

ADAPTIVE NEURAL STAR TRACKER CALIBRATION FOR PRECISION SPACECRAFT POINTING AND TRACKING

David S. Bayard
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
Tel: (818) 354-8208
bayard@bert2.jpl.nasa.gov

ABSTRACT

The star tracker is an essential sensor for precision pointing and tracking in most 3-axis stabilized spacecraft. In the interest of improving pointing performance by taking advantage of dramatic increases in flight computer power and memory anticipated over the next decade, this paper investigates the use of a neural net for adaptive in-flight calibration of the star tracker. Estimation strategies are given for cases when the spacecraft attitude is both known and unknown. As an example, a simulation study is given for which low spatial frequency distortions on the order of .1 pixel, are reduced to .034 pixel using the proposed method. Improvement in measurement quality obtained by such star tracker corrections are important since they translate directly into improved on-board estimates of position and rate, and to improved pointing and tracking capability.

1 INTRODUCTION

The star tracker unit (STU) is a key sensor for determining the overall pointing and tracking performance of most 3-axis stabilized spacecraft [4][5][9][11]. Any improvement in star tracker measurement quality translates directly into improved position and rate estimation, and hence improved pointing knowledge. Tracker sensing becomes even more important in the absence of accurate gyros, since the body rate must be reconstructed analytically. Such gyroless configurations are becoming popular in NASA and in commercial space industry to save power, weight, and overall cost [1]. This means that the design of future planetary flyby missions such as the Pluto Express [3], will be particularly challenging since accurate rate knowledge is essential for image motion compensation during flyby, but must be reconstructed accurately without the benefit of quality gyros.

The emergence of smaller, lighter, and cheaper spacecraft in planetary exploration calls for intelligent adaptive control systems that will maintain and improve upon the quality of science which is already possible with larger spacecraft. Fortunately, flight computer memory and power will be increasing at a dramatic rate over the next decade. This added

computational capability will enable the implementation of more sophisticated algorithms and software, which in turn will help to overcome the performance penalties associated with miniaturization and reduced mass/power configurations.

In the interest of improving performance by leveraging emerging computational power, this paper will investigate the use of a neural net for in-flight calibration of the star tracker. Specifically, a nonlinear distortion profile will be estimated autonomously based on in-flight measurements. The star tracker distortion is then corrected using the estimated profile. Preliminary results indicate that low spatial frequency distortions on the order of .1 pixel, can be reduced to .034 pixel using the proposed method.

2 STAR TRACKER MODEL

The star tracker, sometimes conveniently referred to as a Star Tracker Unit (STU), is an instrument which determines attitude by,

- Acquiring images of stars on a charge coupled device (CCD)
- Performing a pattern recognition of star constellations in the image
- Using an on-board star catalogue to provide the location of the identified stars with respect to an inertial frame of reference.

The location of 2 or more star images on the CCD along with their star locations in inertial coordinates is sufficient to determine the attitude of the STU camera with respect to an inertial frame of reference. The spacecraft attitude can then be calculated from knowledge of how the STU camera is mounted on the spacecraft body. These relationships will be developed in more detail in this section.

The standard vector crossproduct operator $v \times$, where $v = [v_1, v_2, v_3]^T \in R^3$ will often be represented as v^\times using the equivalent matrix notation,

$$v^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (1)$$

Typically, the STU images several stars simultaneously in the same exposure. Let $s_{k\ell}$, $\ell = 1, \dots, n_k$ denote the collection of stars imaged on the CCD during the exposure at time t_k . Let the unit vector $v_{k\ell}$ denote the location of star $s_{k\ell}$ in inertial coordinates. The vector $v_{k\ell}$ is rotated into the STU frame using the sequence of transformations,

$$z_{k\ell} = T A_k v_{k\ell} \quad (2)$$

Here, A_k is the attitude matrix at time t_k which acts to rotate a vector from inertial coordinates into the body frame; T is a matrix which rotates a vector from the body frame into the STU frame; and the unit vector z_{kl} is the location of star s_{kl} in STU coordinates.

The attitude matrix A_k is time dependent since the spacecraft body rotates with respect to the inertial frame. Nominally, the transformation matrix T is not time dependent since it is assumed that the Star tracker is bolted down to the spacecraft. However, slow variations are to be expected in T due to thermal effects, mechanical stability, etc., so that the time invariance assumption is only a convenient approximation.

Because of uncertainty in the attitude, the matrix A_k is more conveniently decomposed as,

$$A_k = \delta A(\phi_k) \bar{A}_k \quad (3)$$

where \bar{A}_k is a known estimate of the attitude at time t_k , and $\delta A(\phi_k)$ is the error parametrized by three Euler angles $\phi_k = [\phi^1, \phi^2, \phi^3]^T$, nominally taken to be in a 1-2-3 rotation sequence. Since the error δA_k is typically a small rotation, it can be expanded to first order as,

$$\delta A(\phi_k) \simeq I - \phi_k^\times \quad (4)$$

Similarly, only a nominal estimate \bar{T} of the transformation T is usually known, so that one can write,

$$T = \delta T(\theta) \bar{T} \quad (5)$$

where $\delta T(\theta)$ is the error parametrized by three Euler angles $\theta = [\theta^1, \theta^2, \theta^3]^T$. Since the error $\delta T(\theta)$ is typically a small rotation, it can be expanded to first order as,

$$\delta T(\theta) \simeq I - \theta^\times \quad (6)$$

Substituting (3)(4) and (5)(6) into (2) and eliminating terms of second order gives (see Appendix A),

$$z_{kl} = \hat{z}_{kl} + \hat{z}_{kl}^\times (\theta + \bar{T} \phi_k) \quad (7)$$

where,

$$\hat{z}_{kl} \triangleq \bar{T} \bar{A}_k v_{kl} \quad (8)$$

At this point, the vector z_{kl} denotes the unit vector in the STU frame pointing to star s_{kl} . Let the components of vector z_{kl} be denoted as,

$$z_{kl} = [z_{kl}^1, z_{kl}^2, z_{kl}^3]^T \quad (9)$$

Then the projection of z_{kl} onto the STU image plane places the image of star s_{kl} at location (x_{kl}, y_{kl}) on the CCD where,

$$x_{kl} = z_{kl}^1 / z_{kl}^3 \quad (10)$$

$$y_{kl} = z_{kl}^2 / z_{kl}^3 \quad (11)$$

Implicit in (10)(11) is the standard assumption that the star location has been normalized to unit focal length (i.e., given the true star location (X, Y) one defines $x = X/f$ and $y = Y/f$ where f is the camera focal length).

The star image is recorded on a CCD and the star location is determined by a centroiding algorithm applied to a pixel readout. To improve centroiding accuracy, the star image is typically defocused to spread photons over several pixels of the CCD. Due to sensor errors, the star is observed at perturbed location $(x'_{k\ell}, y'_{k\ell})$ where,

$$x'_{k\ell} = x_{k\ell} + F_1(x_{k\ell}, y_{k\ell}, p_{k\ell}) + \eta_{k\ell}^1 \quad (12)$$

$$y'_{k\ell} = y_{k\ell} + F_2(x_{k\ell}, y_{k\ell}, p_{k\ell}) + \eta_{k\ell}^2 \quad (13)$$

The quantities $\eta_{k\ell}^1, \eta_{k\ell}^2$ are additive white measurement noise terms which model the contribution of random errors in the determination of star locations. These are typically described in terms of a noise equivalent angle (NEA) in units of radians, which is typically a function of the noise-to-signal (N/S) ratio at starlit pixel locations. The N/S ratio is dependent on star magnitude, quantum efficiency, optical efficiency, integration time and system temporal noise levels. Sources of temporal noise include CCD noise (on-chip), electronics noise (off-chip), quantization noise, background noise, and dark current.

The quantities F_1, F_2 in (12)(13) are distortion functions which model the contribution of bias errors associated with determining star locations. These bias errors can be roughly divided into two groups, depending upon whether the distortion is of low or high spatial frequency.

Low spatial frequency error sources typically include,

- Optical distortion
- Chromatic distortion
- Mechanical stability/shifts
- Thermal expansion of materials
- Thermal sensitivity of refractive index
- Ground calibration accuracy

High spatial frequency error sources typically include,

- CCD photo response nonuniformity
- CCD dark current nonuniformity
- CCD charge transfer efficiency

- Centroiding algorithm error

These dependencies have been captured in the distortion model (12)(13) by letting the functions F_1, F_2 also depend on a “covariate” vector $p_{k\ell} = [p_{k\ell}^1, p_{k\ell}^2, p_{k\ell}^3]^T \in R^3$ whose components are given as follows,

$p_{k\ell}^1$: temperature (or color) of star $s_{k\ell}$.

$p_{k\ell}^2$: CCD temperature at time t_k

$p_{k\ell}^3$: optics temperature at time t_k

Star temperature (or color) is known from the star catalogue, while CCD and optics temperatures must be measured physically using temperature sensing devices.

3 AUTONOMOUS CALIBRATION

The specific approach to calibration depends on whether the spacecraft attitude is known or unknown.

3.1 Case 1: Attitude Known (i.e., $\phi_k \simeq 0$)

In some cases, an estimate \bar{A}_k of the spacecraft attitude A_k is available from alternative sources which is sufficiently accurate for star tracker calibration purposes. This includes cases where there are other sensors on board which provide attitude information, or when there are sufficient numbers of stars being observed in each STU exposure to provide accurate attitude information despite individual star location errors. (Star locations typically have a 2-D Poisson distribution across the field-of-view which acts to randomize the bias contributions from individual stars and hence average out as the number of stars gets large).

In this case, errors due to attitude knowledge are removed and the STU alignment error is estimated as part of the overall distortion profile.

3.2 Case 2: Bootstrapped Attitude Estimate

Often the attitude matrix A_k must be estimated at each time t_k from the same data that is being used to calibrate the star tracker. This problem is much more difficult than Case 1, since the STU frame alignment error $\delta T(\theta)$ is not generally distinguishable from the attitude error $\delta A(\phi_k)$. This issue has been addressed in Shuster and Lopes [7]. Their resolution of the problem is to use a parametrization of F_1 and F_2 which explicitly enforces a zero rotation.

Hence in the single star tracker case, the STU frame alignment error θ is defined to be zero, and the actual physical rotation associated with the STU misalignment is absorbed into the attitude estimation error.

In the present paper, a different approach will be taken compared to [7]. Here, it will be assumed that $F_1(x, y, p) \simeq 0$ and $F_2(x, y, p) \simeq 0$ in some neighborhood of the CCD origin $N_0(x, y) = \{(x, y) : x^2 + y^2 \leq \gamma^2\}$. Hence, any star whose predicted location on the CCD is in neighborhood $N_0(x, y)$ is assumed to be *completely undistorted*. For such stars, any discrepancy between the predicted star location and actual measured star location will be understood to be pure attitude error and used to update the attitude estimate \bar{A}_k . This provides essentially two-axis attitude information, with the STU boresight axis still largely unresolved. After correcting all stars on the FOV at t_k by the two-axis attitude rotation, a pure boresight axis rotation can be fit to the corrected star locations and used to update the attitude estimate further. The star locations are corrected one last time for this boresight rotation, and the remaining errors are used for estimating the distortion profiles F_1 and F_2 . This complete procedure is applied at each time t_k for which there are stars within neighborhood $N_0(x, y)$.

Over time, a sufficient amount of data of the desired form can be collected to provide estimates of the distortion profiles. Of course, the data collection process can always be made more efficient by using special preplanned attitude maneuver sequences which maintain stars in N_0 .

4 NEURAL NET ARCHITECTURE

An important property of neural networks is their capability for function approximation. While many different neural net architectures can be applied to the present problem, a particularly simple method denoted as the General Regression Neural Network (GRNN) due to Specht [8] will be used.

4.1 General Regression Neural Network (GRNN)

The GRNN approach of Specht [8] will be discussed here and motivated for the calibration application. The main advantage of the GRNN is that it converges asymptotically to the conditional mean estimate of the distortion surface as the number of data samples becomes large. This convergence property is essential when working with noisy data.

Let $X \in R^p$, and $y \in R^1$ be random variables. Given X it is desired to estimate y . The best estimate of y in the sense of minimizing the mean-square estimation error is known to be the conditional mean,

$$E[y|X] = \int_{-\infty}^{\infty} y f(y|X) dy \quad (14)$$

Unfortunately, in many applications such as the star tracker calibration problem the conditional probability density $f(y|X)$ in (14) is not known. However, if the joint density function $f(X, y)$ is known, the conditional mean (14) can be equivalently calculated from,

$$E[y|X] = \frac{\int_{-\infty}^{\infty} y f(X, y) dy}{\int_{-\infty}^{\infty} f(X, y) dy} \quad (15)$$

where use has been made of the relation $f(y|X) = f(X, y)/f(X)$.

The joint density $f(X, y)$ in (15) can be estimated from sample values $\{X^i, Y^i\}_{i=1}^n$ of the random variables x and y using any one of a large number of approaches [10]. This discussion will focus on a class of consistent estimators proposed by Parzen [6], extended to the multivariate case by Cacoullos [2], and made into a neural network by Specht [8]. The form of the joint density estimator is given as follows,

$$\hat{f}(X, Y) = \frac{1}{(2\pi)^{(p+1)/2} \sigma^{p+1}} \cdot \frac{1}{n} \sum_{i=1}^n \exp\left(-\frac{\|X - X^i\|^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{|Y - Y^i|^2}{2\sigma^2}\right) \quad (16)$$

The large sample behavior of this estimator is of interest. It is known that the estimator $\hat{f}(X, Y)$ will provide an asymptotically unbiased and consistent estimate of $f(X, y)$ if $f(X, y)$ is continuous, and if the variance parameter σ^2 of the Gaussian pulses in (16) is chosen as a decreasing function of the data length n such that [2][10][8],

$$\lim_{n \rightarrow \infty} \sigma_n = 0 \quad (17)$$

$$\lim_{n \rightarrow \infty} n\sigma_n^p = \infty \quad (18)$$

As an example, one suitable choice is,

$$\sigma_n = n^{-\frac{1}{p+q}} \quad (19)$$

for any $q > 0$.

Substituting the joint density estimate (16) into the expression for the conditional mean (15) and performing the integrations analytically gives the simplified expression [8],

General Regression Neural Network (GRNN)

$$\hat{Y}(X) = \sum_{i=1}^n \gamma_i(X) Y^i \quad (20)$$

where the weights are calculated from,

$$\gamma_i(X) = \frac{\exp\left(-\frac{\|X - X^i\|^2}{2\sigma^2}\right)}{\sum_{i=1}^n \exp\left(-\frac{\|X - X^i\|^2}{2\sigma^2}\right)} \quad (21)$$

If y is a vector, a similar estimator can be used for each component of y .

For a finite sample size, the choice of σ determines the smoothness of the GRNN estimator (20)(21). As σ becomes very large, $\hat{Y}(X)$ simply becomes the sample mean $\frac{1}{n} \sum_{i=1}^n Y^i$. As σ goes to zero, $\hat{Y}(X)$ takes the value of Y^i associated with the observation closest to X . For intermediate values of σ , $\hat{Y}(X)$ is a weighted average of all values of Y^i with particular emphasis on those having points lying closest to X .

Advantages of the GRNN relative to other neural network approaches are summarized as follows [8],

- Since $\sum_{i=1}^n \gamma_i = 1$ the estimate is a convex combination of all measurements Y_i , implying that the estimate is bounded to the range of the observed samples.
- Equation (21) provides an analytic expression for weights. Local minima are avoided since the neural net does not have to be tuned using gradient type methods.
- As the size of the data set n becomes large and σ_n is chosen according the rules (17)(18), the neural net provides an asymptotically unbiased and consistent estimate of the conditional mean distortion surface.

A disadvantage of the GRNN is that the complexity of the expressions (20)(21) increases with n . However, methods to overcome this using clustering algorithms have been investigated in Specht [8]. Alternatively, special purpose hardware architectures can be developed for efficient implementation.

4.2 Application to Tracker Calibration

Consider the GRNN applied to the star tracker calibration problem. Since it is desired to “invert” the distortion profile, the neural net must be set up to learn the appropriate inverse mapping. For this purpose, the GRNN *input* is chosen as the STU *output*,

$$X^{k\ell} = [x'_{k\ell}, y'_{k\ell}, p_{k\ell}^1, p_{k\ell}^2, p_{k\ell}^3]^T \quad (22)$$

In addition to $(x'_{k\ell}, y'_{k\ell})$, the covariate vector $p_{k\ell}$ has been included in (22) to capture dependencies on star, CCD and optics temperatures.

While in principle the STU *input* $\hat{z}_{k\ell}$ can be taken as the neural net *output*, a better choice is to first project it into the STU image plane as follows,

$$\hat{x}_{k\ell} = \hat{z}_{k\ell}^1 / \hat{z}_{k\ell}^3 \quad (23)$$

$$\hat{y}_{k\ell} = \hat{z}_{k\ell}^2 / \hat{z}_{k\ell}^3 \quad (24)$$

The use of the vector $[\hat{x}_{k\ell}, \hat{y}_{k\ell}]^T \in R^2$ rather than $\hat{z}_{k\ell} \in R^3$ reduces the dimension by one. Furthermore, it is best to estimate the small error profile $(x' - \hat{x}, y' - \hat{y})$ which is typically a

fraction of a pixel, rather than the larger quantity (\hat{x}, \hat{y}) which can vary hundreds of pixels across the entire FOV. This is because in the latter approach, the error profile would have to be reconstructed by subtracting two large numbers, which is generally undesirable using finite precision arithmetic. For this reason, the neural net output is defined in terms of the error profile,

$$Y_1^{k\ell} = x'_{k\ell} - \hat{x}_{k\ell} \quad (25)$$

$$Y_2^{k\ell} = y'_{k\ell} - \hat{y}_{k\ell} \quad (26)$$

Since there are two outputs, and the GRNN methodology must be applied to each output Y_1 and Y_2 separately.

At any point in time the neural net learning can be frozen, and the STU distortion corrected by inverting the mapping as follows,

$$\hat{x}_{k\ell} = x'_{k\ell} - E[Y_1^{k\ell} | X^{k\ell}] \quad (27)$$

$$\hat{y}_{k\ell} = y'_{k\ell} - E[Y_2^{k\ell} | X^{k\ell}] \quad (28)$$

where the GRNN is used to produce the required conditional mean estimates.

5 NUMERICAL EXAMPLE

A numerical example is given to demonstrate the neural net approach. For this example, the star tracker is chosen with an 8 degree FOV, and a CCD with a 512x512 pixel array. It is assumed that the attitude is known perfectly at each time instant, that there are 15 stars per exposure frame (as would be consistent with a full frame device), and there are 300 frames. Assuming a frame rate of 1 second, this would involve 300 seconds of data. This gives a total of 4500 stars for calibration purposes. The stars from each exposure are assumed to be Poisson distributed across the FOV.

The STU distortion pattern (in the x direction, F_1) is shown in Figure 1. This distortion is spatially oscillatory and attains a maximum magnitude of .1 pixels. This shape is typical of optical distortion patterns remaining after initial ground calibration and captures the low spatial frequency behavior of the error. High frequency distortions are also important but are ignored in this study. Also, the dependencies on covariates $p_{k\ell}$ (i.e., star temperature, CCD temperature and optics temperature) are ignored for simplicity.

Given the star location data, the neural net (20)(21) with input vector (22) and output (25) (26) is used to estimate the distortion surface. The result is shown in Figure 2, using a value of $\sigma = .042$ rad. The smoothness of the estimate is clearly seen from Figure 2, as is its resemblance to the true surface of Figure 1. The residual distortion after correction is shown in Figure 3, and has a worst-case error of .051 pixels. Hence, the effect of the neural net in this example is to reduce the the worst-case error from .1 pixels to .051 pixels.

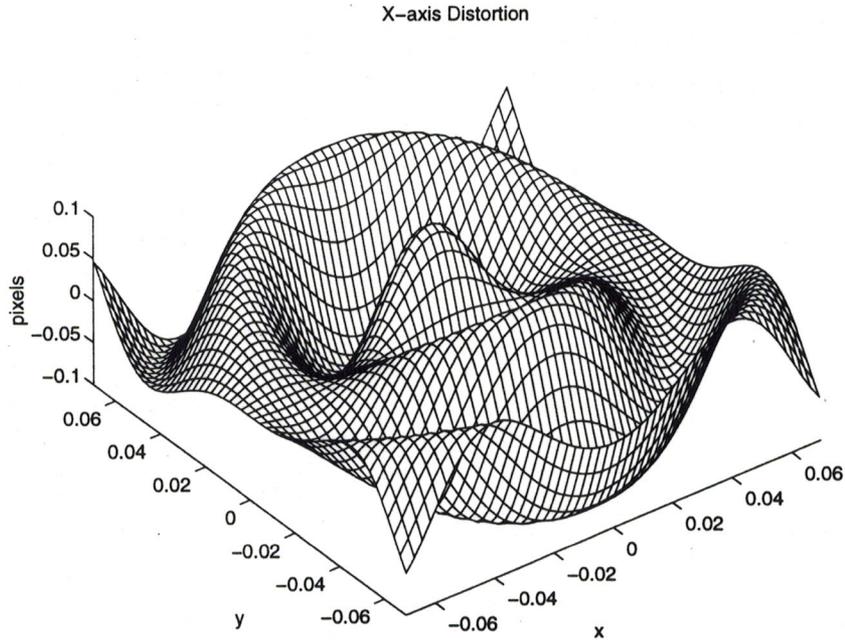


Figure 1: STU distortion pattern of .1 pixels (worst case) for x-axis

As mentioned in Section 4, the neural net performance is sensitive to the choice of σ . This sensitivity is briefly studied by changing the value to $\sigma = .021$ and repeating the previous run. The resulting neural net estimate is shown in Figure 4. As expected, there is less smoothing and the estimate is more “ragged” than the earlier estimate of Figure 2. The residual error distortion after correction is shown in Figure 5, and has a worst-case error of .034 pixels compared to the initial error of .1 pixels. Comparing .034 pixels error with the previous run having .051 pixels error, indicates that the estimation process has been improved by using less smoothing.

In summary, the numerical example indicates that the neural net can be used to reduce error from .1 pixel to between .051 and .034 pixels. This example assumes that the attitude is known at each time. The method of bootstrapping attitude while calibrating the star tracker discussed in Section 3, remains to be tested by simulation. It is expected that comparable results will require larger data sets in order to “smooth” over additional attitude estimation errors.

X-axis Distortion Estimate

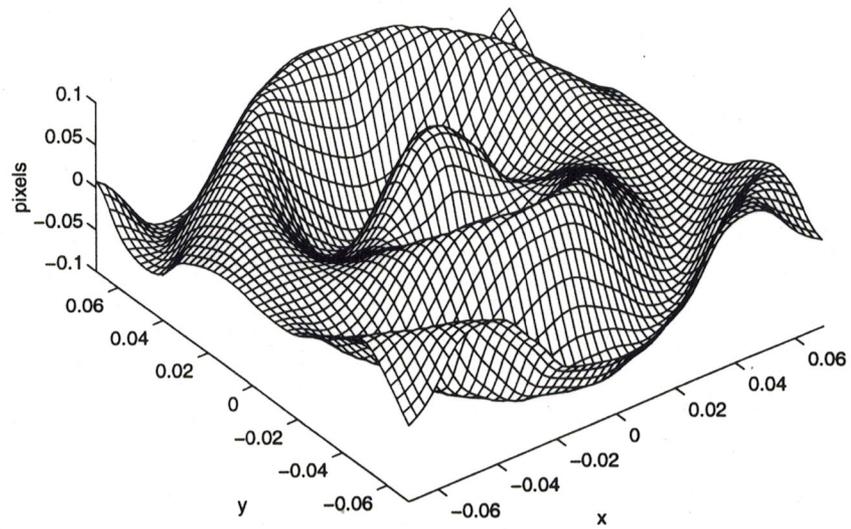


Figure 2: Neural net estimate of X-axis distortion; $n = 4500$, $\sigma = .042$

X-axis Corrected Distortion

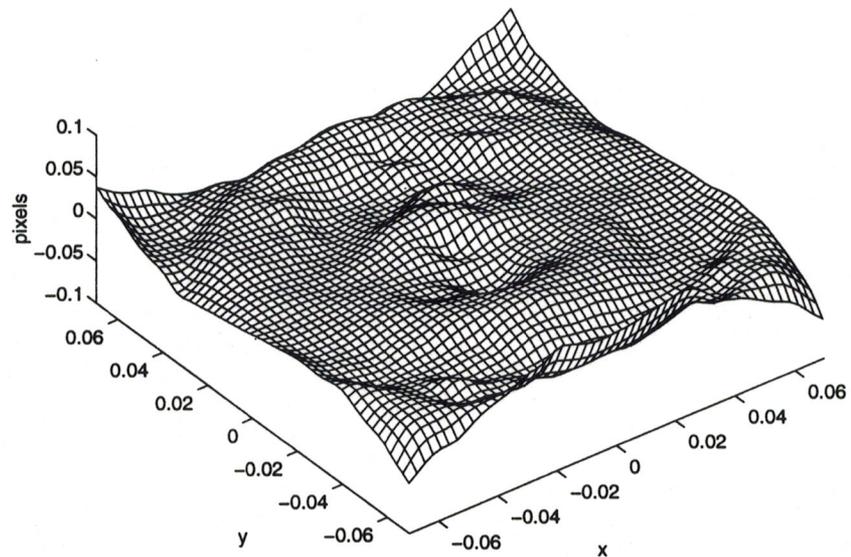


Figure 3: Residual X-axis distortion of .051 pixels (worst case) after neural net calibration; $n = 4500$, $\sigma = .042$

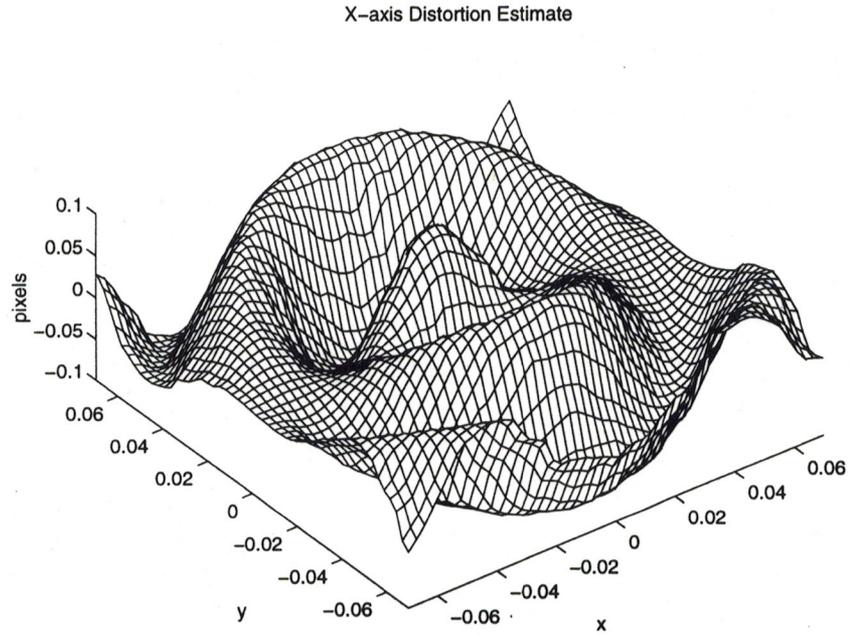


Figure 4: Neural net estimate of X-axis distortion; $n = 4500$, $\sigma = .021$

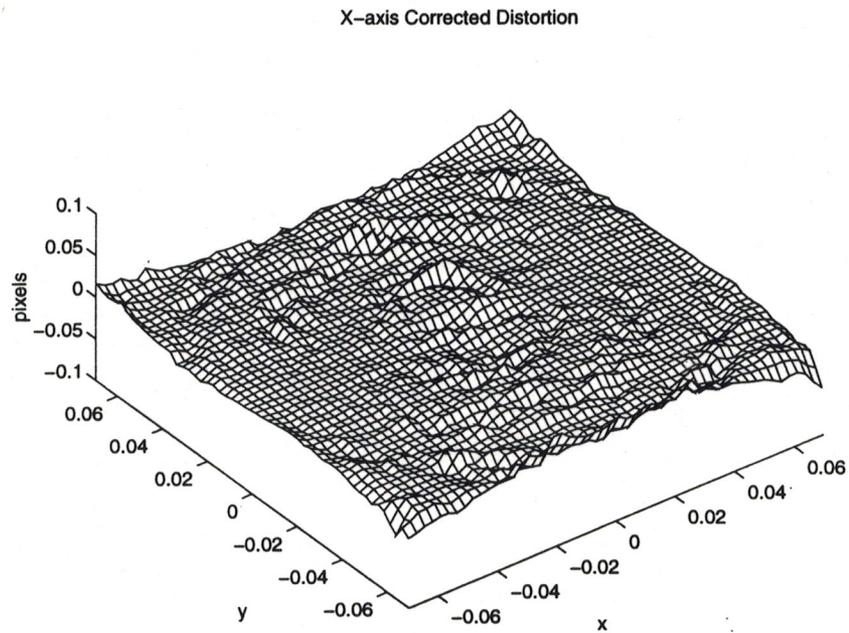


Figure 5: Residual X-axis distortion of .034 pixels (worst case) after neural net calibration; $n = 4500$, $\sigma = .021$

6 CONCLUSIONS

This paper demonstrates how a neural net can be used for in-flight star tracker calibration. Specifically, a nonlinear distortion mapping is estimated autonomously by a neural net using in-flight measurements. The star tracker distortion is then corrected by using the learned mapping. One of the features of the approach is in the choice of neural net, which ensures a consistent estimate of the conditional mean. This is important for calibration purposes since many noise sources enter into the problem.

While the computation and memory requirements for implementing this approach are excessive for most present day spacecraft, they are perfectly reasonable for future spacecraft given the increases in flight computer capacity expected over the next decade.

Acknowledgements

The author would like to thank Dr. Jim Alexander and Dan Chang of the Automation and Control Section of JPL for several technical discussions and interactions. This research was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

A APPENDIX A:

In this appendix, details are given behind the derivation of equation (7).

Substituting (3)(4) and (5)(6) into (2) gives,

$$z_{kl} = T A_k v_{kl} = \delta T(\theta) \bar{T} \delta A(\phi_k) \bar{A}_k v_{kl} \quad (29)$$

$$= (I - \theta^\times) \bar{T} (I - \phi_k^\times) \bar{A}_k v_{kl} \quad (30)$$

Expanding (30) and neglecting second order terms gives,

$$z_{kl} = \bar{T} \bar{A}_k v_{kl} - \theta^\times \bar{T} \bar{A}_k v_{kl} - \bar{T} \phi_k^\times \bar{A}_k v_{kl} \quad (31)$$

$$= \hat{z}_{kl} - \theta^\times \hat{z}_{kl} - \bar{T} \phi_k^\times \bar{T}^T \hat{z}_{kl} \quad (32)$$

$$= \hat{z}_{kl} - \theta^\times \hat{z}_{kl} - (\bar{T} \phi_k)^\times \hat{z}_{kl} \quad (33)$$

$$= \hat{z}_{kl} + \hat{z}_{kl}^\times (\theta + \bar{T} \phi_k) \quad (34)$$

Here, equation (32) follows from (31) by using the definition of \hat{z}_{kl} from (8), and the relation $\bar{T}^T \bar{T} = I$; equation (33) follows by using the relation $\bar{T} \phi_k^\times \bar{T}^T = (\bar{T} \phi_k)^\times$ which is a direct consequence of Identity A.1 below; and equation (34) follows by the crossproduct relation $u \times v = -v \times u$ interpreted in matrix notation.

Identity A.1 For any rotation matrix R (i.e., any matrix $R \in R^{3 \times 3}$ such that $R^T R = R R^T = I$ and $\det R = 1$), and any vector $u \in R^3$ the following equality holds,

$$R u^\times R^T = (R u)^\times \quad (35)$$

Proof: Given vectors $u, v \in R^3$, let

$$w = u \times v \triangleq u^\times v \quad (36)$$

For any rotation vector R , one has,

$$R w = R(u \times v) = (R u) \times (R v) = (R u)^\times R v \quad (37)$$

or solving for w gives,

$$w = R^T (R u)^\times R v \quad (38)$$

Equating (36) and (38) gives,

$$R^T (R u)^\times R v = u^\times v \quad (39)$$

Since v is arbitrary, this implies,

$$R^T (R u)^\times R = u^\times \quad (40)$$

Rearranging (40) gives the desired result (35).

References

- [1] M.S. Grewal and M. Shiva, "Application of Kalman Filtering to gyroless attitude determination and control system for environmental satellites," Proc. IEEE Conference on Decision and Control, Buena Vista, Florida, December 1995.
- [2] T. Cacoullos, "Estimation of a multivariate density," Ann. Inst. Statist. Math, (Tokyo), vol. 18, no. 2, pp. 179-189, 1966.
- [3] G.J. Kissel, "Precision pointing for the Pluto mission spacecraft," 18th Annual AAS Guidance and Control Conference, Paper AAS 95-065, Keystone, Colorado, February 1995.
- [4] C.C. Liebe, "Pattern recognition of star constellation for spacecraft applications," IEEE Aerospace & Electronics Systems Magazine, pp. 31-39, Jan. 1993.
- [5] C.C. Liebe, "Star trackers for attitude determination," IEEE Aerospace & Electronics Systems Magazine, pp. 10-16, June 1995.
- [6] E. Parzen, "On estimation of a probability density function and mode," Ann. Math. Statist., vol. 33, pp. 1065-1076, 1962.
- [7] M.D. Shuster and R.V.F. Lopes, "Parameter interference in distortion and alignment calibration," Paper No. AAS-94-186, Proceedings, AAS Flight Mechanics Meeting, Cocoa Beach, Florida, February, 1994.
- [8] D.F. Specht, "A general regression neural network," IEEE Trans. Neural Networks, vol. 2, no. 6, pp. 568-576, November 1991.
- [9] R.H. Stanton, J.W. Alexander and E.W. Dennison, "CCD star tracker experience - Key results from Astro 1 flight," SPIE Vol. 1949, Space Guidance, Control and Tracking, 1993.
- [10] J.R. Thompson and R.A. Tapia, *Nonparametric Function Estimation, Modeling, and Simulation*. Philadelphia: SIAM Press, 1990.
- [11] J.R. Wertz, (Ed.), *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Boston, 1978.