Extending Target Tracking Capabilities through Trajectory and Momentum Setpoint Optimization

Robert D. Magner
Space Flight Laboratory, University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6
rmagner@utias-sfl.net

Faculty Advisor: Dr. Robert E. Zee
Space Flight Laboratory, University of Toronto Institute for Aerospace Studies

ABSTRACT

Terrestrial target tracking is performed by many communications and Earth observation missions to track ground-fixed targets. Previous Space Flight Laboratory (SFL) missions have demonstrated target tracking with high pointing accuracy, enabled by the use of an on-board star tracker. The allowable target tracking trajectories for these missions is constrained by the need to point the star tracker away from the Sun and Earth, which often means operating with a restricted envelope of allowable targets. An additional issue affecting target tracking missions is the stiction-induced jitter associated with reaction wheel zero speed crossings. This paper presents the development of 3TECS, a software platform with the dual purpose of performing trajectory optimization to achieve minimum energy slews under star tracker exclusion constraints, and momentum setpoint optimization for reaction wheel zero-crossing placement. It is shown that, for SFL’s DAUNTLESS-class spacecraft, the trajectory planning results in the ability to track any target visible below the horizon with greater than 98.7% star tracker availability, across a wide range of orbits. Additionally, we show that the momentum setpoint planning provides low-jitter payload operation windows of greater than 160 s in duration. This is in contrast to the existing solution that can result in zero-crossing jitter directly coinciding with key payload operations.

I. INTRODUCTION

Earth-pointing satellites facilitate society’s modern communications networks, and provide scientific data to help us better understand the makeup of our planet. This category of satellites operate payloads, be it an imager, telescope, or antenna, that are designed to be pointed towards Earth. Specifically, the objective is often to have the payload track a target that is fixed to the Earth’s surface. We refer to this sort of operation as target tracking, and it is executed by the spacecraft’s on-board Attitude Determination and Control Subsystem (ADCS). The level of pointing accuracy that is typically desired for target tracking applications often necessitates the use of one or more star trackers for performing high-accuracy attitude determination. This adds additional complexity to the design of the target tracking maneuver in that the spacecraft needs to be oriented to reduce obstruction of the star tracker’s view of the stars.

The problem of constructing large angle slew trajectories for target tracking under Sun avoidance constraints has been addressed by McInnes [1], who presented a relatively simple potential function approach. The potential function approach notably suffers from not guaranteeing convergence on to a feasible trajectory. Frazzoli et al. [2], developed a robust approach to handling slew planning under multiple pointing constraints using a Probabilistic Road Map (PRM) framework. Algorithms have been developed in the past for checking for constraint violations in maneuver commands. An algorithm was developed for the Cassini mission [3], which monitored attitude maneuver commands from the ground to check if any constraints would be violated, including instrument pointing constraints, and dynamic constraints imposed by the spacecraft actuators. This algorithm, and the majority of the work done in this area, has its applications primarily for large high-budget missions with large sets of constraints. They are, in general, focused on the problem of finding any feasible points in design spaces that are cluttered with constraints. This leaves a gap in the domain of small satellites. In general, small spacecraft operate under fewer constraints, and thus a bigger focus can be placed on maximizing performance, for example by aiming to track a wider range of targets, or by aiming to minimizing actuation cost.

One of the major factors impacting control performance during target tracking is the poor reaction wheel performance that occurs as the wheel speed passes...
through zero, owing to stiction effects. For a system using three reaction wheels, zero-crossings are a virtual guarantee during large slew target tracking. When a zero crossing occurs at a key moment, for example while an image is being captured by a payload camera, its pointing performance impact can degrade the payload output, or create higher-than-desired fixed offsets. By intelligently planning the spacecraft’s momentum state going into a target track, we can mitigate these effects.

The overarching goal of the work contained in this paper is to extend the capabilities of satellite platforms performing target tracking, thereby allowing for more ambitious mission profiles to be pursued. We will attempt to do this through two main avenues:

1) Trajectory optimization to ensure star tracker availability while tracking any target over the entire range of ground visible to the spacecraft below the Earth’s horizon
2) Momentum setpoint optimization to place zero-crossings outside of critical time periods where low-jitter pointing is desired

Although the work aims to be generalized for any target tracking mission, we will refer to a “sample mission” for some analyses in this paper. This hypothetical mission uses an SFL DAUNTLESS-class bus platform [4] and operates in a 1000 km sun-synchronous orbit, at one of two target LTANs, 12:00 and 15:00. The aim for the sample DAUNTLESS-class mission is to track any target that is visible up to 5° below horizon (or about 55° of nadir at 1000 km)

II. BACKGROUND

Reference Frames

Spacecraft Body Frame: The body frame, $\mathcal{F}_b$, is fixed to the spacecraft, with the origin at the center of mass, and the axes pointing normal to the geometric spacecraft faces, which are not assumed to be aligned with the body’s principle axes.

Local Vertical, Local Horizontal Orbit Frame: In this work, spacecraft attitudes are often defined with respect to an orbiting frame, $\mathcal{F}_o$. By convention, we use the Local Vertical, Local Horizontal (LVLH) realization of the orbit frame. The LVLH frame is centered at the spacecraft center of mass and has its Z-axis pointing along the orbit angular momentum vector, the X-axis pointing along the position vector with respect to the center of Earth, and the Y-axis forming the right-handed triad.

Target Tracking Optimization Base Frames: In this paper, an algorithm for optimizing target tracking trajectories will be presented. This algorithm will make use of four “base frames” for defining the attitude trajectories.

The frames all have their Z-axis pointing from the spacecraft body towards the ground target, and their X-axis pointing towards the cross product of the body to ground target vector and some other vector. For example, the X base frame, $\mathcal{F}_{XB}$, has its X-axis pointing towards the cross product of the body to ground target vector and the LVLH X-axis. The other three frames, $\mathcal{F}_{YN}$, $\mathcal{F}_{ZB}$, and $\mathcal{F}_{OB}$, are analogous, but using the LVLH Y-axis, LVLH Z-axis, and Sun Vector in the cross product, respectively.

Attitude Dynamics

The motion of a rigid body is governed by Euler’s equation, which for the case of a spacecraft with three orthogonal wheels, is given as follows, expressed in the body frame $\mathcal{F}_b$ [5]:

$$\mathbf{I} \ddot{\omega} + \omega \times (\mathbf{I} \omega + \mathbf{h}_w) = \mathbf{\tau}_c + \mathbf{\tau}_d$$

(1)

Where $\mathbf{I}$ is the body’s moment of inertia about its center of mass, $\omega$ is the angular velocity, $\mathbf{h}_w$ is the moment of the wheels, and $\mathbf{\tau}_c$, $\mathbf{\tau}_d$ are the control and external disturbance torque, respectively. The kinematics can be described with quaternions as follows:

$$\dot{\mathbf{q}}_{bi} = \frac{1}{2} \left[ \eta \omega - \omega \times \epsilon \right]$$

(2)

Where $\mathbf{q}_{bi} = [\epsilon^T \ \eta]^T$ is the inertial-to-body quaternion, and the Euler’s parameters $\epsilon = a \sin(\frac{\phi}{2})$, $\eta = \cos(\frac{\phi}{2})$ describe the Eigen-axis rotation of angle $\phi$ about axis $a$ [6].

Align-Constrain Formulation

In general we define a target tracking attitude by aligning one body vector with a reference vector, and constraining a second body vector towards a second
reference vector. The formulation for this is as follows: first we define a rotation to align the body align vector $p_B$, towards the desired target vector $p_D$. For this, we define both of these vectors in the body frame, at the body’s initial attitude, $F_b,1$. The Euler axis $a_1$ and angle $\theta$ of this rotation are given as:

$$a_1 = \frac{p_{B,1} \times p_{D,1}}{\|p_{B,1} \times p_{D,1}\|}, \quad \theta = \cos^{-1}\left(\frac{p_{B,1} \cdot p_{D,1}}{\|p_{B,1}\| \|p_{D,1}\|}\right)$$

(3)

The rotation from the body’s initial attitude $F_{b,1}$ to the intermediate align frame $F_{b,2}$ is then found using the standard formulation of a rotation matrix based on Euler axis and angle:

$$C_{21} = \cos \theta I + (1 - \cos \theta)a_1a_1^T + \sin \theta a_1 \times$$

(4)

The next step is determining a rotation from the intermediate align frame to the final desired body frame, $F_{b,\text{des}}$, which we know must be a rotation about the alignment axis, $p_D$. The Euler angle of the rotation, $\phi$, is determined by minimizing the separation between the body constraint vector $c_B$, and the vector towards which it is constrained, $c_D$. This can be achieved by projecting $c_B$ and $c_D$ onto the plane to which $p_D$ is normal. Those projections, expressed in the intermediate frame are:

$$c_{\text{proj},B,2} = \frac{p_{D,2} \times c_{B,2} \times p_{D,2}}{\|c_{B,2} \times p_{D,2}\|^2},$$

(5)

$$c_{\text{proj},D,2} = \frac{p_{D,2} \times c_{D,2} \times p_{D,2}}{\|c_{D,2} \times p_{D,2}\|^2}$$

(6)

Being both on the plane of rotation, these projected vectors can be aligned, which will result in a closest constraint of $c_B$ towards $c_D$. The angle $\phi$ is then given by:

$$\phi = \cos^{-1}\left(\frac{c_{\text{proj},B,2} \cdot c_{\text{proj},D,2}}{\|c_{\text{proj},B,2}\| \|c_{\text{proj},D,2}\|}\right)$$

(7)

We then find the corresponding rotation matrix:

$$C_{\text{des},2} = \cos \phi I + (1 - \cos \phi)c_{\text{proj},B,2}c_{\text{proj},B,2}^T + \sin \phi c_{\text{proj},D,2}$$

(8)

And finally, the initial body to desired body frame rotation is the two rotations discussed above, applied sequentially:

$$C_{\text{des},1} = C_{\text{des},2}C_{21}$$

(9)

**Star Tracker Obscuration**

One of the main constraints on target tracking trajectories for many missions is ensuring that the star tracker is unobscured. In order to register a valid inertial frame-to-instrument quaternion, the star tracker must have a view of the stars, which in general means it’s pointed sufficiently far away from the Earth’s horizon and the Sun. A star tracker has angles $\nu_\oplus$ and $\nu_\odot$ defining how close the boresight can approach Earth’s horizon, or the Sun, respectively. The Sun can be treated as a point, and thus the minimum boresight-to-Sun angle is $\beta_\odot = \nu_\odot$. For the Earth, we must consider the angle from nadir to the horizon, which depends on the earth’s radius $r_\oplus$ and the spacecraft altitude $a$:

$$\beta_\oplus = \zeta_{\text{horizon}} + \nu_\oplus$$

(10)

Where:

$$\zeta_{\text{horizon}} = \sin^{-1}\left(\frac{r_\oplus}{a + r_\oplus}\right)$$

(11)

The goal is to minimize our control error by maximizing our star tracker availability. For the sample mission, we set a requirement on the pointing to have an error of less than 0.55°, (2σ) about all three axes. From simulating the full spacecraft attitude determination and control in target tracking, with and without availability of the star tracker, we can determine its control error for each case. Based on those values, we can convert our original requirement on pointing error into a somewhat more easily analyzed requirement on star tracker availability. What we find is that if we have an available (i.e., unobscured) star tracker at minimum 95% of the time during target tracking, then we will be able to hit a less than 0.55°(2σ) overall pointing error. Using this threshold on star tracker availability, we can analyze different target tracking attitude trajectories to find one that will be able to meet our original requirement.

**Spacecraft Actuator Constraints**

Another main constraint to consider in constructing attitude trajectories is the dynamic limitations of the actuators. It is necessary to ensure that the commanded
rates and accelerations are within the performance capabilities of the reactions wheels. We can use a momentum balance between the bus and the wheels, about the bus’s major principle axis, to determine a limit on the allowable slew rate $\omega_{b,\text{max}}$ based on the speed limit of the wheel $\omega_{w,\text{max}}$, the moment of inertia of the wheel about its spin axis $I_w$, and the major principle body inertia $I_{p,1}$:

$$\omega_{b,\text{max}} = \frac{I_w\omega_{w,\text{max}}}{I_{p,1}}$$  \hspace{1cm} (12)

For the maximum allowable acceleration $\omega_{b,\text{max}}$, we use the maximum torque at a limiting speed $\tau_{w,\text{max}}$:

$$\omega_{b,\text{max}} = \frac{\tau_{w,\text{max}}}{I_{p,1}}$$  \hspace{1cm} (13)

**Constraint Vector Study**

As a first step in tackling the problem of constructing suitable target tracking trajectories, strategies used in past missions were examined. All SFL missions performing target tracking use an align-constrain approach, wherein one body vector is aligned with a ground target, and another body vector is constrained towards a constraint vector typically fixed in the inertial or orbit frame.

Using Systems Tool Kit (STK), simulations of the spacecraft operations over a full year were constructed. First, models of the spacecraft orbital mechanics were built based on the two target orbits, both sun-synchronous 1000 km orbits, with LTAN of 12:00 and 15:00, respectively. Target tracking attitude configurations were built in using an align and constrain approach wherein the antenna boresight (along the body Z-axis) is aligned to the ground target, and the star tracker boresight (along the body negative Y-axis) was constrained to point it away from both the Sun and Earth.

A tool was developed in MATLAB to interface with STK in order to extract data to determine at what points over the course of a year does the star tracker become obscured by either the Sun or Earth during target tracking of the ground station at SFL, as it appeared anywhere below the horizon. The tool further factors in slew times when transitioning from coarse to fine pointing modes when the star tracker becomes available, ultimately giving data on when the spacecraft is in fine pointing mode. Simulations were run with the star tracker constrained towards a number of different possible orbit frame and inertial frame constraint vectors. Table I below shows the star tracker availability over 1 year during target tracking periods, with the star tracker body vector constrained towards some standard constraint vector options. This table shows only a few of the many constructions that were studied, which included many variations of both the body vector being constrained, and the non-body vector to which it is constrained. None of the other variations studied, however, gave substantially better results than those shown here, and none of the strategies were able to meet the availability threshold of 95% discussed in section 4.

For the 12:00 LTAN orbit, there was one strategy that worked well. This strategy involved constraining the star tracker boresight vector to either positive or negative orbit normal depending on whether the ground station is in the positive or negative orbit normal direction from the orbital plane. For example, if the spacecraft is passing from south to north, and is to the west of the target, then we constrain the star tracker to negative orbit normal. The rationale being that this keeps the star tracker from being obscured by Earth in all cases, and for the 12:00 orbit Sun obscuration is not a major issue. This strategy resulted in 97% availability for the 12:00 orbit, meeting the threshold. Unfortunately, for the 15:00 orbit, the availability was much lower at 67% owing mainly to Sun obscuration.

It became increasingly clear that the relatively simple approaches that had worked for past missions were not going to be sufficient for the sample mission. Previous target tracking missions only had to track a narrow range of targets, for example by limiting passes to those where targets passed within 5 degrees of nadir. By contrast, the aim for the sample DAUNTLESS-class mission is to track any target that is up to 55° of nadir. The result of this is that when we consider the entire year, the relative locations of the Earth, the Sun, the spacecraft, and its target can vary hugely. The large variation in relative geometries makes it difficult to select constraint vectors fixed in a conventional frame that will consistently and predictably result in star tracker availability, while also ensuring kinematically feasible trajectories. The solution from here was to turn to numerical approaches. Specifically, the author chose to employ the use of numerical optimization to process the geometries for each individual pass, and compute a constraint vector that ensured star tracker availability and kinematic feasibility. The details of this approach we will presented later in this paper.

### TABLE I: Star tracker availability during target tracking for different star tracker constraint vectors

<table>
<thead>
<tr>
<th>Constraint Vector</th>
<th>% Star tracker availability during TT over 1 year</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00 LTAN Orbit</td>
<td>15:00 LTAN Orbit</td>
</tr>
<tr>
<td>Anti-Sun</td>
<td>82</td>
</tr>
<tr>
<td>Anti-nadir</td>
<td>57</td>
</tr>
<tr>
<td>Along velocity</td>
<td>62</td>
</tr>
<tr>
<td>Anti-velocity</td>
<td>59</td>
</tr>
<tr>
<td>Positive orbit normal</td>
<td>48</td>
</tr>
<tr>
<td>Negative orbit normal</td>
<td>59</td>
</tr>
</tbody>
</table>

---

Magner 4 32nd Annual AIAA/USU Conference on Small Satellites
Fig. 3: Reaction wheel ground test demonstrating noise spike at zero speed crossing

Momentum Setpoint Planning

In the context of reaction wheel performance, a “zero-crossing” refers to when a reaction wheel’s momentum in a direction aligned with its spin axis, crosses from a positive value to a negative one, or vice versa. This region in the wheel’s speed profile is particularly notable as the wheel experiences jitter, degrading the overall control performance. This jitter can be explained primarily by friction and stiction effects close to the zero speed point, resulting in nonlinear torque profiles that the wheel’s controller cannot accurately regulate. At higher speeds, stiction effects are not present, and a much smoother profile can be obtained. Figure 3 shows results from DAUNTLESS reaction wheel testing. In this test, a wheel torque command of 4 mNm was sent, and beginning from −1500 rad/s, the wheel is accelerated to 1500 rad/s. In general, we can see the wheel’s output torque stays within +/− 0.5 mNm of the setpoint. However, at the zero-crossing, a dramatic increase in torque noise amplitude is visible, fluctuating close to 4 mNm from setpoint.

It would of course be desirable during pointing operations to set the wheels at an elevated speed and avoid crossing over zero altogether. For missions performing inertial pointing, or missions using 4 or more reaction wheels, this can be possible. Unfortunately, for missions performing large angle slew maneuvers, for example in nadir pointing or target tracking, and using a basic set of three orthogonal wheels, zero-crossings become an inevitability. This is because the spacecraft’s net momentum vector is constant in inertial space (with reaction wheel control only, and disturbances assumed to be small in the short-term), and thus as the spacecraft slewed, the momentum is redistributed among the wheels. As soon as we go beyond 90° of slewing, we’ve guaranteed that at least one wheel will cross zero. With no consideration of momentum planning, it’s likely that zero crossings will arise very frequently during target tracking slew. There is therefore a clear motivation for deriving a technique to plan the spacecraft’s momentum state intelligently to try to mitigate the effects of zero-crossings. As stated, in general it is impossible to avoid zero-crossings entirely, but we can attempt to “place” the crossings at points in the pass where their impact will be reduced. For example, we often are most concerned with accuracy around the middle of our pass, when we are closest to the target, so it would be preferable to place any zero-crossings towards the beginning or end of the pass, when we are far from the target and the quality of the image or communications link would already be low due to atmospheric effects.

In planning the reaction wheel momentum states for zero-crossing avoidance, we must consider what states are reachable. First, our momentum is limited by the saturation speeds of the wheels, or the maximum speed we want to allow the wheels to operate at. It is also limited by our ability to change our momentum during our momentum management phase from the end of one target tracking phase to the beginning of the next using the secondary actuation source, the on-board magnetorquers. This limit is based on the initial momentum state \( \mathbf{h}_{w,0} \) and the total momentum management capacity of the magnetorquers. The momentum management control scheme is based on Lovera [7] and takes the following form (expressed in \( \mathcal{F}_b \)):

\[
\begin{align*}
\mathbf{m}_m &= -\frac{K_m (b \times \Delta \mathbf{h}_w)}{||b||} \quad (14) \\
\tau_m &= m_m^s b \quad (15) \\
\end{align*}
\]

Where \( \mathbf{m}_m \) is the control dipole that is induced by the magnetorquers. The computation of \( \mathbf{m}_m \) is based on a modified proportional controller where the error term is the difference between the momentum of each wheel and the setpoint wheel momentum \( \Delta \mathbf{h}_w = \mathbf{h}_w - \mathbf{h}_{w,sp} \). We furthermore ensure that \( \mathbf{m}_m \) points right-hand orthogonally to \( \mathbf{h}_w \) and the local magnetic field \( \mathbf{b} \), such that the resulting magnetic torque, given by equation 15, is then pointed in a direction pointed as far as possible from \( \Delta \mathbf{h}_w \). Thus, over time the momentum management will tend to bring the wheels to their desired setpoint. In determining the setpoint that we will pick for a pass beginning at \( t_1 \), we constrain ourselves to those that are reachable from our current state at \( t_0 \), based on effecting the momentum management controller in the interval \( t_0 < t < t_1 \) with the propagated local magnetic field.

III. SOFTWARE OVERVIEW

This section describes 3TECS (Target Tracking Trajectories for Exclusion Constrained Spacecraft). As the name suggests, 3TECS is a software platform that computes optimized target tracking trajectories under exclusion constraints. In addition, it has the functionality
to optimize reaction wheel setpoints to avoid zero-crossings during target tracks. The high-level architecture of 3TECS is shown in block diagram form in Figure 4. The software is implemented in MATLAB, with Systems Tool Kit (STK) used for orbit propagation and orbit-frame ephemeris computation, as well as in the verification phase.

The operator inputs consist of i) the spacecraft’s up-to-date NORAD two-line element set (TLE) to specify its orbit, and ii) the target tracking intervals, expressed as a start and end time. The 3TECS internal configuration parameters consist of things like the spacecraft mass properties, the actuator dynamic properties, the payload and exclusion (star tracker) vectors, the attitude control flight code command flags, and the latitude and longitude of the ground target. STK’s Simplified General Perturbations (SGP4) propagator is used to propagate the orbit based on the input TLE. With the orbital position propagated over the period of interest, the relevant ephemeris vectors, as seen in the spacecraft’s orbit frame, can be computed. These vectors include the spacecraft body-to-ground vector (which defines how the payload must be pointed), as well as the nadir and body-to-Sun vector (which are both used to define the exclusion regions). These vectors, along with the configuration parameters, are passed to the constrained trajectory optimization, where trajectory planning is performed. The full description of the trajectory planner and its internal algorithm are given in section IV.

The planned trajectories are passed through to the spacecraft in the form of a two parameter set the base frame and the \( \theta_{optimal} \) angle (the significance of these parameters, and how they are used to specify a trajectory will be explained in section IV). Verification can also be performed, and the dynamics can be passed to the wheel momentum setpoint optimization, which will be discussed in section V.

With feasibility confirmed from the STK verification, all of the computed values to specify the target tracking trajectories and momentum states are passed to the time-tagged ADCS command generator. Here, the values are used to create commands in a format that can be uploaded and read by the spacecraft’s on-board computer. On the flight code side, new functionality has been developed to allow the ADCS to interpret and execute the 3TECS trajectories. Specifically, the new code interprets the base frame and the \( \theta_{optimal} \) parameters and constructs the trajectory using the formulation that will be presented in section IV.

IV. Trajectory Planner

As discussed in the software overview (section III), the planner takes the propagated orbit frame ephemeris vectors, as well as the set of configuration parameters, and computes the attitude trajectory for each target tracking pass. This section discusses the internal design of the 3TECS trajectory planner, which are used to compute the optimal attitude trajectories.

Trajectory Optimization Problem Formulation

For each target tracking pass, a constrained optimization problem is defined. The optimization aims to construct a quaternion which rotates the spacecraft body with respect to one of the base frames such that the star tracker points as far away as possible from the two obscuration cones formed by the Sun and Earth. We define our body Z-axis as pointing along the payload vector, and thus during target tracking is restricted to pointing towards the ground target (i.e. always being aligned with the Z-axis of our base frames). The problem can therefore be reduced to only one variable which is the roll angle about the Z-axis. The quaternion that rotates from the base frame to the optimal frame is defined below, where \( e(t) \) is the vector pointing from the body to the ground target, expressed in the base frame, and \( \theta_{optimal} \) is the optimal roll angle about that vector.
\( q_{\text{base-optimal}}(t) = \left[ e(t) \sin\left(\frac{\theta_{\text{optimal}}}{2}\right) \cos\left(\frac{\theta_{\text{optimal}}}{2}\right) \right]^T \)

The objective of the optimization is to determine the \( \theta_{\text{optimal}} \) that will result in the minimum work trajectory. The trajectory work is quantified using the work done in rotating the spacecraft about its Z-axis, and as explained above. The Z-axis work over a target track spanning from \( t_1 \) to \( t_2 \) is given by the following equation:

\[
W_z = \int_{t_1}^{t_2} I_{zz} \dot{\gamma} \dot{\gamma} dt = \int_{t_1}^{t_2} I_{zz} \ddot{\gamma} \dot{\gamma} dt \tag{18}
\]

Where \( \gamma \) is the rotation about the body z-axis as seen in an inertial frame, \( \tau_z \) is the applied torque about the body z-axis, and \( I_{zz} \) is the body’s z-axis inertia. The system is subject to the constraint that the star tracker vector cannot enter the Sun or Earth obscuration cones at any point during the pass. Using an obscuration cone of half-angle \( \beta_\odot \) around the Sun vector, and an obscuration cone of \( \beta_B \) around the nadir vector, we define the angles to obscuration \( \phi_\odot(t) \) and \( \phi_B(t) \), where \( r(t) \) is the star tracker boresight vector, \( x_\odot(t) \) is the Sun vector, \( x_B \) is the nadir vector, and \( \sigma_\odot, \sigma_B \) are extra angular safety margins added to the obscuration cones. The full constrained optimization problem is summarized in equation 19.

\[
\min_{\theta} J(\theta) = W_z = \int_{t_1}^{t_2} I_{zz} \ddot{\gamma} \dot{\gamma} dt
\]

s.t.

\[
\begin{align*}
g_1(\theta) &= \phi_\odot(t) = \cos^{-1}\left( \frac{r(t) \cdot x_\odot(t)}{\|r(t)\| \|x_\odot(t)\|} \right) \\
-\beta_\odot + \sigma_\odot &\leq 0, \quad \forall \ t_1 \leq t \leq t_2 \\
g_2(\theta) &= \phi_B(t) = \cos^{-1}\left( \frac{r(t) \cdot x_B(t)}{\|r(t)\| \|x_B(t)\|} \right) \\
-\beta_B + \sigma_B &\leq 0, \quad \forall \ t_1 \leq t \leq t_2
\end{align*} \tag{19}
\]

We solve this problem using an SQP scheme [8], which is well suited to this sort of smooth nonlinear constrained optimization problem. Solving this problem, we find \( \theta_{\text{optimal}} \), which defines the optimal attitude in the base frame, and knowing the trajectory of the given base frame, we can define the full optimized inertial attitude trajectory.

**Base Frame Selection Algorithm**

Finding an acceptable solution to the optimization problem described in the preceding subsection is dependent on the selection of the base frame within which the optimization is performed. For virtually any selection of base frame, there are cases where the optimization will not converge on a solution that meets the constraints, or will converge on a solution that is kinematically infeasible for the spacecraft to perform. For example, using the \( F_{XB} \) frame, there are a set of cases where the spacecraft passes overhead of the ground target, and thus the cross product of the body to ground target vector and the LVLH x-axis approaches a singularity. Even if a singularity is not reached, this sort of configuration can result in slews that are too fast for the reaction wheels and also potentially result in obscuration for some period of time.

After some experimentation, an approach was arrived at where four different base frames would be used (see section 3). By using these four frames, were able to find acceptable solutions for almost all of the potential cases. The full algorithm consists of cycling down through the different base frame options until an acceptable solution is found for the given target tracking pass. The optimization is run for a given base frame, and the outputs reveal whether a solution exists which satisfies the obscuration constraints. If a solution is found by the optimization, it is then used to evaluate an attitude trajectory for the given pass, which is then determined to be kinematically feasible or not based on the spacecraft actuator constraints. If a solution exists and is kinematically feasible, then we can store this solution and proceed to the next pass. Otherwise if no solution is found, or the solution found is kinematically infeasible, then we proceed to the next base frame. We continue on this way until a kinematically feasible solution is found. If no such solution is found by the fourth frame, then we proceed with an imperfect (partially obscured) pass as long as it is kinematically feasible. If this pass is not kinematically feasible, then we discard the target track for the pass, and proceed to the next one.

The base frame sequence that the algorithm cycles down through goes as follows: \( F_{XB}, F_{YB}, F_{XB}, \) and finally, \( F_{GB} \). The rational for finishing with the Sun base frame is that this frame often results in kinematically feasible solutions even if those solutions are not always unobscured. By finishing with this frame we ensure that we hardly ever have to discard the pass, and can instead find a partially obscured but kinematically feasible pass. The order of the other frames is largely arbitrary, although the \( F_{XB} \) frame does tend to result in more solutions than the \( F_{YB} \), which results in more solutions than the \( F_{XB} \), and thus this order tends to result in the lowest computational cost.

**Star Tracker Availability Results**

Once the 3TECS trajectory planning scheme described in this section was implemented, simulations were run to determine the star tracker availability that
would be achieved. Initially, simulations were run for two 1000 km Sun-synchronous orbits, which had LTANs of 12:00 and 15:00 respectively, as these were the target orbits for the DAUNTLESS sample mission. Although star tracker availability is calculated internally in 3TECs, for the purpose of validation, external simulation was done by exporting the trajectories generated by 3TECS into STK, and using that platform to compute exclusion and availability based on its own propagation of orbit and ephemeris. The results indicated that the 3TECS trajectories will allow the spacecraft to achieve greater than 99.9% star tracker availability for both orbits. As discussed in section 4, star tracker availability of 95% or more should allow us to meet our ADCS pointing requirement, and we are therefore well in compliance. Note that this does not fully verify the original pointing requirement; this verification will be presented in section VI. Figure 5 shows the star tracker exclusion plots for the 15:00 LTAN orbit. The plots are obtained by mapping the Sun and nadir vectors into the spacecraft body frame over the course of all target tracking operations over the course of an entire year. In the case of the Earth, we additionally expand an exclusion area around the nadir vector at each time step of half-angle \( \zeta \), which is the size of the Earth visible to the spacecraft, as defined in section 4.

As we can see in these plots, there are substantial “windows” around the star tracker vector, whose size are sufficiently large to ensure little-to-no obscuration based on the boresight-to-Sun and boresight-to-Earth horizon angles defined in section 4. The results further indicate that 100% of the trajectories are kinematically feasible, and for the 15:00 LTAN orbit, 99.1% of the passes are completely unobscured (99.2% for 12:00 LTAN). For the other < 1% of passes, the spacecraft is unobscured for on average 89% of the pass. The passes with < 100% availability all occur in the three months surrounding winter solstice during passes that occur at approximately 19:30 local time.

After examining the two primary target orbits, 3TECS was further tested with additional sample operational orbits. The orbits were all 1000 km Sun-synchronous; the vast majority of these sort of LEO target tracking missions operate in Sun-synchronous orbits, and higher altitudes are generally more demanding as the target is visible more frequently (1000 km being the highest altitude that has been used thus far for an SPL mission). The availability results for these simulations are shown in table II. The results indicate that for orbits with LTANs nearer to 06:00 or 18:00 (dawn-dusk or dusk-dawn) 3TECS has worse availability performance than for orbits with LTANs closer to noon or midnight. However, for all of the examined orbits we see high availability, easily surpassing our 95% goal. Note that we did not test many orbits in the range 12:00 < LTAN < 00:00 as the results for those should be symmetric with the results for the range 00:00 < LTAN < 12:00. This is confirmed with the testing of 18:00 LTAN, which yielded identical results to 06:00 LTAN.

### V. MOMENTUM SETPOINT PLANNER

#### Momentum Setpoint Optimization Problem Formulation

During target tracking operations, we use our reaction wheels as the sole source of control actuation. The Euler’s equation for a rigid body controlled by reaction wheels can be rearranged for the body frame wheel momentum change \( \dot{\mathbf{h}}_w \):

\[
\dot{\mathbf{h}}_w = -\mathbf{I}\dot{\omega} - \omega^\times (\mathbf{I}\omega + \mathbf{h}_w) \tag{20}
\]

Equation 20 can be integrated starting from an initial wheel momentum state \( \mathbf{h}_{w,b,sp} \), and using the known target tracking body dynamics for the given pass \( \omega \), and \( \dot{\omega} \). This wheel momentum state at the beginning of the target tracking phase is the setpoint during the preceding momentum management phase. However, we want to regulate our momentum in the inertial frame as the momentum in the body frame is subject to changing as the attitude of the body changes in the inertial frame. The body frame setpoint momentum given to the momentum management controller is thus:

\[
\mathbf{h}_{w,b,sp}(t) = \mathbf{C}_{bi}(t)\mathbf{h}_{w,i,sp} \tag{21}
\]

Where \( \mathbf{h}_{w,i,sp} \) is an inertial frame wheel momentum setpoint vector that is given to the controller at the beginning of the momentum management phase and stays

<table>
<thead>
<tr>
<th>Orbit LTAN (HH:MM)</th>
<th>Star Tracker Availability During Target Tracking Over 1 Year (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>99.3</td>
</tr>
<tr>
<td>03:00</td>
<td>99.1</td>
</tr>
<tr>
<td>06:00</td>
<td>98.7</td>
</tr>
<tr>
<td>09:00</td>
<td>99.1</td>
</tr>
<tr>
<td>12:00</td>
<td>99.3</td>
</tr>
<tr>
<td>15:00</td>
<td>98.7</td>
</tr>
<tr>
<td>18:00</td>
<td>98.7</td>
</tr>
</tbody>
</table>

**Fig. 5:** 1 year simulated star tracker exclusion regions on body celestial sphere for 3TECS-generated trajectories in 15:00 LTAN orbit
constant throughout, $C_b(t)$ is the time-varying attitude rotation matrix, allowing us to find the body frame wheel setpoint $h_{w,i,sp}(t)$ to be given to the controller at time $t$. Integrating equation 20 over the course of the target tracking pass gives us a profile of the wheel momentum over time, allowing us to identify zero crossing times. We perform this identification simply by performing a numerical search on the wheel speed data to find when the speeds become sufficiently low or change in sign. If we want to achieve good pointing performance at a time $t_p$ when, for example, an image is being captured, then our goal is to maximize $\Delta t_{pzc}$, the time from $t_p$ to the nearest zero crossing time $t_{zc}$:

$$\Delta t_{pzc} = |t_p - t_{zc}| \quad (22)$$

The system is subject to the constraint that the individual wheel momentums have to be less than an operator-defined allowable maximum $h_{w,max}$. Also, as discussed in section 7, we are constrained by the magnetorquer actuation capacity over the course of the momentum management phase. The full constrained optimization problem is summarized in equation 23. Note the time subscripts here: for the given pass, the momentum management phase begins at $t_0$ and lasts until $t_1$, at which point the target tracking phase begins, which spans from $t_1$ to $t_2$.

$$\min_{h_{w,i,sp}} \mathcal{J}(h_{w,i,sp}) = -\Delta t_{pzc}(h_{w,i,sp}) = -|t_p - t_{zc}(h_{w,i,sp})|$$

s.t. \quad \begin{align*}
&g_1(h_{w,i,sp}) = h_{w,1} - h_{w,max} \leq 0 \forall t_1 \leq t \leq t_2 \\
&g_2(h_{w,i,sp}) = h_{w,max} - h_{w,1} \leq 0 \forall t_1 \leq t \leq t_2 \\
&g_3(h_{w,i,sp}) = (h_{w,i,sp} - h_{w,i,0}) - \int_{t_0}^{t_1} \tau_{m,1} dt \leq 0
\end{align*} \quad (23)

**Design Space Study**

Before implementing an optimization algorithm to solve the problem given in equation 23, a worthwhile step was to examine the design space; in other words, to gain a better understanding of the objective function’s behaviour in relation the wheel setpoint design variables. Given that this data is four-dimensional (the objective function value vs. three setpoint components), it is impossible to visualize the entire design space at once. We instead use a technique of capturing 3D cross-sections or “slices” of the 4D design space. The slices are produced by holding one wheel speed at a constant value for example $\omega_z = 100$ rad/s, then varying the other two in a mesh, and evaluating the objective function at every point, producing an objective function surface. We repeat this, but holding the $Y$ then $Z$ wheel at the same constant speed, and varying the other two wheels. Then, we can repeat this entire process with different constant wheel speeds. The plots in figure 6 show the results for such a procedure, for a sample DAUNTLESS target tracking pass, taking slices at wheel speeds of $-100, 0, 100, \text{and } 200$ rad/s. Note that this examination omits consideration of the constraint functions, but is still useful as a first look at the design space.

From these results, we can draw certain conclusions about the design space. First, we can note that the peak objective function values when setting one wheel at a $0$ rad/s setpoint (see second row of plots in figure 6) are lower then when we set a non-zero setpoint. However, as we move from $100$ rad/s to $200$ rad/s (third to fourth row of plots), we don’t see a substantial increase in the peaks of the objective profile. It is thus worthwhile to restrict ourselves to lower wheel speeds because it keeps in a smaller range where it will be easier to perform momentum management from one setpoint to another, in compliance with the $g_2$ constraint in 23, and furthermore because lower speeds are favorable in terms of mechanical wear-and-tear of the units themselves. Next, we can note the three trough-like valleys in each slice, whose relative configurations change in each slice. These illustrate the eigenvectors of the problem, along which the wheel momentum state can be varied arbitrarily, always yielding the same zero objective function value. We can gain some insight to the physical significance of this by considering a case where only the one wheel spinning to start, and a $>90^\circ$ slew is executed. The wheel will always cross $0$ rad/s at the same moment in the slew, when the body crosses $90^\circ$, regardless of whether the wheel was initialized at $10$ rad/s or $1000$ rad/s. Finally, an important conclusion that we are able to draw from this study is the fact that the data has a number of “flat” (zero-gradient) regions, including the trough regions. A gradient-based solver like SQP, which was used for the trajectory optimization, will have trouble with this sort of design space. When it reaches a point where the gradient computation is zero, it will mistake it for an optimum. We thus select a particle swarm method [9], which is gradient free, and from testing on the design space is able to locate sufficiently high values of the objective function within the feasible space.

**VI. Simulation Results**

In this section, we will present simulations performed using SFL’s high fidelity attitude simulation platform, that were used to test and and validate the 3TECS-generated trajectories and momentum setpoints as well as the new additions the flight code.

**Pointing Performance**

As we first discussed in section 4, we want less than $0.55^\circ (2\sigma)$ pointing error for a target at any Earth-fixed
Fig. 6: Zero-crossing objective function evaluation vs. two varied wheel speeds while holding the third constant.

Fig. 7: Simulated control error and rate control error during sample target tracking pass.

Table III: Antenna vector pointing error results for different 1-day simulation cases.

<table>
<thead>
<tr>
<th>Orbit LTAN (HH:MM)</th>
<th>Simulation Date</th>
<th>Antenna Vector Pointing Error, 2σ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00</td>
<td>December 21st</td>
<td>0.14</td>
</tr>
<tr>
<td>12:00</td>
<td>June 21st</td>
<td>0.09</td>
</tr>
<tr>
<td>15:00</td>
<td>December 21st</td>
<td>0.12</td>
</tr>
<tr>
<td>15:00</td>
<td>June 21st</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Zero-Crossing Avoidance

The secondary functionality of 3TECS was momentum setpoint planning for reaction wheel zero-crossing avoidance, which was detailed in section V. Simulations position visible to the spacecraft. We use the simulation to verify the expected performance of the ADCS against this requirement.

Figure 7 shows the pointing control error and rate control error from simulation over the course of a sample target tracking pass. As we can see, the overall Euler axis angle control error generally stays below about 0.06°, and the rate control error stays below 0.005°/s about each axis. We see that the rate control about the Y-axis is worse than the other two axes, and while the pointing bias errors about each axis are similar, the random jitter of the Y-axis pointing is notably larger than about the X or Z axes. This behavior is due to the star tracker boresight pointing along the Y-axis. The estimation accuracy of the star tracker is substantially worse about its boresight than its other two axis (55 arcsec, RMS vs 5 arcsec, RMS). Despite this, all of the components of pointing error for this simulation are well below the 0.55° requirement.

To fully validate 3TECS and the overall ADCS performance, we need to examine a range of different target tracking passes. We want to capture passes with a range of different relative geometries between the spacecraft, Earth, and Sun. This will ultimately allow us to verify that 3TECS can construct trajectories that ensure star tracker availability for any target tracking geometry with the Sun at any position. To capture this, we run simulations that are at least one day in length, which means that the Earth does a complete rotation below the spacecraft’s orbit, and we get approximately 7 passes with unique geometries and kinematics between the spacecraft and the target. We furthermore run simulations at both winter and summer solstice. Given we are nominally tracking a northern hemisphere target (the SFL ground station), the solstices represent the two extremes of the Sun’s position with respect to the spacecraft during target tracking. At summer solstice, during daytime passes, the Sun will appear very “high” in the sky (i.e., towards zenith) with respect to the target tracking spacecraft. At winter solstice it will be much lower, towards the local horizon. Table III captures pointing performance results from 1-day simulations performed at both solstices, in the two target orbits. As we can see, all of the 2σ pointing errors are well below the 0.55° requirement, validating 3TECS. Although simulations were not performed in other orbits or at other times of year, we are confident that the results would be comparable, given that we have verified that high star tracker availability (well above 95%) will be achieved in all orbits (see section 3), and through the simulations in this section, we have show the high level of pointing performance that can be achieved with an available or unobscured star tracker through the simulations presented in this section.

Zero-Crossing Avoidance

The secondary functionality of 3TECS was momentum setpoint planning for reaction wheel zero-crossing avoidance, which was detailed in section V. Simulations
were run in SFL’s ADCS simulation environment for a range of different target tracking passes, where momentum setpoints outputted from 3TECS were passed to the flight code at each momentum management phase.

Simulation was performed for a full day of ADCS operations, including 11 target tracking passes. Figure 8 shows the simulated reaction wheel speeds during a sample target tracking pass, selected as a worst-case in terms of objective function value out of the passes simulated. The momentum setpoints were generated by 3TECS, based on the formulation presented in section V. Our \( t_p \) is the time of closest approach; in other words, we are trying to maximize the time between when any zero-crossings occur and the moment when the spacecraft is overhead of the target. The target tracking maneuver begins with the spacecraft slewing into a target-pointing attitude. It then performs its target tracking slew as the spacecraft passes over the target, lasting about 18 minutes. The rotations involved in the maneuver are primarily about the Y and Z body axes, demanding large speed changes in the wheels along those axes, causing zero-crossings. The X-axis wheel, on the other hand, stays above about 60 rad/s throughout the pass. With the 3TECS momentum setpoints, the zero-crossing times \( t_{zc} \) are “placed” to maximize the time from \( t_p \). The objective function value \( \Delta t_{pzc} = |t_p - t_{zc}| \) for this pass is 82 s. This means that the closest point where any wheel drops below 20 rad/s is 82 s from the time of closest approach. It also means that the “low-jitter window”, i.e. the time frame around \( t_p \) where the wheels are in a desirable speed-range, is at least two-times this length (164 s).

In table IV, we summarize the objective function results from the full day simulation. The simulation was performed twice with the same operational parameters and commands, with the sole difference being the form of momentum planning. The first simulation was representative of what is typically done in SFL operations to date. That is, a set of “default” momentum setpoints are given to the spacecraft, in this case equivalent to 100 rad/s wheel speeds about each axis. The setpoints are simply chosen to be off-zero but still relatively low; no formal momentum planning for zero-crossing avoidance is performed. The simulation was then performed with setpoints generated by 3TECS. As we discussed in detail in section V, these setpoints are generated for each given pass to maximize the objective function value, \( \Delta t_{pzc} \). As we can see, in the worst pass for the default setpoints, the objective function value is zero, meaning that a zero-crossing occurred right at \( t_p \). On the other hand, with 3TECS being used, the worst-case pass had an objective function value of 82 s. The average objective function value for the case with 3TECS-generated setpoints is also much larger at 181 s, with a standard deviation of \( \sigma = 79 s \).

### VII. ON-ORBIT RESULTS FROM NORSAT-2

The NorSat-2 mission is currently operating on-orbit, and was used to test the momentum planning functionality of 3TECS. The target tracking trajectory planning portion of 3TECS requires additional on-board software, and will be tested on-orbit once a satellite with this newer software is launched, and is thus out of the scope of this paper. The test was performed twice, once on August 24th, 2017, and once on March 14th, 2018. Each test consisted of six passes. Operator inputs were passed to 3TECS for the given passes, following the architecture described VII. The time-tagged commands with the momentum setpoints were then uploaded to the spacecraft, and executed.

Figure 9 shows the on-orbit reaction wheel speed data for 2 representative passes from each test. For comparison, the on-orbit data is plotted against the wheel speeds that were expected based on simulation. Overall, the observed behaviour was in strong agreement with expectation. The on-orbit reaction wheel speeds closely followed the simulated profiles, and achieved more-or-less the same level of zero-crossing performance. It was found, in general, that there was about 5-10% error in

**TABLE IV: Comparison of zero-crossing objective function results from simulation, with and without 3TECS momentum planning**

<table>
<thead>
<tr>
<th>Momentum Planning</th>
<th>Objective Function Value [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimum</td>
</tr>
<tr>
<td>Default setpoints</td>
<td>0</td>
</tr>
<tr>
<td>3TECS-generated setpoints</td>
<td>82</td>
</tr>
</tbody>
</table>

![Fig. 8: Simulated wheel speeds during example target tracking pass with 3TECS-generated momentum setpoint](image-url)
the spacecraft’s actual momentum setpoint going into a target tracking pass, with respect to what was desired. There is a fundamental issue in the momentum management control due to the well-documented problem of instantaneous under-actuation using magnetorquers [10]. The torque applied from magnetorquers is always orthogonal to the local magnetic field vector, and thus we instantaneously lack the ability to apply torque (and regulate our momentum) about that direction. Additional sources of error include the inertial body momentum estimation error, the magnetic field estimation error, and the magnetorquer actuation error. In the worst-case, this error caused the zero-crossing to be shifted by about a minute, reducing the objective function value from 234 s to 175 s. This is visible in the $\omega_w,y$ plot for August 24, sample pass 1 (top-row center plot in figure 9).

### VIII. CONCLUSION

There exists a need for more sophisticated planning of the target tracking constraint vectors to allow for fine pointing with a single star tracker across a wide range of targets. In addition, there is a common problem of reaction wheel stiction near the zero angular velocity point causing jitter. In this paper, we then presented 3TECS, a piece of software developed with the dual purpose of target tracking trajectory planning for star tracker obscuration avoidance and momentum setpoint planning for reaction wheel zero-crossing placement. Star tracker availability results demonstrated that with 3TECS, across all examined orbits, it was possible to achieve > 98.7% availability, while tracking any target visible below the horizon, well surpassing the 95% goal. In turn, this high availability allows for high pointing accuracy; across the detailed simulations performed, pointing accuracy was observed to be better than 0.15° ($2\sigma$), well in compliance with the 0.55° ($2\sigma$) requirement. On the momentum setpoint planning side, simulation indicated that the 3TECS planner could allow for windows of no wheel zero-crossing of 164 s or longer for the cases studied, well above the 90 s target, and well outperforming the existing unplanned solution. The momentum setpoint planner was further validated through on-orbit testing on the NorSat-2 spacecraft.

### IX. ACKNOWLEDGMENTS

The author wishes to thank Dr. Robert E. Zee, for providing him the opportunity to study at the Space Flight Laboratory. All of the staff and students who contributed to the development of the DAUNTLESS bus are owed thanks for their hard work. The author would like to specifically acknowledge Niels Roth, who guidance and mentorship was essential for the work presented in this paper. Finally, the author would like to acknowledge the Norwegian Space Centre (NSC) for allowing on-orbit testing to be performed on NorSat-2, as well as Laura Bradbury for her help in facilitating the tests.

### REFERENCES


