A Constrained Attitude Control Module for Small Satellites

Henri Christian Kjellberg, E. Glenn Lightsey
Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin
210 East 24th Street Room 412 D. Austin, TX 78712; (281)622-9265
hck@utexas.edu, lightsey@mail.utexas.edu

ABSTRACT

The Satellite Design Laboratory at the University of Texas at Austin is building a general purpose guidance, navigation, and control (GN&C) module with 6 degree-of-freedom maneuver capability. The GN&C module is capable of meeting multiple pointing constraints autonomously utilizing new constrained attitude control algorithms. Attitude keep-out zones are avoided by first discretizing the unit sphere into a graph using an icosahedron-based pixelization subroutine. An admissible path is found using the A* pathfinding algorithm. The trajectory is followed by a rate and torque constrained quaternion feedback controller. The algorithm is capable of running in real-time on a low power embedded flight computer. The module has secured flight opportunities on two student-built 3U CubeSats for flight projects sponsored by the Air Force and NASA. Both sets of mission requirements are satisfied with the same 3U CubeSat attitude control system, demonstrating the algorithm's versatility as a general purpose controller. The autonomy provided by the advanced constrained control algorithms enables more complex picosatellite missions and decreases the cost of spacecraft subsystems by shifting requirements away from the hardware and onto the control algorithm.

MOTIVATION

Picosatellites have the potential to reduce the cost of conducting missions in space. Programs such as NASA Ames's GeneSat and the National Science Foundation's CubeSat-based Science Missions for Space Weather and Atmospheric Research underscore the notion that CubeSats are increasingly being considered as viable platforms for conducting scientific research.[1,2] However, many measurement and communication payloads require pointing, or orbital maneuvers. Currently, missions that can be performed by picosatellites are limited by the lack of advanced, miniaturized six degree-of-freedom guidance, navigation, and control (GN&C) systems. Development of a reusable, autonomous GN&C module design is needed in order to enable increasingly complex missions to be conducted in the CubeSat form factor.

In support of NASA Johnson Space Center's "Low Earth Orbiting Navigation Experiment for Spacecraft Testing Autonomous Rendezvous and docking" (LONESTAR) program, the University of Texas at Austin is developing a six degree-of-freedom capable 3-Unit CubeSat named Bevo-2, which is scheduled for launch in 2013.[3] Figure 1 shows the Bevo-2 3-Unit CubeSat. The mission of Bevo-2 is to demonstrate the technologies necessary to perform autonomous rendezvous and docking in orbit between two small satellites. In order to satisfy these mission goals, a self-contained, bolt-on GN&C module for CubeSats is being designed and tested. The module shall be capable of performing attitude determination and control using commercial-off-the-shelf (COTS) sensors and actuators. One goal is to allow more complex CubeSat missions to be developed and integrated quickly. In fact, the module's design that is demonstrated on Bevo-2 is planned to be reused on another University of Texas CubeSat, "Attitude Related Maneuvers And Debris Instrument sateLlite in Low Orbit" (ARMADILLO) which is an entry in the Air Force's University Nanosat Program and is planned for launch in 2014.[4] The development of the GN&C module hardware is being supported through a NASA Small Technology Transfer (STTR) grant.[5]

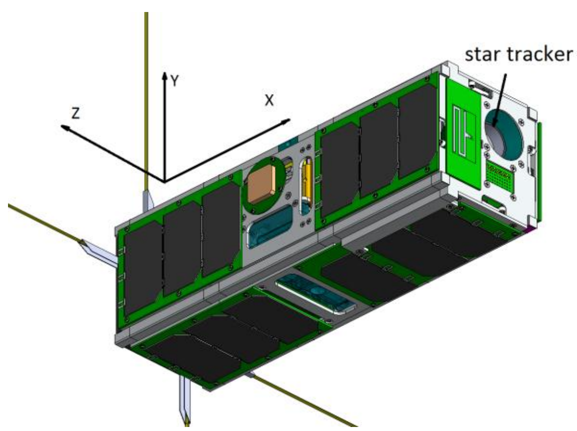

Figure 1: The Bevo-2 CubeSat showing the body coordinate frame and location of the star camera.

Picosatellites (such as CubeSats) generally are highly constrained in their capabilities due to their small size. Innovative solutions are needed to achieve mission requirements within the constraints imposed by the mission payload and GN&C hardware. Large constellations of picosatellites in formation may have complex pointing and formation requirements. As the number of spacecraft increases, the ability of ground controllers to monitor individual spacecraft in the constellation decreases. Therefore, these spacecraft must become capable of satisfying their own state constraints more autonomously.

GN&C MODULE HARDWARE

The GN&C module that will be used to demonstrate the constrained attitude control scheme is shown in Figure 2. This module is approximately 1-Unit in size and houses sensors, actuators, and embedded controllers necessary to perform 6 degree-of-freedom maneuvers.

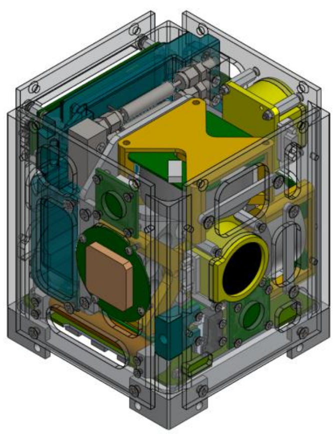

Figure 2: The GN&C module being designed for the Bevo-2 and ARMADILLO missions.

Figure 3 shows an exploded view of the GN&C module hardware with labels. The system utilizes a set of 3 gyroscopes, two sun sensors, and a magnetometer for basic attitude determination. An external star tracker (not shown) provides more accurate attitude determination results when available. A set of 3 reaction wheels are utilized for pointing control. Momentum is managed with a pair of magnetic torque rods. A compact cold-gas thruster allows for small translational impulses.

The GN&C module is being completed during summer 2012.The GN&C module electronic test unit (ETU) has been previously completed allowing for software development and hardware-in-the-loop simulation. The electronic test unit is shown in Figure 4. The ETU provides the opportunity to evaluate different constrained attitude control methods to see if they are computationally efficient enough to run in real-time on the low power, 200 MHz, ARM 9 embedded flight computer.

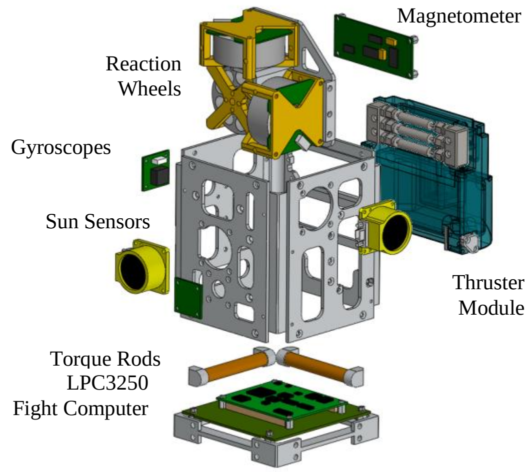

Figure 3: An exploded view of the GN&C module with component labels.

CONSTRAINED ATTITUDE CONTROL

Managing the attitude constraints through algorithms and software can decrease the cost of a spacecraft mission. SpaceDev's "Trailblazer" microsatellite employed hardware that demanded less power, used less volume, and cost less due to its GN&C system's ability to slew the spacecraft while simultaneously satisfying pointing constraints. Costly modifications to its science instrument were avoided by guaranteeing that sun avoidance was achieved through constrained attitude control.[6]

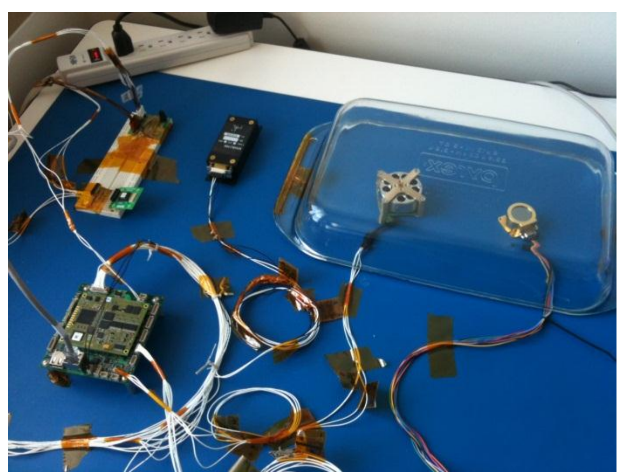

Figure 4: GN&C electronic test unit.

Autonomy has many possible definitions; there are varying degrees of autonomy. By allowing a spacecraft to satisfy some of its constraints on its own, a degree of autonomy can be achieved. Control constraints include the physical constraints of sensors, such as the inequality constraint formed from rate measurement saturation of a gyroscope, or an integral inequality constraint such as the wheel speed limitations of a reaction wheel. Spacecraft state constraints include things such as hard or timed pointing exclusion zones, or spacecraft keep-out regions. Figure 5 shows several common constraints. As long as a feasible trajectory exists to an admissible desired state, the spacecraft should be capable of autonomously choosing a trajectory to the new state that does not violate any constraint requirements. Thus, mission controllers can focus on high level commands to the spacecraft and allow the spacecraft to make sure that operational constraints are maintained.

A spacecraft like Bevo-2 can benefit from the ability to satisfy both control and state constraints autonomously in real-time. Bevo-2 is designed to fly with only two digital sun sensors (instead of six) and a star tracker without a large baffle in order to decrease size and cost. Therefore, the scheme presented in this paper provides the capability to avoid bright objects along the axis of the star tracker (shown as the x-axis in Figure 5) while maintaining the Sun within the field of view of the sun sensor. Each constraint is identified by a vector to the center of the constraint and a half-cone angle indicating its size. By assembling these constraints together, mission controllers can describe the spacecraft hardware requirements in a very high level and allow the spacecraft to find the best way to satisfy those constraints during attitude maneuvers.

Traditional Methods

There are multiple ways to solve specific formulations of the constrained attitude control problem. Consider the attitude avoidance cone problem represented by a sensitive instrument that cannot get within a certain angle of the Sun.

McInnes provided a straightforward example of constrained control that augments potential functions with high potential around the avoidance regions and applying Lyapunov's 2nd method in order to prevent the spacecraft state from passing through the forbidden region as it guides the spacecraft state to the desired state.[7]

Alternatively, Hablani approached the problem from a geometric perspective by creating ideal tangential paths around exclusion cones using vectors on a unit sphere.[8]

Other approaches have been presented by Frazzoli that utilize randomized planning algorithms.[9,10] Here, virtual target attitudes are chosen at random and a tree of possible paths are evaluated to that location. The lowest cost admissible path is chosen. The algorithm can be iterated to improve the efficiency of the attitude trajectory.

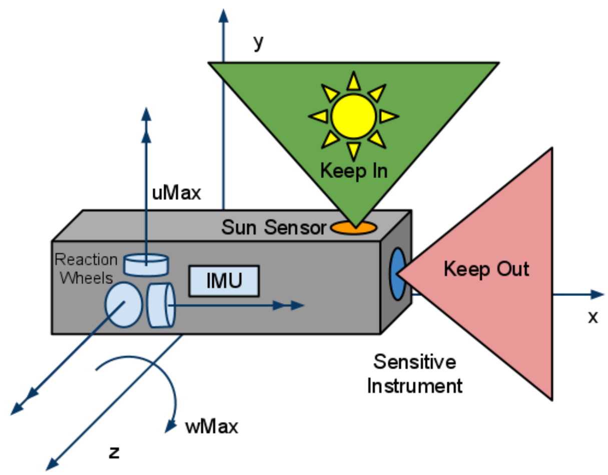

Figure 5: Common spacecraft sensor and actuator constraints.

A succinct and easy to implement alternative to the methods described above uses a form of numerical optimization called semi-definite programming (SDP). SDP allows for optimization of a specific subset of convex problems. The equations that govern the constrained attitude control problem in their generic form are non-convex. However, Kim shows that the problem can be reformulated from a non-convex representation into a convex problem statement by using the quaternion representation of attitude and its unity constraint.[11,12] Efficient optimizers exist for convex problems. Using SDP, the optimization problem is expressed as a set of linear objective functionals and constraints expressed as matrix inequalities. Specifically, the method requires the constraint matrix to be positive semidefinite. The SDP method can use powerful SDP optimization toolkits such as CSDP that are readily implemented onto a spacecraft flight computer.[13]

Contributions of this Paper

Any of the above approaches could be utilized to meet the requirements of the Bevo-2 spacecraft. In particular, the authors have demonstrated the SDP approach proposed by Kim and fount it to be very efficient. However, a new approach is proposed here that leverages the discretization of the unit sphere into a graph and applies efficient, simple graph pathfinder algorithms coupled with a body rate and torque constrained quaternion feedback controller. The new method is demonstrated in a simulation for the Bevo-2 mission.

A DISCRETIZED CONSTRAINED ATTITUDE CONTROL APPROACH

The scenario constructed from Bevo-2 hardware requirements consists of several attitude constraints. The sensitive star tracker, which is aligned along the x-axis, must avoid the Earth, the Sun, and the Moon all with different avoidance cone sizes. The star tracker has a field of view of ±20 degrees. The sun sensor, which is aligned with the y-axis (see Figure 5), must maintain the Sun within the sensor's ±70 degree field of view. An additional attitude constraint is given to keep the spacecraft attitude within a range in which an admissible rotation exists to allow the sun sensor to view the Sun at all times. This constraint only comes into play when the sun is within 10 degrees of the Earth limb and takes the form of a ±20 degree cone in the anti-Sun direction. Additionally, the vehicle body rates are limited to 5 degrees per second to prevent gyroscopes from saturating, and the reaction wheels are only able to deliver up to 1 mNm of torque.

Algorithm Overview

Figure 6 shows a block diagram of the new discretized constrained attitude control scheme. The algorithm begins by discretizing the unit sphere using an efficient icosahedron based pixelization subroutine. The discretization subroutine allows for translating a unit vector into an integer that identifies the pixel that is closest to that vector and vice versa. With the attitude unit sphere discretized into a graph of pixels, an admissible and short path from the starting pointing vector to the desired pointing vector is found using the A* pathfinding algorithm.[14] From this path, a series of quaternions are formed that describe the rotation sequence (with adjustments to maintain sun sensor line-of-sight with the Sun). Finally, the quaternion trajectory is followed using a constrained quaternion feedback controller.

Icosahedron-based Discretization of the Unit Sphere

Astronomers and cosmologists working with measurements of the cosmic microwave background have developed methods for pixelizing the celestial unit sphere. In particular, two methods are the quadrilateralized spherical cube (also known as the COBE sky cube) algorithm and more recently the icosahedron-based scheme.[15,16] In the COBE sky cube algorithm, a sphere is first inscribed inside a cube. The faces of the cube which are pixelated in a square grid are then projected onto the sphere. Finally, the pixels are shifted to minimize the area variation in the square area that can is attributed to each pixel. Alternatively, the icosahedron-based scheme begins by inscribing a sphere inside an icosahedron. Pixels are distributed evenly on each triangular face and then projected onto the sphere. Shifting minimizes the variation in the area occupied between neighboring pixels. In this case the areas are hexagonal.

The icosahedron-based approach has two factors that allow it to be more useful in the application of constrained attitude guidance. First, a smaller number of pixels are required for the icosahedron based approach because the area of the hexagons that each pixel is occupies is more circular than the squares in the COBE sky cube. Second, each pixel has a set of six approximately equidistant neighbors, as opposed to the COBE sky cube which has only four equidistant neighbors. Each pixel's set of neighboring pixels needs

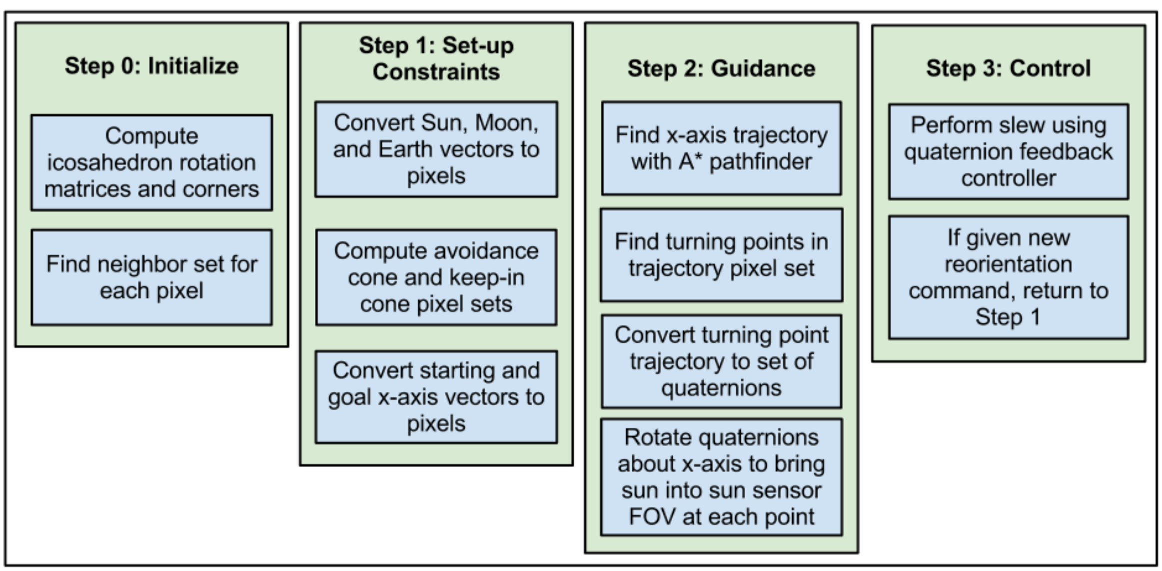

Figure 6: Block diagram of the discretized constrained attitude guidance and control scheme.

only to be computed once. The number of pixels per face of the icosahedron is chosen by a resolution parameter, in this application a resolution of 12 was utilized to give approximately 4411 pixels on the sphere. The average angle between any two neighboring pixels is 3.3 degrees. Figure 7 shows the pixelized sphere with an icosahedron inscribed within. The icosahedron algorithm provides two subroutines. One subroutine takes a vector and provides the identification number of the pixel that represents the hexagon into which the vector is pointing. The second subroutine takes a pixel identification number and provides a vector to the center of the hexagon.

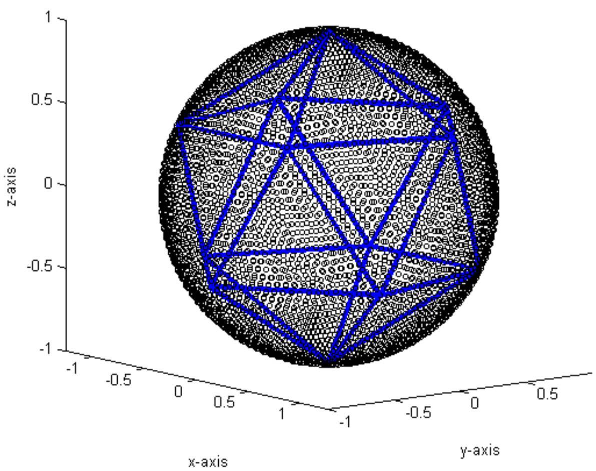

Figure 7: 4411 pixels distributed across the unit sphere.

Pathfinder Algorithm

With the attitude sphere discretized, the problem becomes approachable from a vast array of graph search algorithms. In this particular case, the A* algorithm was chosen because of its ease of implementation. The A* algorithm is described in detail in by Hart.[14] In order to utilize the A* algorithm the set of pixels must be formed into a graph. In order to do this, each pixel must identify its neighboring pixels. This data is searched beforehand and stored on the computer. Next, the path-cost function $g(p(k))$ where $p(k)$ is the pixel node at step k, and the heuristic estimate $h(p(k))$ is needed. The path-cost function and the heuristic are added together to form the distance-plus-cost function.

$$f(p(k)) = g(p(k)) + h(p(k)) \quad (1)$$

The path-cost function in its simplest form will just be the number of degrees between the current pixel and the neighboring pixel. However, a penalty is added in order to minimize the number of segments in the attitude trajectory that require different steady state body rates. This approach allows the trajectory solution to be executed more efficiently by the attitude controller.

If $p(k)$ is a pixel that is not within the prohibited constraint set, then

$$g(x(k)) = \theta(e(k), e(k-1)) + d + g(x(k-1)) \quad (2)$$

Here, $e(k)$ represents the vector connecting $p(k)$ and $p(k-1)$. The function $\theta(e(k), e(k-1))$ is the angle between the $e(k)$ and $e(k-1)$, thus penalizing a change in the eigenaxis vector direction. Figure 8 shows the angle between the pixels. The parameter d is the average degrees between each neighboring pixel, which is 3.3 degrees in this example.

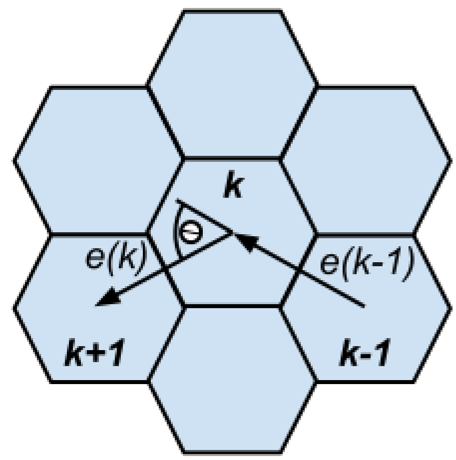

Figure 8: The function $\theta(e(k), e(k-1))$ finds penalizes changing the direction of the x-axis travel on the unit sphere.

If however, $p(k)$ is in the prohibited constraint set then $g(p(k)) = N$ where N is an arbitrarily large constant to prevent a solution from going within the constrained region.

The heuristic is simply calculated as the angle between the current pixel and the final pixel. On the unit sphere, this is the minimum arc length between the current pixel and the final pixel.

$$h(p(k), p(m)) = \arccos\left(v(p(k)) \cdot v(p(m))\right) \quad (3)$$

Here the function $v(p(k))$ converts the pixel at $p(k)$ to a unit vector and $p(m)$ represents the final pixel.

With these functions defined, the A* search algorithm returns the set of pixels that provide an admissible, minimum path-cost (as described by the path-cost function) trajectory for the desired rotation of the vehicle's body-fixed x-axis to the new attitude. This set of pixels is reduced by eliminating all pixels that lie along approximately the same path on the great circle. The remaining pixels are “turning points,” locations where the eigenaxis vector direction must change. The resultant trajectory for the vehicle's body-fixed x-axis is shown in Figure 9 for a typical reorientation problem.

Sun Sensor Keep-in Constraint

Now that the x-axis trajectory has been found, the three-axis rotations that also satisfy the sun sensor keep-in constraint must be computed.

It is assumed that the maneuver begins from an attitude where the sun sensor has the Sun within its field of view. Given a desired inertial target vector x-axis, $\boldsymbol{n}_x^I(k+1)$, the quaternion that rotates the spacecraft’s x-axis from its current attitude to the next turning point is calculated with:

$$\begin{bmatrix} \boldsymbol{n}_x^B(k+1) \\ 0 \end{bmatrix} = \underline{\boldsymbol{q}}_B^I(k) \otimes \begin{bmatrix} \boldsymbol{n}_x^I(k+1) \\ 0 \end{bmatrix} \otimes \underline{\boldsymbol{q}}_B^{*I}(k) \tag{4}$$

$$\phi = \arccos\left(\boldsymbol{n}_x^B(k+1) \cdot \boldsymbol{n}_x^B(k)\right) \tag{5}$$

$$\underline{\boldsymbol{q}}_{B'}^B(k+1) = \begin{bmatrix} \dfrac{\left(\boldsymbol{n}_x^B(k+1) \times \boldsymbol{n}_x^B(k)\right)}{\|\boldsymbol{n}_x^B(k+1) \times \boldsymbol{n}_x^B(k)\|} \sin\dfrac{\phi}{2} \\ \cos\dfrac{\phi}{2} \end{bmatrix} \tag{6}$$

Here, $\boldsymbol{n}_x(k+1)$ is the vector to the next turning point x-axis vector. The superscript I represents the inertial reference frame, B represents the current body frame, and B' represents the body frame at the next turning point. Note that $\boldsymbol{n}_x^B(k) = [1\ 0\ 0]$. The new quaternion from the inertial frame to the body frame then becomes

$$\underline{\boldsymbol{q}}_{B'}^I = \underline{\boldsymbol{q}}_{B'}^B \otimes \underline{\boldsymbol{q}}_B^I \tag{7}$$

However, this quaternion does not necessarily preserve the requirement of maintaining the Sun within the line-

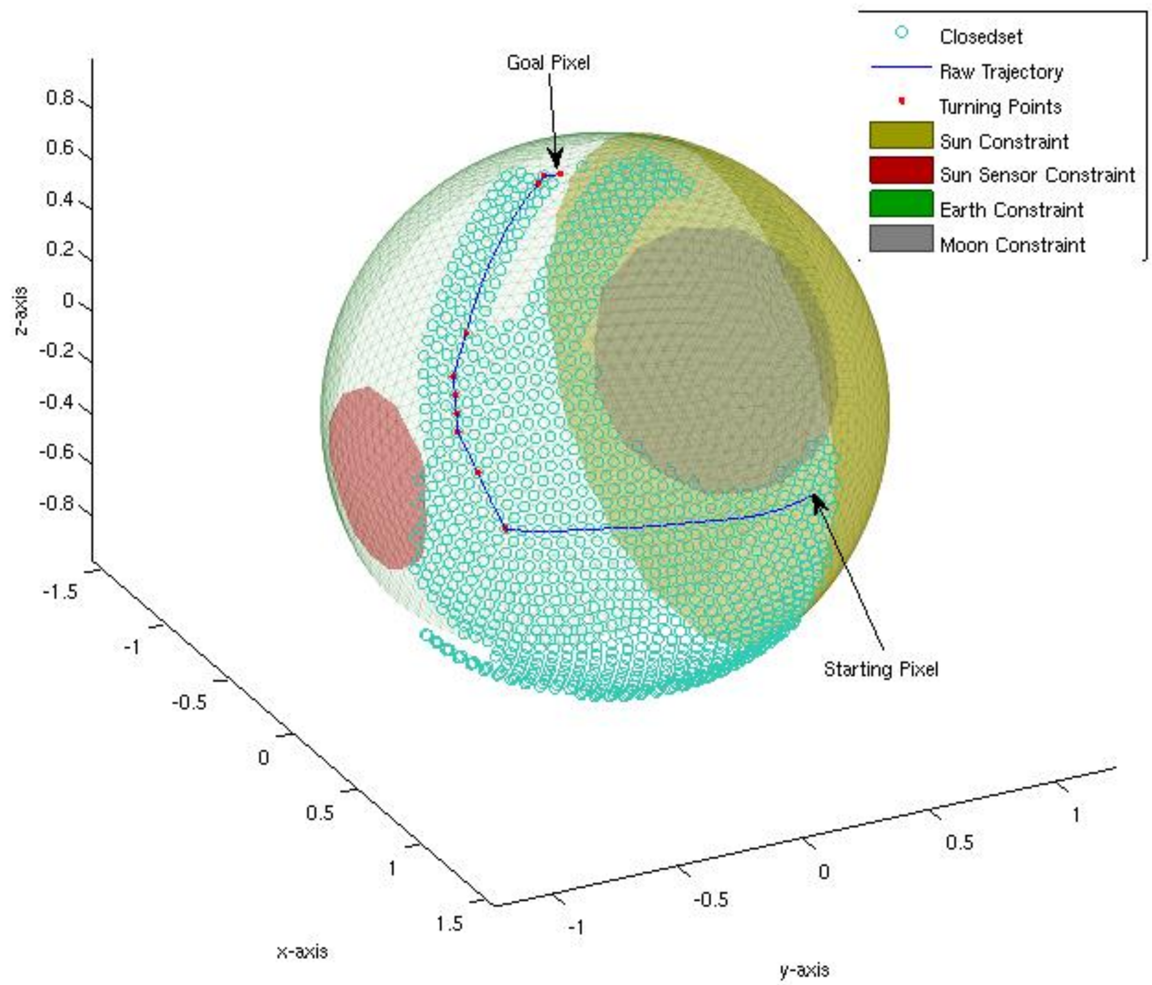

Figure 9: Example x-axis unit sphere trajectory resulting from the A* pathfinder algorithm.

of-sight of the sun sensor. A rotation about the x-axis can be performed to make sure that the Sun is maintained in the sun sensor line-of-sight without causing the sensitive x-axis to stray from its trajectory. The sun sensor alignment rotation quaternion can be found by first rotating the Sun vector in the inertial frame to the body frame

$$\begin{bmatrix} \boldsymbol{n}_s^{B'}(k) \\ 0 \end{bmatrix} = \underline{\boldsymbol{q}}_{B'}^{I}(k) \otimes \begin{bmatrix} \boldsymbol{n}_s^{I}(k) \\ 0 \end{bmatrix} \otimes \underline{\boldsymbol{q}}_{B'}^{*I}(k) \tag{8}$$

Keeping only the y and z components of the vector and setting the x component to 0 forms

$$\boldsymbol{n}_s^{\star B'}(k) = \left[0 \; \boldsymbol{n}_{s_y}^{B'}(k) \; \boldsymbol{n}_{s_z}^{B'}(k)\right] \tag{9}$$

The eigenaxis vector that is sought is aligned with the x-axis of the spacecraft, however to get the direction of the rotation find the cross product between $\boldsymbol{n}_s^{\star B'}(k)$ and $\boldsymbol{n}_y^{B'}(k) = [0\ 1\ 0]$. The angle itself can be found with the inverse cosine of the dot product of the two vectors.

$$\underline{\boldsymbol{q}}_{B''}^{B'}(k+1) = \begin{bmatrix} \dfrac{\left(\boldsymbol{n}_y^{B'}(k) \times \boldsymbol{n}_s^{\star B'}(k)\right)}{\left\|\boldsymbol{n}_y^{B'}(k) \times \boldsymbol{n}_s^{\star B'}(k)\right\|} \sin\dfrac{\psi}{2} \\ \cos\dfrac{\psi}{2} \end{bmatrix} \tag{10}$$

$$\psi = \arccos\left(\boldsymbol{n}_y^{B'}(k) \cdot \boldsymbol{n}_s^{\star B'}(k)\right) \tag{11}$$

The final rotation that satisfies all the pointing constraints is given by

$$\underline{\boldsymbol{q}}_{B''}^{B} = \underline{\boldsymbol{q}}_{B''}^{B'} \otimes \underline{\boldsymbol{q}}_{B'}^{B} \tag{12}$$

Constrained Quaternion Feedback Controller

To maintain the body rate and actuator torques within the limits of the spacecraft hardware, a constrained rate and torque feedback controller as described by Wie is

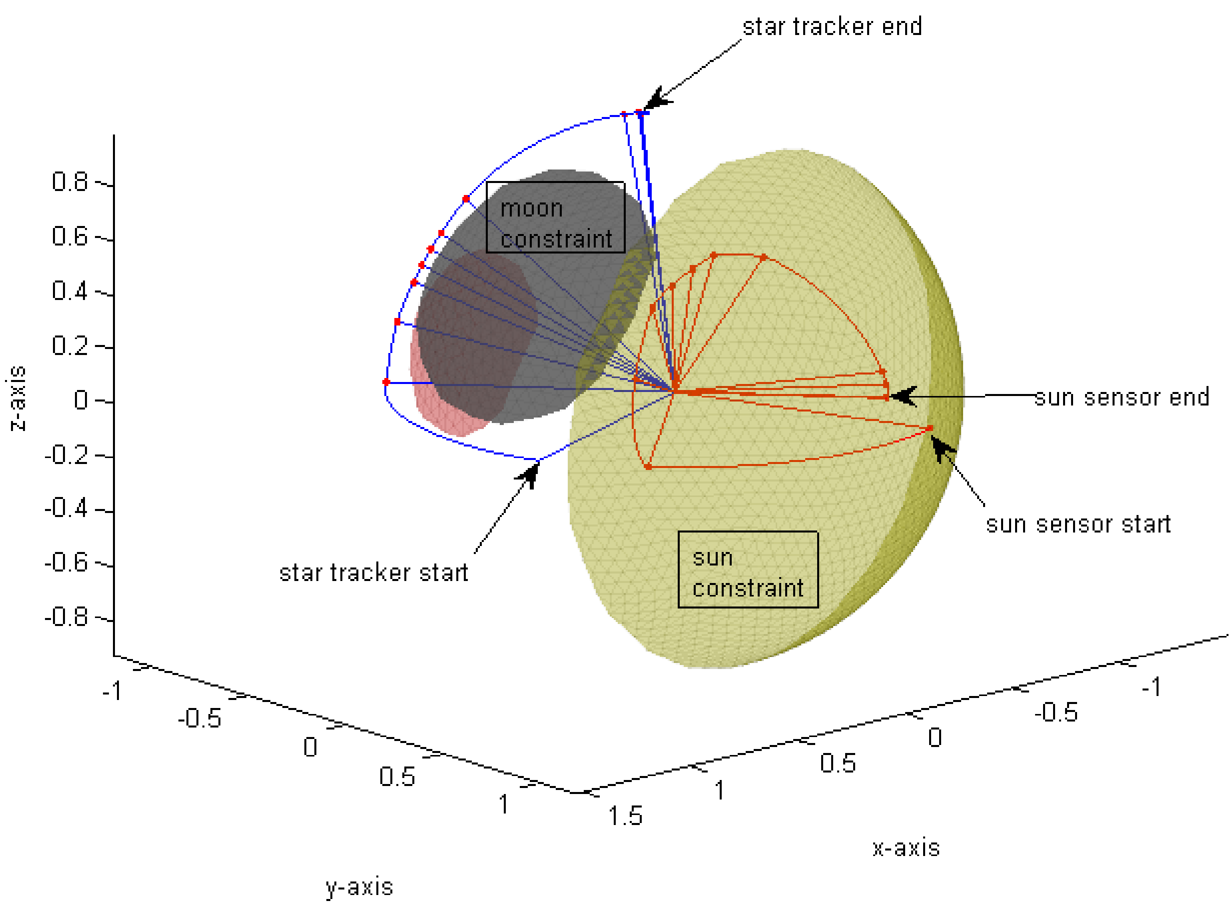

Figure 10: Star tracker trajectory and sun sensor trajectory with the sun sensor field of view keep-out constraint (red), moon keep-out constraint (grey), and sun sensor keep-in constraint (yellow) visible. Earth constraint is hidden for clarity.

implemented.[17] The controller assumes a rigid body vehicle. The control signal $\boldsymbol{u}$ is determined by the relation

$$\boldsymbol{u} = -sat[\boldsymbol{K} sat(\boldsymbol{P}\boldsymbol{q}_e) + \boldsymbol{C}\boldsymbol{w}] \tag{13}$$

Here, $\boldsymbol{q_e}$ represents the quaternion error between the current attitude quaternion and the goal attitude quaternion. $\boldsymbol{C}$ is a diagonal gain matrix and $\boldsymbol{w}$ is the angular velocity of the spacecraft. The i elements of the diagonal gain matrix $\boldsymbol{K}$**,** are obtained from Eq. 14.

$$\boldsymbol{K}_i = -\frac{\left|q_{i}{}_{B''}^{I}(k-1)\right|}{\left\|\boldsymbol{q}_{B''}^{I}(k-1)\right\|}\dot{\theta}_{\max}, \quad i = 1,2,3 \tag{14}$$

The gain matrix $\boldsymbol{P}$ can be solved using Eq. 15.

$$\boldsymbol{K}\boldsymbol{P} = c\boldsymbol{J}\dot{\theta}_{\max} \tag{15}$$

Here, c is a positive scalar, and $\boldsymbol{J}$ is the spacecraft inertia matrix. The outer saturation function uses the maximum torque (1 mNm in this case) to normalize the torque in order to maintain the same torque vector direction. As a result the body rate will never exceed $\dot{\theta}_{\max} = 5$ deg/s and the reaction wheel torque will not exceed $\tau_{\max} = 1$ mNm.

Results

A full dynamic simulation of a slew maneuver is shown in Figure 10. Here the blue trajectory shows the star camera on the spacecraft x-axis successfully avoids the Sun (yellow), Moon (grey), and sun sensor field of view limitation (red) keep-out constraints as it rotates to the desired attitude. The sun sensor field of view limitation constraint (red) exists to keep the x-axis within a range that allows the sun sensor to bring the sun within the sensor's ±70 degree field of view with a rotation about the x-axis. The red trajectory shows the sun sensor on the y-axis maintaining the sun within its field of view (yellow).

The momentum and torque histories are shown in Figure 11. Here the control constraints are enforced as the reaction wheel torques are limited to 1 mNm on the vehicle body axes. The reaction wheel speeds for these wheels are shown to provide momentum less than 10 mNm-s, however this constraint is not actively enforced with the constrained guidance and control approach presented in this paper. The wheel speed constraints are left for future work. Figure 12 shows the spacecraft angular velocity. These are constrained to be smaller than 5 deg/s in order to keep the gyroscopes within operation limits and prevent saturation.

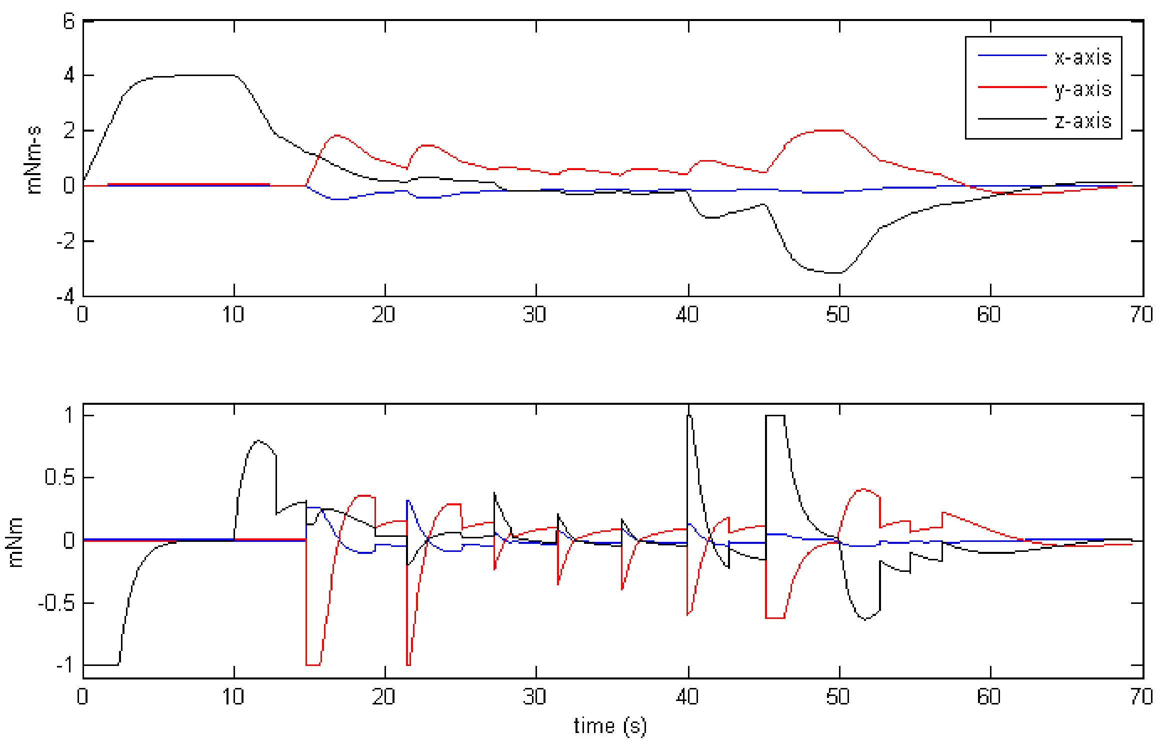

Figure 11: Momentum (top) and torque (bottom) histories. Note that the torque is constrained at ±1 mNm.

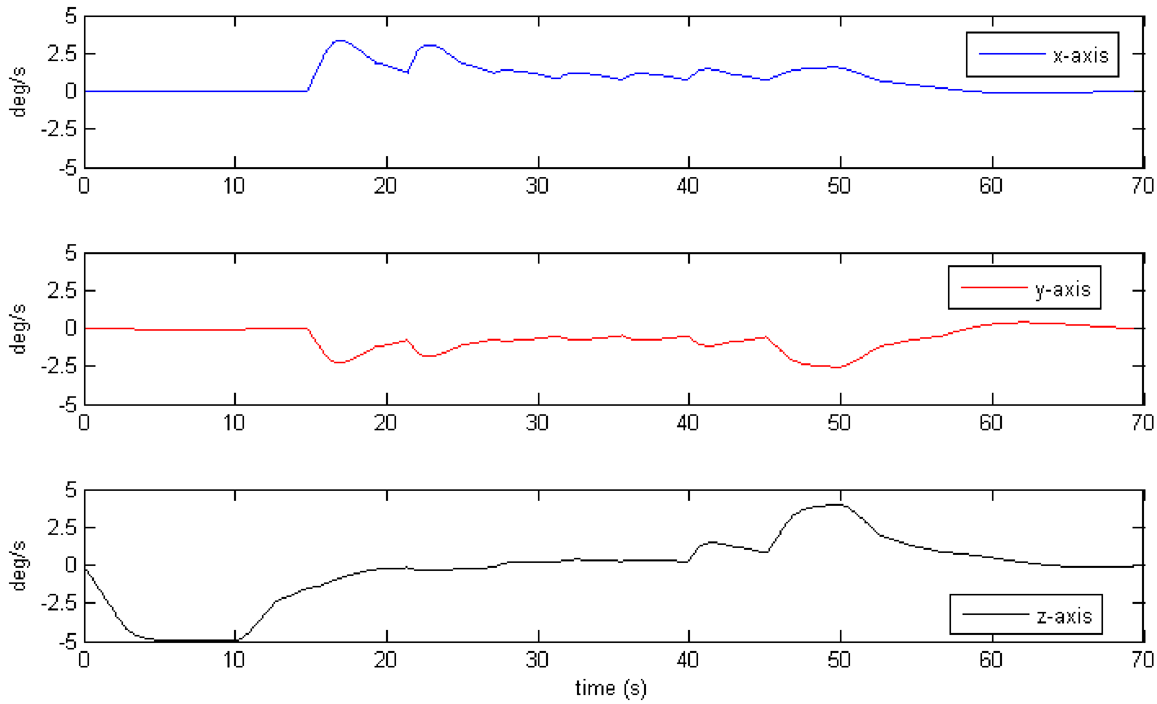

Figure 12: Angular velocity of the spacecraft. Note that the angular velocity is constrained to ±5 deg/s.

CONCLUSION AND FUTURE WORK

The discretized pathfinding approach to constrained attitude control presented here was designed to be general and satisfies the pointing requirements of both the Bevo-2 and ARMADILLO 3U CubeSat spacecraft. Future work will focus on integrating reaction wheel speed, additional non-rigid body dynamics, and different control models for the spacecraft and dynamic constraint sources directly into the graph costs. Guiding the coupled dynamics of all axes simultaneously using the graph approach rather than solving first the x-axis and then the remaining axis may prove effective at satisfying more complex constraints.

ACKNOWLEDGMENTS

The authors wish to thank the NASA Space Technology Research Fellowship program grant number NNX11AN26H for funding continued research in the field of constrained control and our NASA technical mentor Dr. Steve Provence. We would also like to recognize the students of the Satellite Design Laboratory who design and build the spacecraft which provide the relevance for this research. Special thanks to Travis Imken for providing some illustrations.

REFERENCES

1. Yost, B., et al., “The GeneSat-1 Test Demonstration Project: A Unique Use of SmallSats,” Proc. 19th Annual AIAA/USU Conference on Small Satellites, Logan UT, 2005.
2. Jorgensen, T., “CubeSat-based Science Missions for SpaceWeather and Atmospheric Research,” Internet, July 2006, National Science Foundation: http://www.nsf.gov/pubs/2010/nsf10537/nsf10537.htm, Date accessed: May 11, 2012.
3. Lightsey, E.G., Muñoz, S., and Brumbaugh, K., "The TEXAS Satellite Design Lab - An Overview of Our Current Projects", 8th Annual CubeSat Workshop, San Luis Obispo, CA, April 20-22, 2011.
4. Brumbaugh, K.M., Kjellberg, H.C., Lightsey, E.G., Wolf, A., Laufer, Rene., “In-Situ Sub-Millimeter Space Debris Detection Using CubeSats,” 35th AAS Guidance & Control Conference, AAS 12-001, Breckenridge, CO., February 2012.
5. Lightsey, G., “Guidance, Navigation, and Control System for Maneuverable Pico-Satellites,” Internet, May 2010, NASA STTR 2008 Proposal

Summary: http://sbir.gsfc.nasa.gov/SBIR/abstracts/08/sttr/phase2/STTR-08-2-T6.01-9905.html, Date accessed: May 11, 2012.

6. Koenig, J. D., "A Novel Attitude Guidance Algorithm for Exclusion Zone Avoidance," Aerospace Conference, 2009 IEEE, pp.1-10, 14(7) March 2009.

7. McInnes, C. R., "Large Angle Slew Maneuvers with Autonomous Sun Vector Avoidance," AIAA Journal of Guidance, Control, and Dynamics, 17(4):875-877, 1994.

8. Hablani, H., B., "Attitude Commands Avoiding Bright Objects and Maintaining Communications with Ground Station," AIAA Journal of Guidance, Control, and Dynamcis, pp.759-767 22(6), December 1999.

9. Frazzoli, E., Dahleh, M. A., Feron, E., and Kornfeld, R., "A Randomized Attitude Slew Planning Algorithm for Autonomous Spacecraft," Proceedings of the AIAA Guidance Navigation, and Control Conference, Montreal, Canada 2001.

10. Frazzoli, E., Dahleh, M. A., Feron, E., and Kornfeld, R., "Real-Time Motion Planning for Agile Autonomous Vehicles," Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit, Denver, CO, August 2000.

11. Kim, Y., Mesbahi, M., Singh, G., and Hadaegh, F.Y. "On the Constrained Attitude Control Problem," Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. Providence, RI, August, 2004.

12. Kim, Y., Mesbahi, M., Singh, G., and Hadaegh, F.Y., "On the Convex Parameterization of Constrained Spacecraft Reorientation," Aerospace and Electronic Systems, IEEE Transactions on, vol.46, no.3, pp. 1097-1109, July 2010.

13. Borchers, B., "CSDP, a C Library for Semidenite Programming," Optimization Methods & Software, 11-2(1-4:613-623, 1999.

14. Hart, P.E., Nilsson, N.J., Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," pp. 100-107, IEEE Transactions on Systems Science and Cybernetics, July 1968.

15. O'Neill, E., and Laubscher, R.E., "Extended Studies of a Quadrilateralized Spherical Cube Earth Database," Computer Sciences Corp. EPRF Tech. Rep. 1976.

16. Tegmark, M., "An Icosahedron-Based Method for Pixelizing the Celestial Sphere," The Astrophysical Journal Letters., Vol. 470, 81-84, , October 20, 1996.

17. Wie, B., Space Vehicle Dynamics and Control, Second Edition, AIAA Education Series, AIAA, 2008.