IAA-CU-13-11-06
Using independent combinations of CubeSat solar panels as sun sensors for separating inbound sunlight information

Martin Nygren*, Roger Birkeland†, Jan Tommy Gravdahl‡

Abstract

The NTNU Test Satellite (NUTS) is a satellite being built as a student CubeSat project. The project started in September 2010 as a part of the Norwegian student satellite program run by Norwegian Centre for Space-related Education (NAROM). The NUTS project goals are to design, manufacture and launch a double CubeSat by 2014. Using solar panels on 5 of 6 sides of the NUTS CubeSat, we have a lot of attitude determinating information readily available during much of the periodical orbit. Using three or more solar panels in combination can trigonometrically produce a sun vector. However, treating all inbound sunlight as direct light from the sun leaves the result vulnerable to disturbances like the Earth albedo effect (reflected sunlight from the Earth). Not compensating for this effect can corrupt or introduce large offsets in computed sun vectors. Treating the output current from the solar panels as an angle of 3D inbound light towards the 2D plane of the panel can produce a set of independent mathematical conical shells. Given the cubic shape of the satellite, these independent conical shells can iteratively be combined into sets of two and two orthogonal conical shells which will produce intersection curves with eachother if the angle of aperture is large enough. Even though the panels are physically spaced apart, it can be shown that without loss of generality, these conical shells can be translated along their respective axis of the satellite Body frame to have overlapping origins, which we can use to map the convergences of the parabolic intersection curves into two possible inbound sun vectors for each pair of solar panels. Knowing that opposing panels can't be combined to yield valid information, our 5 sides of panels can be combined into 8 different pairs from which we can interpret information. Algorithms to compare and evaluate the produced cluster of vectors to extract desired information and suppress disturbance are being investigated. Making use of instantaneous and tracked individual vectorial attitudes as well as their rates of change seems promising, however work remains to thoroughly physically test the mathematical model before a full blown algorithm can be made.

*NTNU - Dept. of Engineering Cybernetics, Norway, martinyg@stud.ntnu.no
†NUTS project lead, NTNU - Dept. of Electronics and Telecom., Norway, roger.birkeland@iet.ntnu.no
‡Prof. NTNU - Dept. of Engineering Cybernetics, Norway, jan.tommy.gravdahl@itk.ntnu.no

Motivation

The Attitude and Determination System (ADCS) on the 2U CubeSat NTNU Test Satellite (NUTS) require reference vectors to be able to properly compensate for gyroscopic bias drift, and one of these reference vectors will be a calculated Sun vector. As the solar panels leaves almost no uncovered surface space on the satellite, it is desireable to be able to utilize these as Sun sensors. Solar cells are indeed not truly sensors in their own, but can be used to detect the direction to the Sun by monitoring the output current. The output from a solar cell depends on the angle between the solar panel and the received rays of light. Previously done modeling and testing concludes with that a simple 3-angle trigonometric model employed is too susceptible to Earth albedo disturbance, as the solar panels are sensitive to every light source in space. This means that a compensation is needed in order to avoid producing a vector with a possibly large angular deviation from the desired true Sun vector. In general, this leaves two possible courses of action; either to implement a model of the Earth albedo intensity for calculating the magnitude of the expected disturbance, or to develop a model which can dynamically compensate for the experienced disturbance. This paper aims to investigate the latter option.

Previous Work

The CubeSat projects at NTNU has already seen a good amount of work towards attitude determination, using magnetometers and sun vector for reference vectors to compensate for gyro bias drifting. A presentation of the work, including control theory, can be seen in the publishings of Gravdahl et al. [1], building on the work of Svartveit [2] in regards to coarse sun sensing. The publishings of Jenssen [3] shows a comprehensive study on the development and comparison of the Extended Quaternion Estimator (EQUEST) and the Extended Kalman Filter (EKF), a work which was further developed as shown in the publishing of Rinnan [4]. In the presentations of Holberg [5], the attitude determination system is described to utilize the values for power delivered from each of the panels and integrate them into the EQUEST algorithm. In this work, a Sun vector is found by computing the directional sun intensity from the intensity measured by each of the panels.

During these CubeSat projects, the attitude determination problem has undergone iterative changes and incremental improvements. Svartveit [2] and Ose [6] made work on a discrete Kalman filter, which was subsequently made into an extended Kalman filter in the work of Rohde [7]. Based on the work of Psiaki [8] and Markley [9], Jenssen and Yabar [10] made comparisons of the estimation methods used in the extended Kalman filter, by extending the quaternion estimation (QUEST) method to include non-vectorized gyroscope measurements and prediction terms, making it more suitable for attitude estimation. Rinnan [11] developed the EQUEST further, extending it to include attitude predictions and gyroscope information in the cost function, as well as comparing the developed EQUEST method to a nonlinear observer and combining the two.

Sun vector: A conical shell model

By using a simple trigonometric approach, we would harvest all available incoming sunlight information indiscriminately, and piece it together in the form of a single sun vector given inbound sunlight angles of all three active solar panels. A different approach is to treat each solar panel surface as a 2D sensor with a 1D output. To illustrate the concept, refer to Figure 1 for a 2-dimensional vector inbound towards a fixed point on a 1-dimensional solar panel, meaning two different vectors will be able to produce the same inbound $0 - 90°$ angle and thus the same solar panel output when the light intensity is kept constant. Extending this to a 3-dimensional vector towards a 2-dimensional solar panel as can be seen in Figure 2 results in a conical shell of infinite possible attack vectors around the axis normal to the solar panel plane, which all will produce the same angle with the plane, and thus the same output.

When the angle of aperture of these infinite conical shells are large enough, they will intersect with each other, producing an intersection curve which runs along the conical shells, uniquely defined by the angle θ about the cones axis of revolution. The thought here is that we should be able to use this information to derive a common source for the light which produce these angles, in the form of a vector pointing towards this source. Figure 3 illustrates the concept in its most basic state, with two orthogonal, overlapping panels both experiencing a $45°$ inbound sunlight angle, resulting in two infinite orthogonal cones with overlapping origins, producing a single common vector which can induce the measured angles. However, when the combined angles with the planes falls short of $90°$, the cones will overlap and intersect in two lines instead of one, meaning that two different vectors can produce the same angles, like the single panel in the 1-dimensional case. And when the cones do not share a common origin, they will instead produce a parabolic intersection curve, rather than a line. In general terms; when two infinite orthogonal cones share a common origin (or share their axis of revolution with the axis of a common Cartesian coordinate system) together possess a total half aperture of $45°$ or more, they will intersect into each other and produce an intersection curve along their shell. This underlines an important aspect of this approach; the underdetermined nature of the model leave an ambiguous result in the two possible output vectors, and another input is needed to determine which is the real and which is the mirrored vector. This can be done by e.g. using a third panel in conjunction to produce another orthogonal cone, or as simple as a light dependent resistor as a binary switch.

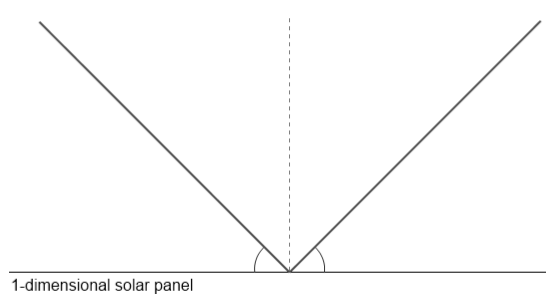

Figure 1: 2D vectors inbound on 1D panel, allowing two inbound vectors to produce the same angle and thus the same solar panel output.

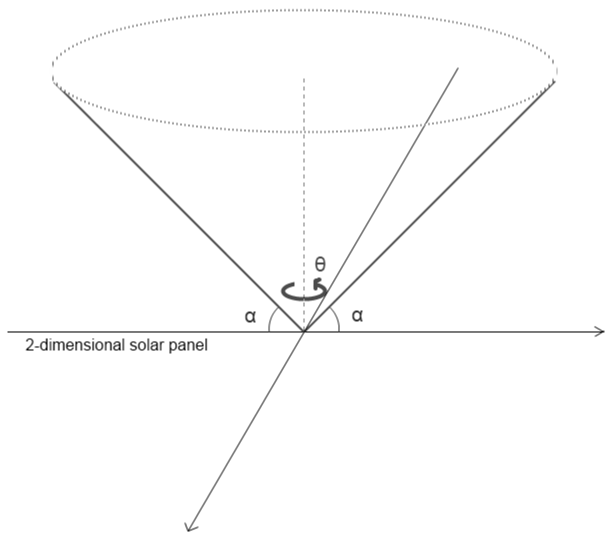

Figure 2: 3D vectors inbound on 2D panel, allowing an infinite number of inbound vectors to produce the same angle and thus the same solar panel output.

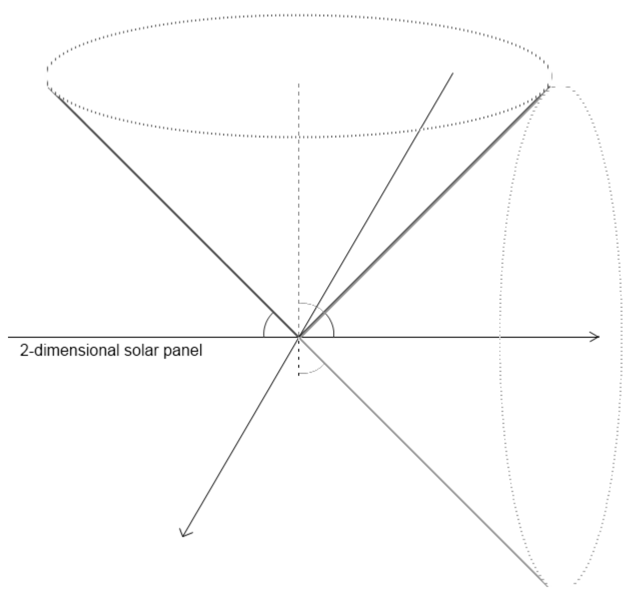

Figure 3: 3D vector inbound on two orthogonal 2D panels, producing a single intersection line representing the only inbound vector which can produce both values simultaneously.

With physically accurate origins

To express the problem mathematically, we need to note that our satellite spaces the opposing solar panels (in NUTS' case, roughly 5 and 10 cm) away from the center of the satellite, or the origin of the Body frame. Seeing as the nature of a solar panel requires the entire panel to receive light in order to produce the predicted current output, and that all inbound sunlight is parallel, we assume that the panels can be modeled as a single point somewhere along the axis of the Body frame. If we were to model the inbound sunlight angles in the form of infinite conical shells onto two of the solar panel surfaces (YZ and XZ plane chosen here for demonstration purposes), they would look like this in Body frame coordinates:

$$y^2 + z^2 = r_x^2 = (x - d_1)^2 \tan^2 \alpha_x \tag{1}$$
$$y = (x - d_1) \sin \theta_x \tan \alpha_x \tag{2}$$
$$z = (x - d_1) \cos \theta_x \tan \alpha_x \tag{3}$$

$$x^2 + z^2 = r_y^2 = (y - d_2)^2 \tan^2 \alpha_y \tag{4}$$
$$x = (y - d_2) \sin \theta_y \tan \alpha_y \tag{5}$$
$$z = (y - d_2) \cos \theta_y \tan \alpha_y \tag{6}$$

where d_1 and d_2 are abitrary distances along x and y axis respectively, representing the solar panels and the origins of the cones, α is the inbound sunlight angle producing the cone about the indicated subscript axis where θ is the angle about the indicated axis of revolution, normal to the plane. It can be noted that these cones produce symmetrical top and bottom parts, like proper mathematical cones do, but we will only draw the cones outward from these points ($x \geq d_1$, $y \geq d_2$ respectively), as the other (bottom) parts of the cones will expand into the satellite and be of no use to us. For the bottom parts of these cones, we would rather use the same approach on opposing solar panels where available.

By combining the equation for the cone about the x-axis (1) with the cone about the y-axis (4) we get the following equation:

$$\begin{gathered}(y - d_2)^2 \tan^2 \alpha_y - x^2 = (x - d_1)^2 \tan^2 \alpha_x - y^2 \\ (\tan^2 \alpha_y + 1) y^2 - 2d_2 \tan^2 \alpha_y y + d_2^2 \tan^2 \alpha_y = (\tan^2 \alpha_x + 1) x^2 - 2d_1 \tan^2 \alpha_x x + d_1^2 \tan^2 \alpha_x\end{gathered} \tag{7}$$

Substituting the expression for y from the cone about x (2) into the left side, we get:

$$\begin{aligned}(\tan^2 \alpha_y + 1)((x - d_1) \sin \theta_x \tan \alpha_x)^2 - 2d_2 \tan^2 \alpha_y ((x - d_1) \sin \theta_x \tan \alpha_x) + d_2^2 \tan^2 \alpha_y \\ = (\tan^2 \alpha_x + 1) x^2 - 2d_1 \tan^2 \alpha_x x + d_1^2 \tan^2 \alpha_x\end{aligned} \tag{8}$$

Expanding and collecting the terms yields

$$
\begin{aligned}
&\left((\tan^2\alpha_y+1)\sin^2\theta_x\tan^2\alpha_x-\tan^2\alpha_x-1\right)x^2\\
&\qquad+2\left(d_1\tan^2\alpha_x(1-(\tan^2\alpha_y+1)\sin^2\theta_x)-d_2\tan^2\alpha_y\tan\alpha_x\sin\theta_x\right)x\\
&+d_1^2\tan^2\alpha_x\left((\tan^2\alpha_y+1)\sin^2\alpha_x-1\right)+2d_1d_2\tan^2\alpha_y\tan\alpha_x\sin\theta_x+d_2\tan^2\alpha_y=0 \qquad (9)
\end{aligned}
$$

Which ties our variable of interest, θ_x, up against the distance along the cones axis of revolution, x. Both α_x and α_y have known values. As expected, the intersection curve of the two cone shells appear to be a parabola mapping values of θ_x into values of x. This holds for three dimensional case even if we are investigating the projection onto the single variable x, as each x, θ_x pair correspond to a single y and z value on the intersection curve along the cones. However, what is essential to note about this equation is that only the lower degrees of the polynomial are depending on the axial offset translations (d_1 and d_2) from the Body reference frame origin. Knowing that a parabolic function approaches a straight line when moving sufficiently far away from its vertex, we can investigate for which values of θ_x the function stops changing, by differentiating (9) twice with regard to x:

$$
f(x,\theta_x)=\psi(\theta_x)x^2+\beta(\theta_x)x+\gamma(\theta_x) \qquad (10)
$$

$$
\begin{aligned}
\frac{\delta^2}{\delta x^2}f(x,\theta_x)&=2\psi(\theta_x)\\
&=2\left((\tan^2\alpha_y+1)\sin^2\theta_x\tan^2\alpha_x-\tan^2\alpha_x-1\right) \qquad (11)
\end{aligned}
$$

Where ψ,β,γ in (10) replace the corresponding parts of the polynomial, for the sake of readability. We note that the expression is now only depending on a single variable, θ_x, and when setting (11) equal to zero we get

$$
\begin{aligned}
2\left((\tan^2\alpha_y+1)\sin^2\theta_x\tan^2\alpha_x-\tan^2\alpha_x-1\right)&=0\\
(\tan^2\alpha_y+1)\sin^2\theta_x\tan^2\alpha_x&=\tan^2\alpha_x+1\\
\sin^2\theta_x&=\frac{\tan^2\alpha_x+1}{\tan^2\alpha_x(\tan^2\alpha_y+1)}\\
\theta_x&=\sin^{-1}\left(\pm\sqrt{\frac{\tan^2\alpha_x+1}{\tan^2\alpha_x(\tan^2\alpha_y+1)}}\right) \qquad (12)
\end{aligned}
$$

where θ_x are the two symmetrical theta angles about x which the parabolic intersection curve along cones converges towards as it stretches out towards infinity. It can be noted that $\theta_{x1}=-\theta_{x2}$. The model can be seen illustrated as an example MATLAB plot in Figure 5 and 6.

With coinciding origins

As hinted onto earlier, what is interesting about this is that we can see that the equation for the intersecting θ_x when the cones extend towards infinity will be the exact same as if we were to set d_1 and d_2 equal to zero, effectively merging the centers/origins of the cones. The difference is that the latter model will not produce intersecting curves, but rather generate straight lines outward along the intersection, as long as the combined cone apertures are large enough, which can be seen easily by taking the same approach to modeling the cones:

$$y^2 + z^2 = r_x^2 = x^2 \tan^2 \alpha_x \tag{13}$$
$$y = x \sin\theta_x \tan\alpha_x \tag{14}$$
$$z = x \cos\theta_x \tan\alpha_x \tag{15}$$

$$x^2 + z^2 = r_y^2 = y^2 \tan^2 \alpha_y \tag{16}$$
$$x = y \sin\theta_y \tan\alpha_y \tag{17}$$
$$z = y \cos\theta_y \tan\alpha_y \tag{18}$$

Again, by combining the equation for the cone about the x-axis (13) with the cone about the y-axis (16) we get the following results:

$$\begin{aligned}
y^2 \tan^2 \alpha_y - x^2 &= x^2 \tan^2 \alpha_x - y^2 \\
y^2(\tan^2 \alpha_y + 1) &= x^2(\tan^2 \alpha_x + 1) \\
\left(\frac{y}{x}\right)^2 &= \frac{\tan^2 \alpha_x + 1}{\tan^2 \alpha_y + 1}
\end{aligned} \tag{19}$$

Using the expression for y from the cone about x (14), we can substitute the ratio between them into the left hand side and get

$$\begin{aligned}
(\sin\theta_x \tan\alpha_x)^2 &= \frac{\tan^2 \alpha_x + 1}{\tan^2 \alpha_y + 1} \\
\sin^2\theta_x &= \frac{\tan^2 \alpha_x + 1}{\tan\alpha_x(\tan^2 \alpha_y + 1)} \\
\theta_x &= \sin^{-1}\left(\pm\sqrt{\frac{\tan^2 \alpha_x + 1}{\tan^2 \alpha_x(\tan^2 \alpha_y + 1)}}\right)
\end{aligned} \tag{20}$$

The result is identical to the angle we found that the intersection between the displaced cones would converge to in (12). This can be interpreted as that when the cones have origins on the satellite Body axis, and $|x|, |y| \gg d_1, d_2$, it is safe to simplify the equation for the cones in such a way that they have overlapping origins in the center of the Body frame, which greatly reduce the complexity and computational load for determining the intersection curves. The result shown here holds for all axis of the Body frame, and it is safe to assume that we are not interested in the details of the intersection curves close to the satellite, as we are trying to pinpoint a source which is on average about 150 million kilometers away. The model can be seen illustrated as an example MATLAB plot in Figure 4.

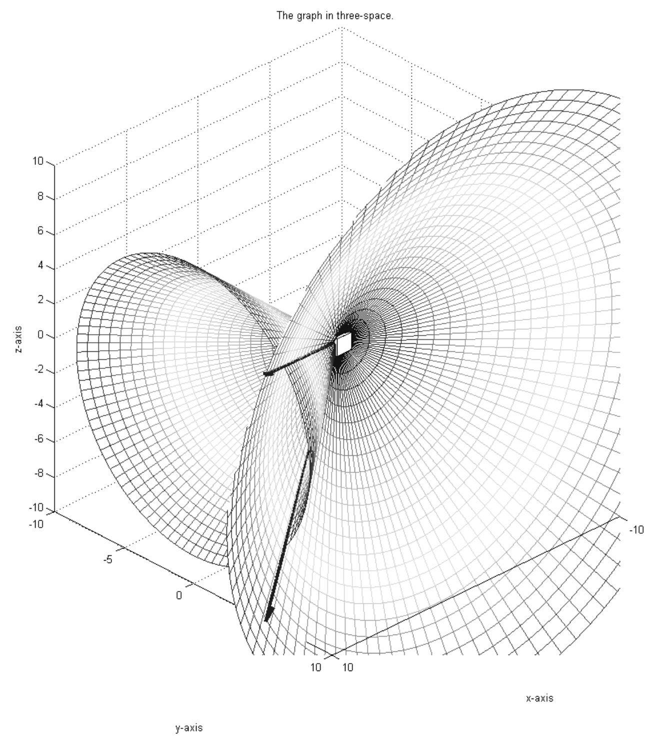

Figure 4: MATLAB: Orthogonal common origin cones produced by 30° and 50° angles of incidence. Intersection vectors represented by thick lines with arrowheads.

Sun reference model

In order to be able to utilize the measured body frame sun vector, we need to know the sun vector in orbit or NED frame in such a way that we can estimate the rotation between the two. A sun sensor reference model, relating the Sun's position to the satellite orbital position, is presented in the work of Rinnan [11], using the previous work of Svartveit [2] and Sunde [12]. The model assumes a simplification where the Sun is modeled to revolve around the Earth, making for a geocentric approach. This approach was demonstrated in the work of Kristiansen [13] to not introduce inaccuracies of noteworthy magnitude.

The elevation of the Sun varies periodically through the year and is given by:

$$\epsilon_{sun} = \frac{2\pi}{180}\sin(\frac{T_{sun}}{365}2\pi) \tag{21}$$

The parameter T_{sun} is the time in days since the Earth passed the vernal equinox, defined as the spring equinox as seen from the Northern Hemisphere.

The azimuth angle between the satellite and the Sun is given by:

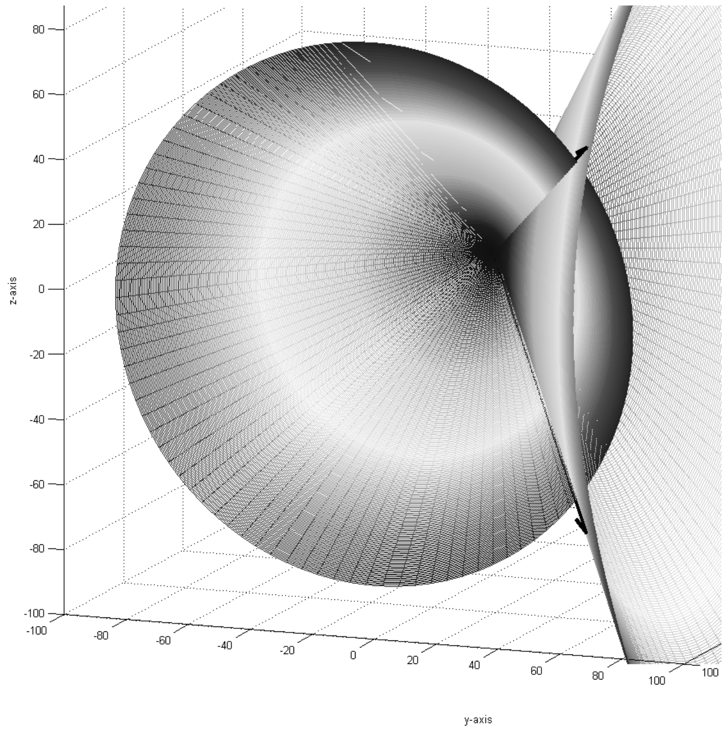

Figure 5: MATLAB: Far out view of orthogonal separate origin cones produced by 30° and 50° angles of incidence. Intersection vectors represented by thick lines with arrowheads.

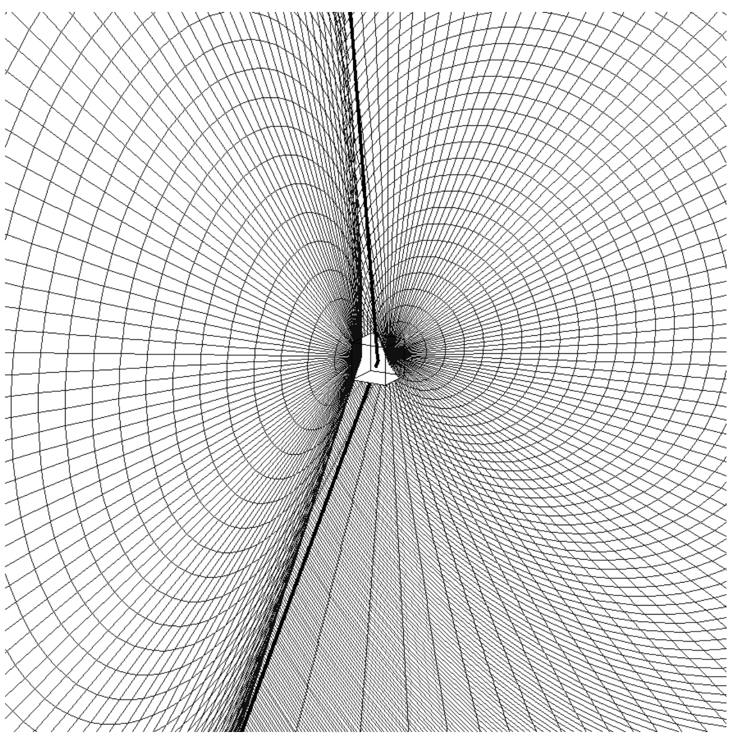

Figure 6: MATLAB: Close in view of orthogonal separate origin cones produced by 30° and 50° angles of incidence. Intersection vectors represented by thick lines, showing the convergence of the parabolic intersection curve.

$$\lambda_{sun} = \frac{T_{sun}}{365} 2\pi \tag{22}$$

The time-varying sun sensor reference vector to be used by methods such as the developed EQUEST or the nonlinear observer is:

$$\mathbf{r}_1 := \mathbf{R}_\theta(\epsilon_{sun})\mathbf{R}_\psi(\lambda_{sun})\mathbf{r}_1^0 \tag{23}$$

where $\mathbf{r}_1^0 = [1 \quad 0 \quad 0]^\top$ and the rotation matrices are expressed using Euler angles.

Preliminary disturbance suppression algorithm

For the model described to have any real value, we need a way to treat the vectors generated, for meaningful comparing, extracting, combining and discarding of data. This may include: Identifying indirect (reflected) sunlight information components, using this information to mathematically compensate for disturbance on panels which receive direct sunlight, or enforcing rules of expected behavior for a true, direct Sun vector.

A number of observations can be made to help the investigation of the possible content of an algorithm for separating relevant inbound sunlight information from disturbance.

It seems reasonable to expect a calculated sun vector to change in a continuous manner, that there should be no instant jump in the produced vector unless the line of sight with the sun is broken, or a disturbance suddenly appears. This underlines the beneficial value of keeping track of the history of observed sun vectors. From this history the rate of change in the orientation of the vector can also be calculated, which also could prove valuable to utilize for identifying disturbance.

The idea of that the change in sun vector orientation should be smooth can be manifested by identifying states consisting of combinations of active solar panels which the system should or should not be allow to transfer between. E.g. the system should not be able to produce sun vectors from output from one panel, then its opposing panel without registering output from one of the solar panels in between.

In an ideal, disturbance free scenario with solar panels on all six sides, we could formulate the problem as a system with 27 states in the form of a graph theory inspired cube; 6 faces, 8 vertices, 12 edges and 1 special case where there is absolutely no inbound sunlight. The faces of the cube represent a single active panel, the edge 2 active panels, and the vertices 3 active panels. Choosing only the three-panel states as vertices is a design choice given by the fact that these are obviously being the most commonly used state, as it does not require inbound sunlight to be parallel to any one solar panel plane. The other states may, for the sake of this representation, be thought of as transitions between vertexes, or edges. Let us name the panels x, y and z, given their respective positions on the satellite Body axis plane. As opposing panels can not receive sunlight from the same source simultaneously, we can assign them values to show which of the two are active; x = 1 for sunlight striking the panel on the positive side of the Body x-axis, -1 for the opposing one, and 0 for neither. Same with regard to y and z axis panels.

With these definitions, we can draw the described graph cube, as depicted in Figure 7, where the face [x, y, z] = [1, 0, 0] represents sunlight only striking the panel on the positive x-axis, and corresponding for all the other states made from combinations of x, y and z. $3^3 = 27$ is our total number of states, as expected, where only the 8 different 3-panel states are vertices in the model. What can be noted about this model, is that transitions between the states should only be allowed in a certain pattern; the system state can only move from one vertex to one of its three neighboring vertices, and it has to be through the edge connecting the two (equivalent to 1 Hamming distance if you will, somewhat paraphrased). Also, through a special case when only a single panel is active, represented by a face on the cube, it can move to any vertex connected by an edge on the same face. When no panels are active [0, 0, 0], a jump to any state can occur. This is equivalent to turning off the light, rotating the cube arbitrarily, and turning it back on again. For the sake of argument, such a rotation does not necessarily need to be arbitrary - one might keep track of the satellite rotation during the blackout and calculate where the sun vector is expected to be when it returns.

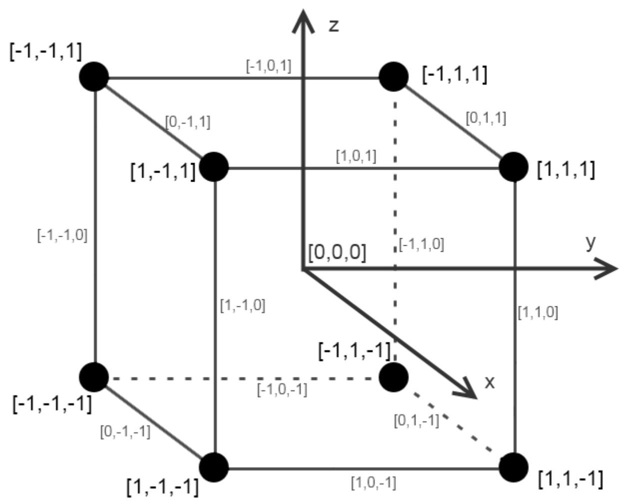

Figure 7: Visualization of graph inspired state cube.

To be able to separate unwanted information from the incoming sunlight, logic will have to be developed, like the idea that we can rule out incoming sunlight that strikes a solar panel on the opposite panel of where we know the sun to be, meaning if we know all three sides receiving direct sunlight, we can disregard light which hits the opposing three sides of the satellite entirely, or possibly calculate the effect of on a third panel by knowing the disturbance that hits two of them. Also, the disturbance hitting a panel which receives direct sunlight should generate additional current, tightening the cone of the panel in question, which means that two of the intersection vectors will be slightly off, but the redundancy of the system could possibly limit the effect the disturbance makes. In general, the aim will be to compare generated vectors with each other. In a disturbance free environment, the system should be able to provide three overlapping vectors, all being true Sun vectors, in addition to the false vectors which will all be pointing in distances mirrored by the edge between the producing panels, making none of these overlap. If in a scenario with three panels receiving direct sunlight, and one of them receives a disturbance, which means that it produces a tighter cone due to the higher current level, the panel can be identified from it's lack of ability to produce intersections with the other cones. Also, vectors produced by Earth albedo influence should have slower rates of change, as the disturbance comes from a wide area, covering an array of angles in the field of view compared to the single-spot origin of the Sun. Concepts like these will need to be expanded upon and combined into a logic system which can interpret and handle disturbance in a satisfying manner, by hopefully being able to calculate the amount of a polluted vector reading which is contributed by a source other than the Sun.

Final remarks

Conclusion

This paper shows the development of a mathematical model for representing sun vectors in the Body frame coordinate system, based on the intersections of conical shells generated by the inbound 3D light onto the 2D planes spanned by the satellite solar panels. The requirement for two panels to be able to generate intersecting cones is that the two combined half apertures of the cones, which are depending on the angle of inbound sunlight, are at least 45°. This means that the combined angles of inbound sunlight on the two solar panel planes can not exceed 90°, as there is no mathematical way for the two panels to experience a single-point, common light source at these angles.

The simplification of letting the spatially dislocated panels to be modeled as a point in the center of the satellite, the origin of the Orbit and Body coordinate frames, lets us model and calculate the angles about the Body frame axes which the intersection curves will converge against when propagating away from the satellite. This holds as long as we are not interested in the transient phase of the parabolic intersection curve, which happens close to the satellite. This intersection curve will approximate a dual sun vector inbound on the body frame origin, of which only one is true, while the other is a mirror image generated by the inherent ambiguity of the underdetermined system. Through valid combinations of solar panels, we can potentially calculate 8 different pairs of vectors to describe a single inbound light source on the satellite, all depending on which sides receive inbound sunlight.

Through enforcing rules of behavior of the Sun vector, we should be able to extract and combine information from a selection of panels, and use information from others to negate the impact of disturbance, or selectively discard information which we know to not carry Sun vector information. Ensuring that observed Sun vector orientation is continuous and, if possible, aligned with measured rotation from gyroscopes during eclipses, can make our system less susceptible to introduced disturbances and even help identify them to use this information to negate their impact.

Future work

Whether or not this mathematical model holds up in physical testing remains to see, the first round of preliminary testing is laid out and scheduled for February 2013, along other experiments on the NUTS hardware. If the first proof of concept is successful, then extended tests will be planned and carried out shortly after.

If the necessary testing in a disturbance free environment proves successful enough for the model to be able to identify isolated single-point light sources, then the work of establishing a proper disturbance rejecting algorithm begins. A general scheme needs to be carefully laid out, and corner-cases which can confuse the algorithm identified.

It should also be mentioned that the actual implementation of the model into the physical system has yet to be given excessive thought, so in future development one might run into unforeseen problems such as sample rate limits, logical corner cases or problems with the actual signals we are trying to measure due to the load on the batteries/EPS because of other subsystems drawing variable power. Also, in it's most basic state, this model will be sensitive to deviations from ideal solar panel characteristics, like intensity saturation level and non-sinusoidal behavior of current to incidence angle relation, which most likely will happen at excessively acute angles. These are important issues that will have to be attended to during physical testing.

References

[1] J.T. Gravdahl, E. Eide, A.Skavhaug, K. Svartveit, K. M. Fauske and F. M. Indergaard: Three axis Attitude Determination and Control System for a picosatellite: Design and implementation, Proceedings of the 54th International Astronautical Congress, Bremen, Germany, 2003

[2] K. Svartveit, Attitude determination of the ncube satellite, Masters thesis, NTNU, 2003.

[3] K. Jensen, K.H. Yabar and J.T. Gravdahl: A comparison of attitude determination methods: theory and experiments, In proceedings of the 62nd International Astronautical Congress, Cape Town, South Africa, 3 - 7 October, 2011

[4] T.B. Rinnan, R. Birkeland, J.T. Gravdahl: Development on the EQUEST method for attitude determination, Presented at the Fourth European CubeSat Symposium, Ecole Royale Militaire, Brussels, Belgium, 30 January - 1 February, 2012

[5] F.S. Holberg, R. Birkeland, J.T. Gravdahl: Implementation of three axis attitude determination and control system for a double cubesat, Presented at the Fourth European CubeSat Symposium, Ecole Royale Militaire, Brussels, Belgium, 30 January - 1 February, 2012

[6] S.S. Ose, Attitude determination for the norwegian student satellite ncube, Master's thesis, NTNU, 2004.

[7] J. Rohde, Kalman filter for attitude determination of student satellite, Master's thesis, NTNU, 2007.

[8] M.L. Psiaki, Attitude-determination filtering via extended quaternion estimation, Journal of Guidance, Control, and Dynamics, vol. 23, no. 2, pp. 206-214, Mar.-Apr 2000.

[9] F.L. Markley and D. Mortari, Quaternion attitude estimation using vector observations, The Journal of the Astronautical Sciences, vol. 48, no. 2-3, pp. 359-380, Apr.-Sep. 2000.

[10] K.L. Jenssen and K. H. Yabar, Development, implementation and testing of two attitude estimation methods for cube satellites, Master's thesis, NTNU, 2011.

[11] T.B. Rinnan, Development and comparison of estimation methods for attitude determination, Master's thesis, NTNU, 2012.

[12] B. O. Sunde, Sensor modelling and attitude determination for micro-satellite, Master's thesis, NTNU, 2005.

[13] R. Kristiansen, Attitude control of mini satellite, Master's thesis, NTNU, 2000.