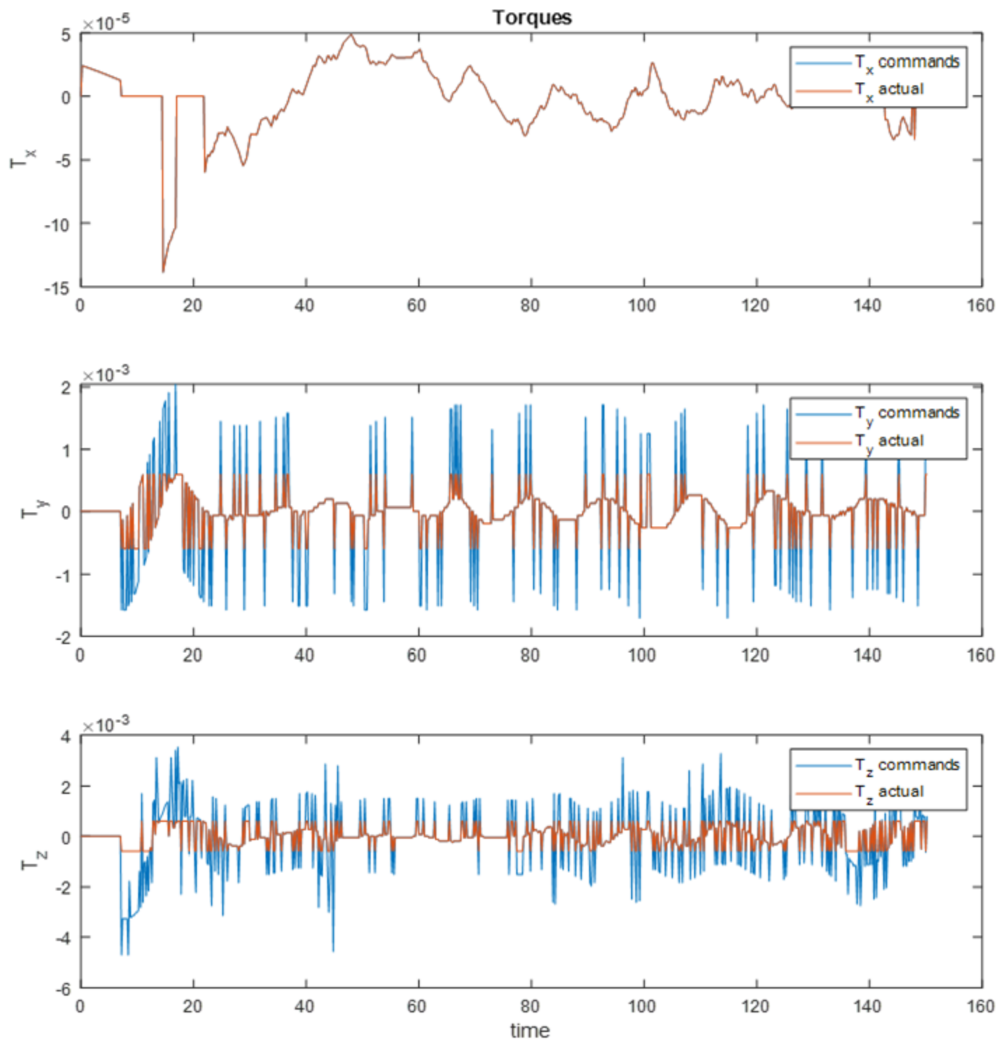


Fig. 7 Torque commands & actual torques: $T_{max} = 0.6mN \cdot m$.

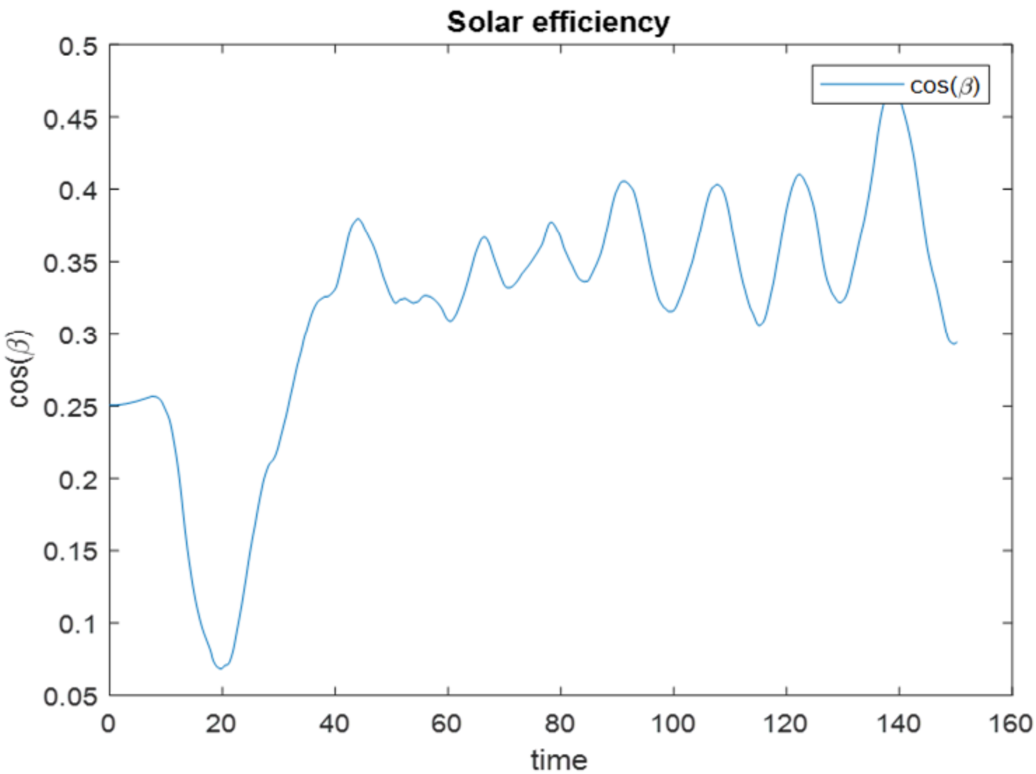


Fig. 8 Solar efficiency.

2. Reentering & High dynamics

The reentering test is the intermediate between the baseline and the high dynamics test. Whether the controller can successfully track the target when it reenter the camera FoV highly depends on reentering speed of the target. Given a higher maximum torque limit, the controller can track a faster target. The specific reentering situation can vary a lot. It can cross the center of FoV with a very high speed, or appear on the edge of the FoV and disappear again quickly. But generally speaking, the slower the target is, the longer it will stay inside the camera FoV, the higher chance it will be reacquired. The controller with the current wheel limit can handle a moving target as fast as around 20 degree/s. In other words, it needs the target to stay within the FoV for at least 3-5 second to capture it again. Any target that is moving faster than this will be considered high dynamics.

3. Eclipse drift

The Eclipse drift test turns out to be less interesting. Since the target is relatively close to the main spacecraft and tethered to it, the relative motion due to orbital dynamics is subtle compared with the attitude error that comes from the controller. From the baseline response, note that the SSE is around 2 degrees, and the target is drifting with an speed of 0.4 degree/s. Once the spacecraft enters eclipse and the target becomes too dark to see, it will keep drifting with this speed. Assuming the orbit is in eclipse for half of the orbit period, which is around 30-40 minutes, the target has sufficient time to rotate about the spacecraft for 2 rounds, which basically means an actively search for the target is needed. Otherwise, the spacecraft will have to wait for the target to reenter with a small velocity.

4. Failure mode

Apart from the dynamics condition exceeding the hardware limits, i.e. the target is fast relative to the wheel limits, resulting in either unstable system or endless oscillations, the tracking task can also be compromised if the sensors, especially the camera, are malfunctioning. This could be the lighting condition causes the target to be unrecognisable, resulting in target loss. The worst case would be the camera takes some wrong object as its target and starts to track it.

VII. Conclusions & Future work

In this study, a attitude tracking controller is built using only relative positioning information of the target. A basic testing environment with HWIL simulation is also developed to examine the performance of the controller. It passes the baseline test, but there is still a lot that can be examined in detail given the testing environment. The parameters of the controller can be fine-tuned. It can also be made adaptive to system parameter errors. But the successful tracking of the target is built on each subsystem, i.e. the sensors need to tell the controller the correct target location and sun vector, the controller should generate reasonable torque commands, and at last the actuators need to execute those commands in a desired manner.

References

[1] Schaub, H., and Junkins, J. L., *Analytical mechanics of space systems*, American Institute of Aeronautics and Astronautics, 2009.

[2] Sidi, M. J., *Spacecraft dynamics and control: a practical engineering approach*, Cambridge University Press, 2006.

[3] Nudehi, S. S., Farooq, U., Alasty, A., and Issa, J., "Satellite attitude control using three reaction wheels," *2008 American Control Conference*, 2008. doi:10.1109/acc.2008.4587262.

[4] Gui, H., Vukovich, G., and Xu, S., "Attitude tracking of a rigid spacecraft using two internal torques," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 4, 2015, pp. 2900–2913. doi:10.1109/TAES.2015.140670.

[5] Zhao, L., Yu, J., and Yu, H., "Adaptive Finite-Time Attitude Tracking Control for Spacecraft With Disturbances," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 3, 2018, pp. 1297–1305. doi:10.1109/TAES.2017.2780638.

[6] Kearney, M., and Steyn, W. H., "An attitude control system for the deployment and stabilisation of a tethered dual CubeSat mission," *2013 Africon*, 2013, pp. 1–5. doi:10.1109/AFRCON.2013.6757729.