Autonomous Recalibration of Star Trackers

John Enright, *Member, IEEE*, Ilija Jovanovic, *Student Member, IEEE*, and Brendon Vaz

***Abstract*—Star trackers must be calibrated prior to flight so that they can make accurate measurements of star positions within the instrument field of view. This calibration is usually performed in atmosphere and after the sensor is launched; it is not uncommon to observe a small shift in some of the calibration parameters. In this paper, we explore several autonomous strategies for on-orbit recalibration of star trackers. We present an improved version of a popular camera model, develop optimizations to identify optimal parameter values, and validate performance using the data collected from on-orbit sensors. When compared with human-mediated batch processing, autonomous methods have comparable reliability, performance, and commissioning time. The sensor datasets used in this paper come from six Sinclair Interplanetary ST-16 star trackers launched between November 2013 and July 2014. Both batch and autonomous approaches to on-orbit calibration yield improvements in measurement availability as well as a 20%–80% reduction in residual geometric error compared to ground calibrations.**

***Index Terms*—Star trackers, calibration, parameter estimation, extended Kalman filter, nonlinear least squares.**

I. Introduction

An ESSENTIAL step in star tracker processing is to calculate the direction vector corresponding to an imaged star. This process relies on the use of a geometric camera model that can relate image coordinates to the sensor's external reference frame. Most camera models include parameters for focal distance and the principal point location [1], but many also attempt to correct distortion and other non-ideal behaviors [2], [3]. Using incorrect parameter values during star-vector calculations not only degrades the accuracy of the star tracker's attitude estimates, but can also impair its ability to identify stars at all.

Camera parameters are typically calibrated during the manufacture of the sensor. A variety of factors can subsequently introduce changes into these parameter values. These include: changes in refractive index from calibration conditions (i.e., air vs. vacuum); thermal expansion; disturbances from launch vibration; and component aging [4]. Some adjustment is often necessary during the initial satellite checkout phase to optimize sensor performance, however it may also be necessary to make periodic adjustments to the sensor parameters over the course of the mission lifetime.

Several different approaches to adjusting the camera parameters have been proposed in the literature. Pittelkau [5] and others have highlighted a number of the risks and benefits of different approaches to recalibration. Human-mediated batch-processing carries little operational risk but can cause excessive operational and bandwidth demands during satellite commissioning. This approach is most suitable for occasional updates under the direct supervision of the satellite operators. Autonomous, online algorithms require less operator intervention, but must be implemented with care to reduce mission risk. These algorithms may incorporate sliding windows or artificial dynamics to permit the estimator to track slow parameter changes over time.

This manuscript expands upon an earlier study that presented our preliminary analysis of recalibration using orbital data [6]. We consider the efficacy of both batch and sequential reprocessing techniques. We discuss details of the estimator formulation, compare the performance of different implementation against batch methods, and validate our approach using flight data from a number of Sinclair Interplanetary ST-16 star trackers.

A. Prior Work

Misaligned or uncalibrated attitude sensors and gyros may cause large measurement residuals in attitude determination systems. Ground-based calibration for attitude sensors, including star trackers, has been well documented, (e.g., for examples see [7] or [8]), however on-orbit adjustments are frequently required [1]. Batch processing to improve alignment has been standard practice for some time, and a significant body of research has explored Kalman Filters and other sequential estimators for online recalibration. Pittelkau [5] provides a good overview of this practice and discusses a number of missions including SIRTF [9] and Cassini [10] that used these techniques. Although these estimators have had success identifying tens of calibration parameters, they generally concentrate on system-level recalibration emphasizing sensor-to-sensor alignment and other similar bias corrections. Other sensors, such as the Inertial Stellar Compass [11] have used an internal estimator to fuse data between dissimilar sensing components. The ISC consists of an integrated star tracker and rate gyro, and uses an online filter to estimate gyro bias.

A number of researchers have studied the estimation of camera parameters using only data obtained by the star tracker itself. Much of this arose from John Junkins and his research group at Texas A&M University. Saaman *et al.* [12] formulated an estimator that would minimize errors between the star vectors calculated from imaged stars and the corresponding inertial star vectors. Singla *et al.* [13] extended the study

Manuscript received June 14, 2018; accepted July 8, 2018. Date of publication July 19, 2018; date of current version August 22, 2018. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was Dr. Shanhong Xia. *(Corresponding author: John Enright.)*

The authors are with the Department of Aerospace Engineering, Ryerson University, Toronto, ON M5B2K3, Canada (e-mail: jenright@ryerson.ca).

Digital Object Identifier 10.1109/JSEN.2018.2857621

TABLE I
SUMMARY OF ON-ORBIT ST16 SENSORS

Sensor	Mission Type	Altitude (km)	Inclination (deg.)	Calibration Dataset		Validation Dataset	
				Motion	Sample Period (s)	Motion	Sample Period (s)
1	Earth Obs.	570	97.8	Nadir Pointing	40	Nadir Pointing	40
2	Earth Obs.	570	97.8	Nadir Pointing	40	Nadir Pointing	40
3	Earth Obs.	620	98.4	Nadir Pointing	40	Nadir Pointing	40
4	Earth Obs.	620	98.4	Nadir Pointing	40	Nadir Pointing	40
5	Space Science	730	97.7	Slew	10	Slew	10
6	Space Science	670	98.0	Coarse Hold + Slew	10	Coarse Hold + Slew	10

and evaluated performance techniques using simulations and ground tests. Finally, Griffith *et al.* [14] proposed using a modified recursive least squares algorithm with additive process noise and a forgetting factor to give preference to the most recent available data. This is especially useful when the parameters may drift over time.

Other researchers have taken online star tracker calibration in different directions. Pal and Bhat [15] solve for the focal length and principal point using a closed form solution and then use a least-squares estimator to determine distortion parameters. Liu *et al.* [16] suggest collapsing several successive estimates into a single Kalman filtering step. Shen *et al.* [17] also demonstrate how the focal distance, principal point, and radial and tangential distortion parameters can all be estimated with a single filter.

This paper extends the existing research in several important ways.

- First, our model inverts a recent model developed by Wang *et al.* [2]. Unlike many popular approaches, our model calculates the measurement star vector directly and avoids the need to numerically solve nonlinear equations for each detected star.
- Second, we have extended Wang's method to use a star-pair arclength error metric and analytic gradient information. This avoids the inadvertent measurement weighting that arises when the calibration residual is calculated using the cosine of the arclength.
- Third, we evaluate batch and recursive performance using a sizable archive of orbital data. These datasets span multiple sensor units and many different areas of the sky. Because it is difficult to include effects of modeling errors in simulation, many ground-based performance predictions tend to be overly optimistic; our inclusion of a large amount orbital data helps to evaluate the benefits of these estimators under realistic operating conditions.

B. On-Orbit Sensors

The sensor we have used as a baseline design in this study is the Sinclair Interplanetary ST-16 star tracker. This is a small (90 g, 59 × 56 × 32 mm) star tracker intended for microsatellites or large nanosatellites. The sensor provides three-axis attitude measurements (∼ 7 arcseconds, cross axis) at 2 Hz. We have detailed additional features of the design and performance of the sensor in [18]–[20].

The first ST-16 sensors were launched in November 2013, and by July 2017 a total of 21 star trackers were on-orbit. In this paper, we examine telemetry from six of these sensors (on four satellites), and use these data to evaluate the effectiveness of our recalibration techniques. The batch algorithms presented in this study were used to tune the camera parameters for the on-orbit sensors and the sequential estimators have been tested on engineering model hardware.

It is worth mentioning a few details of the ST-16 processing model that are germane to this study. Each time the sensor is asked to take a reading it takes two image exposures in quick succession, locates and identifies stars in each image, and returns a single quaternion estimate corresponding to the image with the highest quality solution. This double-exposure sequence allows us to estimate the sensor rotation rate and compensate for warping effect of the sensor's electronic rolling shutter [21], without the need to carry persistent state in the sensor (this processing approach means that there is essentially no start-up time required to obtain an attitude lock). We term each reading a *telemetry frame*. A series of frames constitutes a *telemetry set*. In normal sensor operation, the telemetry frames only contain status information, attitude, and rate, however, the operator can command the sensor to return the star centroid locations, and a summary of star matches. Although the ST-16 can produce telemetry frames at 2 Hz, operational limitations typically dictate that our collected telemetry be sampled much more sparsely.

The extended telemetry information is essential for recalibrating the sensor camera parameters. Using a ground-based implementation of the sensor flight software, we can re-run the star vector, matching, and attitude algorithms and evaluate the effect of changes to the calibration parameter values. This approach bypasses the raw image processing. When compared to the original version of the telemetry frame, the re-matched frame may have different calculated star vectors. If these star vectors agree better with the known (cataloged) star-pattern geometry, the matching routines may successfully match additional stars, and the overall attitude solution may improve as well.

Table I shows a summary of the sensors considered in this study. From the available telemetry we selected two subsets for each sensor. The first is a longer dataset we use for calibrating the sensor parameters; the second — a shorter segment, imaging a different part of the sky — is a validation set. The time intervals between the paired datasets range from a few hours to several days — short enough that we do not expect any aging effects, but long enough to decouple any temperature-induced changes.

To assess the performance improvement of the optimized camera parameters we compute error and availability metrics using re-matched versions of the validation telemetry set. The telemetry sets have been edited to remove periods of eclipse as well as Earth and sun incursions into the sensor field of view (FOV). All orbits are sun-synchronous, and the specified altitude is an approximation based on the orbital semi-major axis and the equatorial earth radius.

II. Parameter Fitting

In this section we develop the mathematical framework that will allow us to optimize the ST-16 camera parameters using on-orbit data. We start with a summary of the camera model used by the ST-16. We then formulate a cost function using the arclength between pairs of stars. This cost function and its Jacobian form the basis for deriving batch and sequential least-squares estimators for the sensor parameters.

A. Camera Model

A camera model defines the series of relationships relating unit star vectors and star centroid locations on the detector. Model parameters can be physically motivated (e.g., focal length) or merely abstract correction coefficients. The emphasis in model selection is to allow precise calculation of the incident star vectors, not on general optical characterization. Thus, traditional aberrations such as coma and astigmatism are not modeled explicitly; it is only their net contribution to centroid motion that must be represented. Chromatic aberration can be significant since its effect can vary with stellar spectrum, however per-star corrections to the calibrated focal length can be quite effective in mitigating this problem [22].

The ST-16 calibration model is an eight-parameter, inverted version of the model presented by Wang *et al.* [2]. In their work, they use the incident ray to calculate the ray intersection with the detector; we solve for the incident ray corresponding to a known detector location. Thus, our formulation avoids the need for numerical iteration when calculating the star vectors. Other models are common in the computer vision (e.g., [3]) and star tracker (e.g., [8]) literature, but we focus this study on the model used in the actual ST-16 flight software. We have made a limited comparison to the expressive power of other camera models by restricting the parameters altered during optimization. In this manner we can mimic the behavior of reduced-order models such as those developed by Samaaan *et al.* [12].

This study considers only the intrinsic parameters of the sensor. Determining extrinsic quantities, such as the misalignment between a star trackers and the spacecraft, would rely on a secondary source of attitude information. No such secondary reference was available in this study.

We define the detector-plane reference frame D with respect to the virtual detector plane. This virtual detector lies in front of the theoretical pinhole (see Figure 1) and avoids the need to handle image inversion in our derivations. The x-axis lies in the direction of increasing rows and the y-axis lies in the direction of increasing columns. The nominal z-axis lies along the optical axis and points out of the sensor. Starting with the detected centroid location (m_c, n_c), we calculate the detector-plane displacements, u and v, from the optical center, (m_0, n_0).

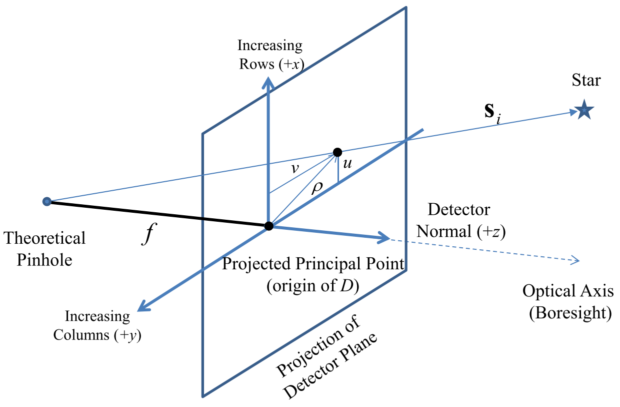


Fig. 1. Pinhole camera model.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \Delta\,(m_c - m_0) \\ g_y \Delta\,(n_c - n_0) \end{bmatrix} \tag{1}$$

where Δ is the pixel size and g_y is the a relative skew factor between pixel spacing in the x- and y-directions.

We apply a distortion correction to compensate for the decentering and prism effects. These aberrations are parameterized by the constants a_1 and a_2 and represent linearized rotations in the X- and Y-axes. These rotations give the modified coordinates similar to those given by Wang *et al.* [2]

$$U = \frac{u \cdot f}{a_1 v + a_2 u + f} \tag{2}$$

$$V = \frac{v \cdot f}{a_1 v + a_2 u + f} \tag{3}$$

From these coordinates, we calculate a radial correction factor, B, of the form

$$B = 1 + b_2 \rho^2 + b_4 \rho^4 \tag{4}$$

where

$$\rho = \sqrt{U^2 + V^2} \tag{5}$$

and b_2 and b_4 are the radial distortion coefficients.

We then have two position vectors of points that lie on the ray to the star: the theoretical pinhole and the focal plane intersection. Thus, the direction to the star is

$$\mathbf{r}_D = \begin{bmatrix} BU \\ BV \\ f \end{bmatrix} \tag{6}$$

We can then normalize this vector to arrive at $\mathbf{s}_D$.

$$\mathbf{s}_D = \frac{\mathbf{r}_D}{\|\mathbf{r}_D\|} \tag{7}$$

We can succinctly write the parameter values as a state vector

$$\mathbf{x} \triangleq \begin{bmatrix} f & m_0 & n_0 & b_4 & b_2 & a_1 & a_2 & g_y \end{bmatrix}^T \tag{8}$$

This model allows us to predict the star vector $\mathbf{s}_D$ in the detector frame corresponding to a measured centroid location (m_c, n_c). During ground calibration we are able to supplement the calibration with an alignment step to measure the rotation between the detector frame and alignment features on the sensor chassis. This allows us to calculate star vectors (and attitude) relative to the external frame of the sensor, e.g., $\mathbf{s}_S$. We store the rotation matrix $\mathbf{C}_{SD}$ as an additional intrinsic parameter in the sensor memory.

B. Error Models

Formulating our calibration cost-function in terms of the angular separation between observed stars ensures that the resulting optimization is independent of absolute attitude. Given two star vectors expressed in the detector frame, $\mathbf{s}_{D,i}$ and $\mathbf{s}_{D,j}$ the measured angular separation — i.e., arclength — between them is

$$\tilde{\varphi}_{ij} = \arccos\left(\mathbf{s}_{D,i}^T \mathbf{s}_{D,j}\right) \tag{9}$$

Alternately, using the cataloged, inertial-frame star vectors $\mathbf{s}_{I,i}$ and $\mathbf{s}_{I,j}$ we can calculate the true separation between the same two stars

$$\varphi_{ij} = \arccos\left(\mathbf{s}_{I,i}^T \mathbf{s}_{I,j}\right) \tag{10}$$

Many extant calibration models use the difference of dot-products as a surrogate for calculating the actual arclength error, i.e.,

$$\begin{aligned} e_{ij} &= \cos\tilde{\varphi}_{ij} - \cos\varphi_{ij} \\ &= \cos\left(\varphi + \delta_2 - \delta_1\right) - \cos\varphi_{ij} \\ &\approx \left(\delta_1 - \delta_2\right)\sin\varphi \end{aligned} \tag{11}$$

where δ_i and δ_j are the one-dimensional error contributions from each star. This result is problematic as the error is proportional not just to the contributions from each centroid, but to $\sin\varphi$ as well. Therefore, this error formulation introduces arbitrary and unjustified weights into the error contributions of each star pair. Instead, we calculate arclength errors explicitly

$$e_{ij} \triangleq \delta\varphi_{ij} = \tilde{\varphi}_{ij} - \varphi_{ij} \tag{12}$$

We can enumerate all the admissible star pairs in a set of observations and form the observation vector

$$\boldsymbol{\delta\varphi} = \left[\delta\varphi_{ij}\right] \tag{13}$$

In a scene with N stars there are $N\left(N-1\right)/2$ possible φ_{ij} measurements available. We must select a non-redundant set of measurements to ensure that the measurement covariance remains positive definite. We can deduce that there are at most $2N-3$ *independent* arclengths in a scene of N stars. This follows from the observation that each star contributes two position measurements, but there are three bulk rotations of the whole star pattern that are unobservable in the arclength pseudo-measurements.

Optimization of the chosen arclengths may be possible, but such a scheme lies beyond the scope of this study. Instead we have adopted the following heuristic to select arclengths: *First, number the N observed stars so that no three sequential stars are collinear. Arclengths are computed from Star-i to Star-*$(i+1)$ *and Star-*$(i+2)$. *This defines the* $2N-3$ *observations in* $\boldsymbol{\delta\varphi}$ It is trivial to show that this generates a completely connected set of stars with the desired number of arclengths.

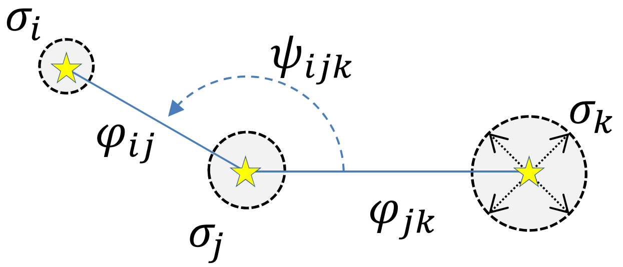


Fig. 2. Geometric relationships between three stars. This figure illustrates the arc-lengths (φ), dihedral angles (ψ), and star position uncertainty (σ).

We can form $\boldsymbol{\delta\varphi}$ for a single telemetry frame or as the aggregate of all the star pairs over a batch of telemetry frames.

Our quadratic cost function J is then

$$J = \frac{1}{2}\boldsymbol{\delta\varphi}^T \mathbf{W}\boldsymbol{\delta\varphi} \tag{14}$$

Here we have introduced a weighting matrix $\mathbf{W}$ to account for differences between the measurements. Measurements are weighted using the inverse covariance of the arclength error. To find an expression for the elements of the covariance, $\mathbf{R}$, we consider how this matrix is defined

$$\mathbf{R} \equiv E\left\{\boldsymbol{\delta\varphi}\boldsymbol{\delta\varphi}^T\right\} \tag{15}$$

If the star-position errors are isotropic and small compared to the arclengths themselves we can write simple expressions for the elements of $\mathbf{R}$. Figure 2 illustrates the relevant geometric relationships and Appendix A provides a proof that $\mathbf{R}$ is invertible, but details of these derivations have been omitted for the sake of brevity. The diagonal elements are related to the variance of the star centroid locations, e.g., σ_i^2.

$$R_{ij,ij} = E\left\{\delta\varphi_{ij}^2\right\} = \sigma_i^2 + \sigma_j^2 \tag{16}$$

The off-diagonals depend on the specific geometry of the stars involved. Off diagonal elements corresponding to arclengths that share no common stars will be zero. When stars are shared, we assume that the quantity $E\left\{\delta\varphi_{ij}\delta\varphi_{jk}\right\}$ depends only on the error at the common star, j. This allows us to approximate the off-diagonal term as

$$R_{ij,jk} \equiv R_{ijk} = E\left\{\delta\varphi_{ij}\delta\varphi_{jk}\right\} \approx \sigma_j^2 \cos\psi_{ijk} \tag{17}$$

The quantity ψ_{ijk} represents the dihedral angle between the two outgoing arcs from star-j. From spherical geometry we calculate

$$\cos\psi_{ijk} = \frac{\cos\varphi_{ik} - \cos\varphi_{ij}\cos\varphi_{jk}}{\sin\varphi_{ij}\sin\varphi_{jk}} \tag{18}$$

This formulation of the $\mathbf{R}$ matrix allows us to use any available model for the centroid error. The σ_i values can be set to a constant centroid accuracy or may depend on other factors such as star brightness or position in the field of view.

Cost functions are evaluated over the star pairs extracted from one or more telemetry frames. We use the notation,

T_k to refer to a single telemetry frame or $\mathbf{T}$ to refer to a telemetry set. When we re-match a telemetry frame T_k with a new set of sensor parameters, $\mathbf{x}$, we obtain a re-matched telemetry frame, $T_k(\mathbf{x})$. Thus, we can use the following shorthand to represent the current cost function evaluation at any time during the optimization

$$J = J(\mathbf{x}) = J\{\mathbf{T}(\mathbf{x})\} \tag{19}$$

Similarly $\boldsymbol{\delta\varphi}(\mathbf{T}(\mathbf{x}))$ denotes the vector of arclength errors for the re-matched telemetry set $\mathbf{T}(\mathbf{x})$. This telemetry set has RMS error

$$\delta\varphi_{\mathrm{RMS}}(\mathbf{T}(\mathbf{x})) = \sqrt{\frac{J(\mathbf{T}(\mathbf{x}))}{N_\varphi}} \tag{20}$$

where N_φ is the number of star pairs in each set.

The matching process in the ST-16 includes geometric consistency checks so different sets of parameter values — e.g., $\mathbf{x}_A$ and $\mathbf{x}_B$ — may give rise to arclength error vectors of different length. Thus $N_\varphi(\mathbf{x}_A) \neq N_\varphi(\mathbf{x}_B)$. To ensure that error comparisons between parameters sets are fair, we must evaluate $\delta\varphi$ using a common set of star pairs. Thus, we use star matches identified using one of the sets $\mathbf{T}(\mathbf{x}_A)$, but calculate star vectors for the identified stars using a different set of parameters $\mathbf{s}_D(\mathbf{x}_B)$. We denote this common-pair metric as $\delta\varphi_{\mathrm{RMS}}(\mathbf{T}(\mathbf{x}_A), \mathbf{s}(\mathbf{x}_B))$. In our study, we used the largest set of star pairs available for these calculations.

C. Batch Estimator

For the batch estimator, we require the Jacobian of the sensor cost function

$$\mathbf{H}_{\mathrm{batch}} \equiv \frac{\partial \boldsymbol{\delta\varphi}}{\partial \mathbf{x}} = \frac{\partial \tilde{\boldsymbol{\varphi}}}{\partial \mathbf{x}} \tag{21}$$

We build the Jacobian using the available star pairs in a sequence of telemetry frames, $\mathbf{T}$. For the ST-16, each frame, T_k contains two exposures, and each image will contain a number of stars. To prevent the possibility of incorrect matches corrupting the fit, we only consider images in which we match at least a minimum number of stars, i.e., $N_{\mathrm{match}} \geq N_{\mathrm{min}}$. In these tests $N_{\mathrm{min}} = 4$ is a good heuristic for selecting good-quality images. Appendix B gives analytical expressions for the partial derivatives needed to assemble $\mathbf{H}$. With error, $\boldsymbol{\delta\varphi}$, and the Jacobian, $\mathbf{H}_{\mathrm{batch}}$, we can use any batch minimizer to find the optimal $\hat{\mathbf{x}}$. In this study we use Matlab's Levenberg-Marquardt implementation (through `lsqnonlin`), but a simpler Gauss-Newton solver may also be suitable. Initial parameter values, $\mathbf{x}_0$, can be derived from ground-based calibration but in practice the optimization is robust to starting values. For the sensors examined in this study, initializing $\mathbf{x}$ with nominal values, particularly for the distortion coefficients, has yielded good results.

Once we have found $\hat{\mathbf{x}}$, we can reprocess $\mathbf{T}$ with the optimized camera parameters. This often leads to increases in both the availability as well as the average number of matched stars. Repeating the optimization with the additional matching information can sometimes improve the quality of $\hat{\mathbf{x}}$, but after 2-3 optimization cycles there tends to be little change in the calibration.

TABLE II
MODEL PARAMETER NORMALIZATION

Parameters	$\bar{x}_p$	σ_p
f	16 mm	8 μm
m_0	972 pixels	30 pixels
n_0	1296 pixels	30 pixels
b_4	0	10^8
b_2	0	10^3
a_1	0	0.01 rad
a_2	0	0.01 rad
g_y	1	2×10^{-4}

We can improve the numerical conditioning of the optimization by transforming the model parameters so they have zero mean and unity variance (between units). This transformation takes the form

$$x'_p = \frac{x_p - \bar{x}_p}{\sigma_p} \tag{22}$$

Thus, the modified states become normalized perturbations to the nominal values.

The means, $\bar{x}_p$, and standard deviations, σ_p, describe typical parameter variations from unit-to-unit. Table II shows the values derived from our ground calibration of many ST-16 sensors. Significant improvement in numerical conditioning can be achieved, even if these quantities are only known approximately.

To make use of the transformed model values, we consider the scaled Jacobian

$$\mathbf{H}' = \mathbf{H}_{\mathrm{batch}} \mathbf{W}_J \tag{23}$$

where $\mathbf{W}_J$ is a diagonal matrix whose elements are the σ_p. Thus, $\mathbf{H}'$ effectively measures the effect on the cost function of a *relative* change in the transformed parameter value. Examining the $\mathbf{S}$ and $\mathbf{V}$ matrices in the typical SVD

$$\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{H}' \tag{24}$$

allows us to classify the relative observability of the different model parameters. Parameters included in model vectors corresponding to large singular values are more observable than those corresponding to small singular values. To implement this approach, the Jacobians are scaled during each optimization iteration (using Section II-C) and the parameters are scaled using Section II-C.

This scaling brings the condition number of $\mathbf{H}$ from about 10^{14} down to around $120 - 150$ depending on the dataset. This represents a significant improvement in the numerical conditioning. In absolute terms, the condition number is still fairly high, suggesting that separating some model parameters may still be difficult. Analysis of the columns of $\mathbf{V}$ shows the primary problem: although the net effect of m_0 and a_2 (and similarly, n_0 and a_1), are readily observable, their differential effect is harder to separate. This observation supports our earlier discussion regarding the similar effects of these parameter pairs and explains how different optimizations can arrive at different optimal parameter values.

Finally, we note that our error metric is attitude independent. Consequently, we cannot detect any bulk rotational bias that the parameter changes may impart. To avoid the need to re-align the star tracker with other spacecraft instruments, after the recalibration is complete, we take a final pass through the data comparing the attitude estimates obtained from $\mathbf{T}(\mathbf{x}_0)$ with those from $\mathbf{T}\left(\hat{\mathbf{x}}\right)$. We solve for the average rotation [23] between these results and update $\mathbf{C}_{SD}$ to null out any secular change.

D. Sequential Estimators

Batch recalibration is effective but makes significant demands of the mission operations team. Scheduling full telemetry downloads and uploading new calibration constants can introduce unwanted delays during satellite commissioning. Migrating the recalibration logic onto the sensor itself is attractive from an operational standpoint but processing large telemetry batches on the sensor would be impractical. Sequential estimators can achieve comparable performance levels with only modest memory requirements. Although the overall computational requirements are similar, the required calculations can be spread out over time, minimizing the impact on sensor performance. In this section, we consider several estimators based on the Extended Kalman Filter for determining $\hat{\mathbf{x}}$.

Starting with initial estimates of the sensor parameters, $\widehat{\mathbf{x}}_0$, and state covariance, we process the sensor readings one telemetry frame at a time. For telemetry frame T_k we employ the standard set of EKF update equations for Kalman Gain, $\mathbf{K}$, state, $\widehat{\mathbf{x}}$, and state covariance, $\mathbf{P}$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{25}$$

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\delta\varphi}_k \tag{26}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^- \tag{27}$$

where $\mathbf{H}_k \equiv \mathbf{H}\left(\hat{\mathbf{x}}_k\right)$. The observation vector $\boldsymbol{\delta\varphi}_k$ is formed from the two image exposures in the k-th telemetry frame. We assume that the measurement covariance, $\mathbf{R}_k$ is defined similarly to Section II-B and Section II-B, but here they will only make use of arclengths from a single telemetry frame.

We assume that the model parameters are constants, so there are no dynamics to affect state propagation from measurement to measurement. Artificial process noise, represented by covariance $\mathbf{Q}$ can be added to keep the elements of $\mathbf{P}$ from getting too small. Between one frame and the next we have

$$\mathbf{x}_{k+1}^- = \mathbf{x}_k^+ \tag{28}$$

$$\mathbf{P}_{k+1}^- = \mathbf{P}_k^+ + \mathbf{Q}_k \tag{29}$$

Our initial approach to recalibration [6], employed a standard EKF for sequential estimation. To improve numerical conditioning and reduce computational costs we have adopted a Carlson-Schmidt Square-Root filter (SQRF) algorithm [24]. The SQRF filter equations are presented in Appendix C.

Several variations of the sequential filter are possible. As part of our experiments, we evaluate how the following changes can help tune the estimators performance.

1) Noise Covariance: The performance of the parameter estimators depends heavily on our choices for the process and measurement noise covariance matrices. As part of this study we consider how different values of $\mathbf{Q}$ — including none at all — can alter convergence behavior. Without the $\mathbf{Q}$ matrix, the estimator is effectively just a sequential least-squares estimator. Although artificial process noise is a common technique for improving EKF convergence, it may not be necessary.

2) Role of Re-Matching: Each T_k contains the image locations of the matched stars, but also the locations of other stars that were not successfully matched. This matching process uses the calculated geometry of stars in the star tracker image to identify the corresponding catalog stars and potential matches are allowed or rejected based on the consistency of their geometry with the catalog values. As the camera parameters improve, so too does the matching performance and $T\left(\mathbf{x}_k^+\right)$ may contain more matched stars than $T\left(\mathbf{x}_k^-\right)$. Thus we consider the effectiveness of the following variations:

- Match only with the baseline sensor values, i.e., $T_k = T_k(\mathbf{x}_0)$.
- Match with the latest sensor estimate, i.e, $T_k = T_k\left(\mathbf{x}_k^-\right)$

We recognize that keeping the baseline sensor values is a conservative approach, but this may be operationally attractive in that it will always preserve performance of the sensor in case of filter divergence

III. Results and Discussion

We wish to understand the effectiveness of both the batch and sequential estimators in recalibrating the camera parameters on the ST-16 star trackers. Evaluating the performance of these algorithms against orbital data helps guide future development and gives us insight into the accuracy of our sensor models. Repeating these tests for different sensors provides some perspective into any common trends in parameter behavior and may help improve the initial ground calibration.

Performance improvements can be assessed using several different metrics:

- Availability. This is the fraction of telemetry frames for which the star tracker returns a valid attitude fix. Poor calibration leads to poor measurement of the star vectors. When the sensed geometry is of questionable quality, the sensor star-matching algorithms may report a failed match. Good camera calibration parameters allow the sensor to confidently match image stars against its onboard catalog.
- Matched Stars. For a given telemetry dataset, this metric captures the total number of stars that are successfully matched. Matching additional stars generally correlates with lower attitude estimation errors and helps maintain availability if viewing conditions degrade (e.g., higher body rates, stray light, etc.). For this study we express this as the Matched Star Ratio

$$MSR \triangleq \frac{N(\mathbf{x}_A)}{N(\mathbf{x}_B)} \tag{30}$$

- RMS arclength error ($\delta\varphi_{\mathrm{RMS}}$). We use the arclength error as the basis for our optimization cost function. Not only does this have a strong effect on the matching performance, but improvements in arclength error also have a significant effect on attitude error. As discussed previously, calibrations are compared by evaluating the RMS error calculated from a common set of identified star pairs.

TABLE III
PARAMETER CHANGES DURING BATCH RECALIBRATION

Parameter	Sensor					
	1	2	3	4	5	6
f	-8.54	-8.04	-8.17	-9.46	0.08	-0.46
m_0	1.34	0.71	1.12	0.39	-2.59	3.16
n_0	1.68	-8.37	1.61	8.08	4.00	-9.81
b_4	1.04	1.41	1.37	0.67	-0.10	0.44
b_2	-1.03	-1.17	-1.22	-0.90	0.09	-0.60
a_1	0.51	-4.96	1.02	3.58	2.34	-4.93
a_2	0.15	0.47	0.41	-0.10	-1.52	1.94
g_y	3.76	5.14	3.05	2.97	0.36	-1.30

TABLE IV
BATCH CALIBRATION PERFORMANCE RESULTS

Sensor	Avaliability		Matched	$\delta\varphi_{RMS}$(μrad)	
	Initial	Final	Star Ratio	Initial	Final
1	0.96	1.00	1.55	422.6	80.3
2	0.98	1.00	1.20	332.7	91.4
3	0.44	0.98	2.46	377.0	89.9
4	0.46	1.00	2.70	412.9	95.6
5	1.00	1.00	1.00	83.0	62.2
6	0.93	0.97	1.04	248.3	123.1

A. Batch Estimator Results

Our first set of tests involved batch calibration of each sensor. In these trials we determined the optimal $\hat{\mathbf{x}}$ using the telemetry frames in the calibration telemetry sets. Not only are these results useful for tuning the performance of these sensors, but by examining the ensemble results over all the data, we can look for common trends.

Table III shows the parameter changes for all six sensors. The parameters have vastly different scales, so the changes represent the number of σ_p increments between the initial lab calibration and the final optimized values (see Table II for the scaling constants). With the exception of Sensor-5, many of the parameters exhibit similar changes in both magnitude and direction: the f, b_4, b_2, and g_y parameters are remarkably consistent; These results suggest that the ground calibrations exhibit a consistent bias, particularly in focal length. The cause of this effect is unclear; possibilities include equipment limitations, testing in atmosphere, or the numerical behavior of the ground-calibration optimization.[1]

In Sensor-2 and Sensor-4 we see opposite motions in n_0 and a_1. The principal point and detector tilt parameters are known to have significant linear dependence and the observed complementary motions suggests that the cost function is fairly shallow in the direction of combined motion.

Table IV summarizes the performance changes that accompany the optimizations. Most sensors show improvement in availability and an increase in the number of matched stars (the matched star ratio is the ratio of successfully identified stars after and before the optimization). All sensors also show sizable reductions in error — most are better than 70%. Sensor-6 data was collected with sub-optimal star detection settings, and demonstrates higher error than is typical.

The calibration behavior of Sensor-5 is strikingly different than the other sensors (as seen in both Table III and Table IV). We see very little change in f, and the changes in the other parameters do not follow the trends observed in the other sensors. Judging from the performance metrics, it appears that some of the initial Sensor-5 parameters were quite close to the sensor's optimal values. Consequently this sensor saw smaller changes in parameters and performance during recalibration. Sensor-6 displays some of the same characteristics — e.g., a smaller improvement in error — but the effects are not quite as pronounced.

B. Sequential Estimator Results

Having established the baseline performance of the batch estimators we wish to assess the relative performance of the sequential estimators. Establishing a basis for comparison across multiple parameters, sensors, datasets, and filter variants is difficult. To help clarify our analysis we identify a number of specific questions that we would like to resolve:

1) Is the error performance of the sequential estimators similar to that of the batch recalibrations?
2) Which filter variants are most or least promising?
3) Do the filters exhibit acceptable convergence?
4) Are the optimized parameter values from the sequential estimators similar to those obtained from batch recalibration?
5) How does our camera model compare to the Samaan *et al.* [12] model?

To implement the comparison to the Samaan model, we reformulate the filters to only optimize the focal length and principal point (i.e., f, m_0, and n_0). This does not represent a complete re-implementation of the associated measurement and error models, but should be indicative of the power of the reduced-order model to capture the sensor behaviors.

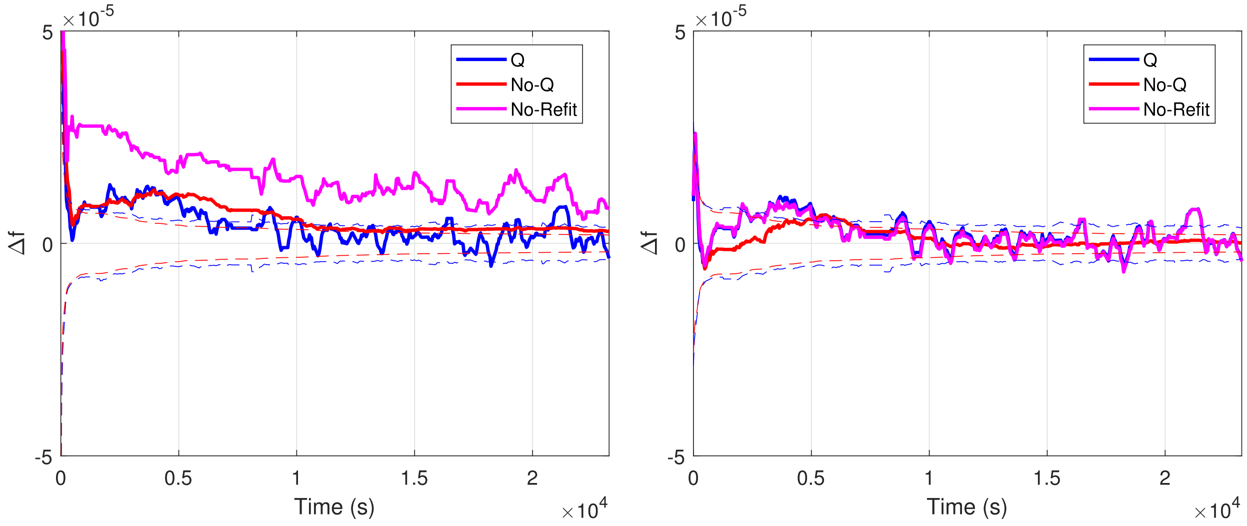
The following sections highlight some of the most important results from our tests.

1) Error and Effectiveness: We measured the filter performance by recalibrating the sensor parameters using a calibration dataset and then evaluating the corresponding RMSE values for a validation dataset. By using disjoint datasets we lose the ability to track any physical changes in the sensor but do not need to worry about overfitting. The three primary variants of the sequential estimator were:

- The `Q` formulation. This filter includes process noise and each telemetry frame is rematched before the measurement updates using the current value of $\mathbf{x}_k$.
- The `No-Q` formulation. This filter does not include process noise; the algorithm re-matches the telemetry frames as in the `Q` case.
- The `No-Refit` formulation. This filter includes process noise and only the raw telemetry, i.e., $\mathbf{T}(\mathbf{x}_0)$.

Table V shows the error results from the sequential trials. The RMSE from the batch-calibrations are provided for comparison. Although it is clear that the batch values are not necessarily optimal for the validation datasets — some of the sequential estimators do better — they generally represent

[1] Ground calibrations use centroid data gathered using an artificial star and a motorized platform. The cost function is similar, but not identical to the methods detailed in this study.



(a) (b)

Fig. 3. Focal length convergence for Sensor-4. Initial parameter values can be obtained from a) initial laboratory calibration or b) the final batch values. The dashed lines indicate $3-\sigma$ bounds.

good performance. Examining the data and details of spacecraft motion during the datasets allows us make the following observations:

- For most of the sensors the performance of the SQRF No-Q filters come very close to that of the batch estimator.
- Slow convergence is likely the cause in the few cases where the filters have appreciably higher error than the batch sensor (i.e., Sensors-5 and Sensor-2). We have observed that providing additional calibration data will eventually remove the remaining excess error.
- The No-Refit errors are higher than the other filter variants for most sensors. The overall performance may still be acceptable (with the exception of Sensor-6), but unless computational resources are severely limited, refitting appears to be of significant benefit.
- The Q trials have similar errors to the No-Q trials. Unless relatively fast parameter drift is expected — i.e., on timescales of a single orbit — the use of process noise appears to be unnecessary.
- RMS errors from the reduced order models are between 5% and 13% higher than the the full calibrations. The `1R` test in Table V illustrates the typical performance difference.
- Finally we note that the spacecraft motion and observations during these datasets can affect the overall error magnitudes. For example, the validation errors for Sensors-1, 2, 3 are significantly higher than the calibration error ratios, but for Sensor-6, the opposite appears to hold.

Table VI shows the star-matching and availability results from the sequential trials and the validation dataset. The matched star ratios are evaluated relative to the raw lab-calibrated telemetry. The differences in availability are due to a handful of telemetry frames with marginal star availability. We note that all three filter variants are generally effective in improving these metrics.

TABLE V
ESTIMATOR VALIDATION RMSE (μRAD)

Sensor	Raw	Batch	Q	No-Q	No-Refit
1	385.3	101.8	108.0	98.9	93.5
1R[a]	385.3	175.6	132.3	114.0	103.3
2	316.9	106.8	118.9	106.4	102.1
3	402.1	86.2	85.8	85.0	90.0
4	415.1	104.2	103.6	105.0	111.5
5	71.3	52.9	59.7	59.0	59.2
6	247.4	174.1	164.3	167.5	201.6

[a] Sensor-1 recalibrated with reduced parameter set described in [12]

2) Transient Performance: Examining the sensor model parameters and their behavior over time, Figure 3a shows Δf_k obtained from the SQRF estimators for Sensor-4. These plots are typical of both sensor and parameter behaviors. The plots show differences between the KF derived estimates and the corresponding batch values, i.e.,

$$\Delta x_{i,k} \equiv \widehat{x}_{i,k} - \widehat{x}_{i,\text{batch}} \tag{31}$$

We have shown in Table V that the sequential estimators do not yield the exact same error values as the batch estimates. Thus, it is unsurprising that the parameters differ as well (Figure 3a). Insofar as we are concerned with reducing error and improving availability, the batch or sequential estimators are roughly equivalent. If, instead, we wish to look for gradual parameter changes, possibly driven by thermal or aging effects, these phenomena may be obscured by presence of local minima. Figure 3a and Figure 3b illustrate this effect. In the first figure, the sequential estimator is initialized with the ground-calibrated parameters; in the second, the batch-optimized values are used. Both the `Q` and `No-Q` results show consistent parameter values. The `No-Refit` parameters differ between the two tests on account of the different stars that make up the $T_k(\mathbf{x}_k)$ telemetry frames. These results suggest that the transient parameter values are driven by the

TABLE VI
MATCHING AND AVAILABILITY COMPARISONS

	Matched Star Ratio				Availability			
Sensor	Batch	Q	No-Q	No-Refit	Batch	Q	No-Q	No-Refit
1	1.41	1.41	1.41	1.41	1.00	1.00	1.00	1.00
2	1.16	1.16	1.16	1.16	0.95	0.95	0.95	0.95
3	0.98	0.99	1.01	0.97	0.95	0.95	0.95	0.95
4	1.01	1.01	1.02	0.88	0.94	0.94	0.94	0.91
5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6	1.01	0.99	0.99	0.99	0.85	0.85	0.85	0.85

available observations and not by the convergence dynamics of the sequential estimators. Although we have illustrated this anecdotally, this finding holds across all of the sensors studied.

An additional feature of interest is that the Q-variant estimates tend to wander, even after the initial period of convergence. Two variations of the RMS error calculations provide insight into the filter behavior during this period. The conventional $\delta\varphi_{\text{RMS}}(T(\widehat{\mathbf{x}}_k))$ definition uses a single set of $\widehat{\mathbf{x}}_k$ parameters, i.e., the RMS error from

$$\boldsymbol{\delta\varphi}_{\text{TAIL}}\left(\widehat{\mathbf{x}}_k\right) = \begin{bmatrix} \boldsymbol{\delta\varphi}_1\left(\widehat{\mathbf{x}}_k\right) \\ \boldsymbol{\delta\varphi}_2\left(\widehat{\mathbf{x}}_k\right) \\ \vdots \end{bmatrix} \tag{32}$$

where $\boldsymbol{\delta\varphi}_j$ are the arc length errors from telemetry frame-j. The error is calculated over the *whole* dataset using the current parameter values, $\widehat{\mathbf{x}}_k$. These RMS values change with the filter parameters. In contrast we can also define the *aggregate* RMSE in terms of the per-frame parameter values:

$$\boldsymbol{\delta\varphi}_{\text{AGG}} = \begin{bmatrix} \boldsymbol{\delta\varphi}_1\left(\widehat{\mathbf{x}}_1\right) \\ \boldsymbol{\delta\varphi}_2\left(\widehat{\mathbf{x}}_2\right) \\ \vdots \end{bmatrix} \tag{33}$$

This latter quantity is a scalar value that depends on the time-history of $\widehat{\mathbf{x}}$. It represents a bound on the error that lets us assess how much of the per-frame error is influencing the filter parameter values. Figure 4 shows a plot of these quantities for the Sensor-4 data. That the No-Q error ends up very close to its aggregate value suggests that the filter is converging well. The comparatively large difference between the tail and aggregate Q performance provides further evidence that the larger state covariance is actually causing the filter to overfit the parameter estimates to each telemetry frame. Nevertheless the performance of both sequential filters would be adequate in most situations.

The SQRF estimators can sometimes arrive at notably different distortion parameters (see Table VII) than the batch optimizations. The combined effect of f, b_4, and b_2 can be plotted graphically (see Figure 5). The curves show the net radial difference, relative to the batch results as a function of off-axis distance. Although the No-Refit curve behaves differently than the other two, the overall radial behavior is quite similar to the batch results within about 800 pixels of the boresight.

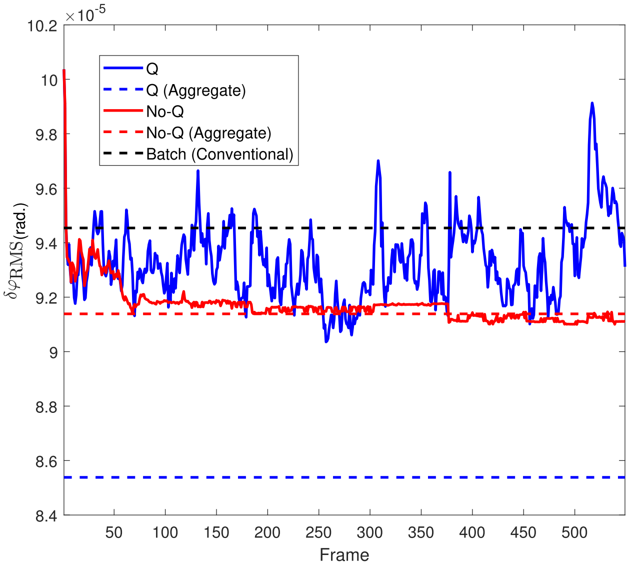


Fig. 4. Conventional and aggregate errors using the Q and No-Q methods for Sensor-4 ($\mathbf{x}_0 = \mathbf{x}_{\text{batch}}$).

TABLE VII
COMPARISON OF DISTORTION PARAMETERS (SENSOR-2)

		Sequential		
Parameter	Batch	Q	No-Q	No-Refit
f (mm)	16.031	16.035	16.039	16.035
b_4 ($\times 10^8$)	1.370	0.583	0.320	1.374
b_2 ($\times 10^3$)	-2.152	-1.771	-1.662	-2.144

C. Overall Performance

To summarize the overall findings we consider the following observations. On-orbit calibrations outperformed ground calibrations in all respects (see Table V and Table IV). Comparing the performance of the batch and online parameter estimates yields no clear winner — the errors are typically within a few percent of one another and the availability and matched-star counts are virtually identical.

In most cases, the introduction of process noise (i.e., the Q trials) does not offer significant improvements in error or availability over the No-Q implementation. However, using a small amount of process noise could yield better response when parameter tracking is deemed necessary. Tuning the $\mathbf{Q}$ matrix may yield better performance than that reported here, but systematic errors in the observed scenes

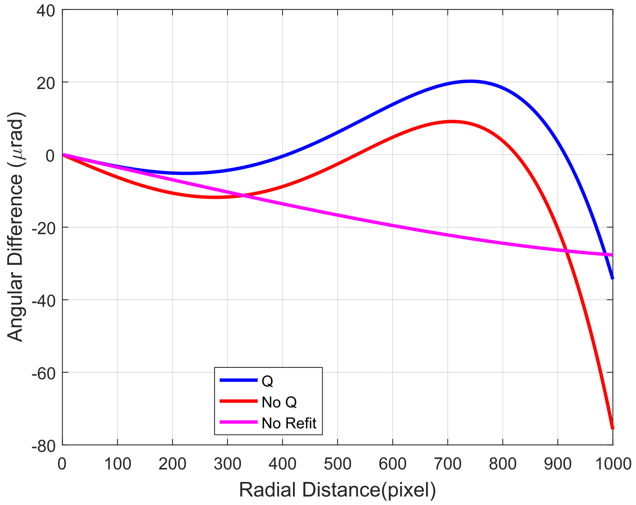


Fig. 5. Radial distortion differences between sequential estimates and batch results (Sensor-3).

appear to be the largest contributors to wandering parameter values. Further work is necessary to help distinguish between real physical parameter changes and other systematic errors (e.g., partially uncorrected stellar aberration or parallax, unresolved binaries, etc.). A simple, pragmatic approach to minimizing systematic effects may be to flag certain stars in the onboard catalog as less desirable to use in the online recalibration.

IV. Conclusions

In this study we have presented a variety of effective methods for improving star tracker calibration parameters using on-orbit observations. Our error model adapts the parameterization proposed by Wang *et al.* [2] for use with the star-field images captured on-orbit. The baseline, batch recalibration approach using ground-based optimization has proven effective in tuning performance of the six sensors for which we have orbital data. This recalibration procedure yields improvements in availability and matched stars as well as sizable reductions in arclength error ($30-70\%$, typically). The sequential estimator results show great promise, performing as well as the batch methods with minimal need to tune the measurement and process noise covariance.

Although sensor batch or sequential recalibration is not entirely new, this study makes several important contributions. Our model builds on the improved linear independence provided by Wang's parameter selection and further improves numerical conditioning by shifting and scaling parameters. The resulting formulation provides better distortion parameter observability than many traditional approaches. Additionally we provide analytic expressions for the measurement covariance matrix, a detail not often discussed. Furthermore, formulating the cost function in terms of arclength rather then the cosine of arclength avoids inadvertent and arbitrary measurement weighting. The calibrations are robust to starting conditions and avoid the need implement multi-step optimizations. As long as the initial calibrations are good enough for some successful matches, these optimizers can correct significant errors in the initial parameter values. Reliance on orbital data collected from different sensors adds significant weight to the this study's results. This approach avoids the inevitable optimism of simulation-based results and gives further confidence that the study results are not particular to a single unit.

Some results from the sequential estimator bear special mention. We are currently in the process of migrating the sequential estimator from the laboratory to an embedded implementation that will run on the ST-16 itself. This study has yielded valuable insights that can be used to minimize any risks involved with deploying autonomous algorithms in flight software. First, we note that the majority of our availability gains are made in the first few processed telemetry frames, but error reduction requires significantly more time to settle. This can help us devise operational procedures that can provide a quick turn-around in improving sensor availability (often of top priority during on-orbit commissioning), but allow the error to improve over a longer period. Second, constantly rematching with the updated parameter values seems to ensure the best performance, but may carry some risk. We can mitigate this risk at the expense of computational cost by duplicating the star-matching and attitude solution calculations with both the original parameters and the updated estimates.

In addition to the embedded deployment of these algorithms, some unanswered questions remain. The Q trials with the sequential filters have indicated time-varying parameter values. We are investigating several strategies to reduce the effect of any systematic errors and improve the ability of the filter to track slow changes in parameter values.

The autonomous algorithms prototyped in this study rely on EKF-based estimators. Other estimators such as Unscented Kalman Filters (UKF) frequently demonstrate improved convergence than EKFs for nonlinear systems. This performance gain must be balanced against increased computational requirements. If the ST-16 does not have enough processing capability to compute UKF updates every frame, then a tradeoff study would be necessary to determine the relative benefits of full-rate processing with an EKF against less frequent updates with a UKF. The optimal solution is not self-evident, but well worth investigating.

This study validates its algorithms using data from ST-16 star trackers but there is little reason to believe that these methods cannot be employed on other sensors. The individual behaviors — e.g., common parameter variations, etc. — may change, but the estimator will likely work just as well with other sensor designs and processing routines.

Appendix A
Covariance Invertibility

It is straightforward to show that $\mathbf{R}$ is both symmetric and positive definite (SPD), and hence, is invertible. Consider that for any SPD matrix, $\mathbf{B}$, it follows that:

$$\mathbf{a}^T\mathbf{B}\mathbf{a} > 0, \quad \forall\ \mathbf{a} \neq \mathbf{0} \tag{34}$$

Now from Section II-B, we can bring $\mathbf{a}$ inside the expectation. Thus

$$\mathbf{a}^T\mathbf{R}\mathbf{a} = \mathrm{var}\left(\mathbf{a}^T\delta\boldsymbol{\varphi}\right) \geq 0 \tag{35}$$

That is, $\mathbf{R}$ as a covariance matrix, must be positive semi-definite. Equality is only possible with zero variance, i.e.,

$$\mathbf{a}^T \boldsymbol{\delta\varphi} = c \tag{36}$$

for some constant, c. This, in turn, can only be satisfied if some elements of $\boldsymbol{\delta\varphi}$ are linearly dependent. The linear independence of the $\delta\varphi_{ij}$ elements follows directly from our construction rule. The only way for arcs to be linearly dependent would be for three arcs to be collinear, and this is expressly disallowed. Thus, $\mathbf{R}$ is SPD and hence, $\mathbf{R}^{-1}$ exists.

Appendix B
Model Jacobian

All of the estimators described in this paper depend on the Jacobians for the arclength errors. Restating Section II-C:

$$\mathbf{H}_{\text{batch}} = \frac{\partial \tilde{\boldsymbol{\varphi}}}{\partial \mathbf{x}} \tag{37}$$

Each star-pair arclength, φ_{ij}, contributes a row to $\mathbf{H}_{\text{batch}}$ (we have omitted the frame of reference subscript, D):

$$\frac{\partial \delta\varphi_{ij}}{\partial \mathbf{x}} = -\frac{1}{\sin\varphi_{ij}} \left\{ \frac{\partial \mathbf{s}_i}{\partial \mathbf{x}}^T \mathbf{s}_j + \frac{\partial \mathbf{s}_j}{\partial \mathbf{x}}^T \mathbf{s}_i \right\}^T \tag{38}$$

Evaluating $\mathbf{H}_{\text{batch}}$ using Section IV requires calculating intermediate Jacobians of the unit vectors. These matrices can be written in column-form as:

$$\frac{\partial \mathbf{s}_i}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{s}_i}{\partial f} & \frac{\partial \mathbf{s}_i}{\partial m_0} & \frac{\partial \mathbf{s}_i}{\partial n_0} & \frac{\partial \mathbf{s}_i}{\partial b_4} & \frac{\partial \mathbf{s}_i}{\partial b_2} & \frac{\partial \mathbf{s}_i}{\partial a_1} & \frac{\partial \mathbf{s}_i}{\partial a_2} & \frac{\partial \mathbf{s}_i}{\partial g_y} \end{bmatrix} \tag{39}$$

the components of which depend on the partials, $\partial \mathbf{r}/\partial x_q$.

If we have an expression for a unit vector $\mathbf{s}$ in terms of an un-normalized vector $\mathbf{r}$, i.e., $\mathbf{s} \equiv \mathbf{r}/r$, then the partial derivative of $\mathbf{s}$ with respect to state variable x_q is a vector given by the following expression

$$\frac{\partial \mathbf{s}}{\partial x_q} = \frac{1}{r^2}\left[r\frac{\partial \mathbf{r}}{\partial x_q} - \mathbf{s}\left\{ \frac{\partial \mathbf{r}}{\partial x_q}^T \mathbf{r} \right\} \right] \tag{40}$$

Thus the partials derivatives that make contribute to Section IV can be derived for each parameter. Starting with focal length, f, we have:

$$\frac{\partial \mathbf{r}}{\partial f} = \begin{bmatrix} \frac{\partial B}{\partial f}U + \frac{\partial U}{\partial f}B \\ \frac{\partial B}{\partial f}V + \frac{\partial V}{\partial f}B \\ 1 \end{bmatrix} \tag{41}$$

$$\frac{\partial B}{\partial f} = \left(4b_4\rho^3 + 2b_2\rho\right)\frac{\partial \rho}{\partial f} \tag{42}$$

$$\frac{\partial \rho}{\partial f} = \frac{1}{\rho}\left(U\frac{\partial U}{\partial f} + V\frac{\partial V}{\partial f}\right) \tag{43}$$

$$\frac{\partial V}{\partial f} = \frac{v(a_1v + a_2u)}{(a_1v + a_2u + f)^2} \tag{44}$$

$$\frac{\partial U}{\partial f} = \frac{u(a_1v + a_2u)}{(a_1v + a_2u + f)^2} \tag{45}$$

Similarly for m_0:

$$\frac{\partial \mathbf{r}}{\partial m_o} = \begin{bmatrix} \frac{\partial B}{\partial m_0}U + \frac{\partial U}{\partial m_0}B \\ \frac{\partial B}{\partial m_o}V + \frac{\partial V}{\partial m_0}B \\ 0 \end{bmatrix} \tag{46}$$

$$\frac{\partial B}{\partial m_0} = \left(4b_4\rho^3 + 2b_2\rho\right)\frac{\partial \rho}{\partial m_0} \tag{47}$$

$$\frac{\partial \rho}{\partial m_0} = \frac{1}{\rho}\left(U\frac{\partial U}{\partial m_0} + V\frac{\partial V}{\partial m_0}\right) \tag{48}$$

$$\frac{\partial V}{\partial m_o} = \frac{-a_2vf\frac{\partial u}{\partial m_0}}{(a_1v + a_2u + f)^2} \tag{49}$$

$$\frac{\partial U}{\partial m_0} = \frac{f(a_1v + f)\frac{\partial u}{\partial m_0}}{(a_1v + a_2u + f)^2} \tag{50}$$

$$\frac{\partial u}{\partial m_0} = -\Delta x \tag{51}$$

For n_0 :

$$\frac{\partial \mathbf{r}}{\partial n_0} = \begin{bmatrix} \frac{\partial B}{\partial n_0}U + \frac{\partial U}{\partial n_0}B \\ \frac{\partial B}{\partial n_0}V + \frac{\partial V}{\partial n_0}B \\ 0 \end{bmatrix} \tag{52}$$

$$\frac{\partial B}{\partial n_0} = \left(4b_4\rho^3 + 2b_2\rho\right)\frac{\partial \rho}{\partial n_0} \tag{53}$$

$$\frac{\partial \rho}{\partial n_0} = \frac{1}{\rho}\left(U\frac{\partial U}{\partial n_0} + V\frac{\partial V}{\partial n_0}\right) \tag{54}$$

$$\frac{\partial V}{\partial n_0} = \frac{f(a_2u + f)\frac{\partial v}{\partial n_0}}{(a_1v + a_2u + f)^2} \tag{55}$$

$$\frac{\partial U}{\partial n_0} = \frac{-a_1uf\frac{\partial v}{\partial n_0}}{(a_1v + a_2u + f)^2} \tag{56}$$

$$\frac{\partial v}{\partial n_o} = -g_y\Delta y \tag{57}$$

For b_4:

$$\frac{\partial \mathbf{r}}{\partial b_4} = \begin{bmatrix} \frac{\partial B}{\partial b_4}U \\ \frac{\partial B}{\partial b_4}V \\ 0 \end{bmatrix} \tag{58}$$

$$\frac{\partial B}{\partial b_4} = \rho^4 \tag{59}$$

For b_2:

$$\frac{\partial \mathbf{r}}{\partial b_2} = \begin{bmatrix} \frac{\partial B}{\partial b_2}U \\ \frac{\partial B}{\partial b_2}V \\ 0 \end{bmatrix} \tag{60}$$

$$\frac{\partial B}{\partial b_2} = \rho^2 \tag{61}$$

For a_1:

$$\frac{\partial \mathbf{r}}{\partial a_1} = \begin{bmatrix} \frac{\partial B}{\partial a_1} U + \frac{\partial U}{\partial a_1} B \\ \frac{\partial B}{\partial a_1} V + \frac{\partial V}{\partial a_1} B \\ 0 \end{bmatrix} \tag{62}$$

$$\frac{\partial B}{\partial a_1} = \left(4 b_4 \rho^3 + 2 b_2 \rho\right) \frac{\partial \rho}{\partial a_1} \tag{63}$$

$$\frac{\partial \rho}{\partial a_1} = \frac{1}{\rho}\left(U \frac{\partial U}{\partial a_1} + V \frac{\partial V}{\partial a_1}\right) \tag{64}$$

$$\frac{\partial U}{\partial a_1} = \frac{-v u f}{(a_1\, v + a_2\, u + f)^2} \tag{65}$$

$$\frac{\partial V}{\partial a_1} = \frac{-v^2 f}{(a_1\, v + a_2\, u + f)^2} \tag{66}$$

For a_2:

$$\frac{\partial \mathbf{r}}{\partial a_2} = \begin{bmatrix} \frac{\partial B}{\partial a_2} U + \frac{\partial U}{\partial a_2} B \\ \frac{\partial B}{\partial a_2} V + \frac{\partial V}{\partial a_2} B \\ 0 \end{bmatrix} \tag{67}$$

$$\frac{\partial B}{\partial a_2} = \left(4 b_4 \rho^3 + 2 b_2 \rho\right) \frac{\partial \rho}{\partial a_2} \tag{68}$$

$$\frac{\partial \rho}{\partial a_2} = \frac{1}{\rho}\left(U \frac{\partial U}{\partial a_2} + V \frac{\partial V}{\partial a_2}\right) \tag{69}$$

$$\frac{\partial U}{\partial a_2} = \frac{-u^2 f}{(a_1\, v + a_2\, u + f)^2} \tag{70}$$

$$\frac{\partial V}{\partial a_2} = \frac{-v\ u\ f}{(a_1\, v + a_2\, u + f)^2} \tag{71}$$

For g_y:

$$\frac{\partial \mathbf{r}}{\partial g_y} = \begin{bmatrix} \frac{\partial B}{\partial g_y} U + \frac{\partial U}{\partial g_y} B \\ \frac{\partial B}{\partial g_y} V + \frac{\partial V}{\partial g_y} B \\ 0 \end{bmatrix} \tag{72}$$

$$\frac{\partial B}{\partial g_y} = \left(4 b_4 \rho^3 + 2 b_2 \rho\right) \frac{\partial \rho}{\partial g_y} \tag{73}$$

$$\frac{\partial \rho}{\partial g_y} = \frac{1}{\rho}\left(U \frac{\partial U}{\partial g_y} + V \frac{\partial V}{\partial g_y}\right) \tag{74}$$

$$\frac{\partial U}{\partial g_y} = \frac{-a_1 u f \frac{\partial v}{\partial g_y}}{(a_1 v + a_2 u + f)^2} \tag{75}$$

$$\frac{\partial V}{\partial g_y} = \frac{f (a_2 u + f) \frac{\partial v}{\partial g_y}}{(a_1 v + a_2 u + f)^2} \tag{76}$$

$$\frac{\partial v}{\partial g_y} = \Delta y \left(n_c - n_0\right) \tag{77}$$

Careful implementation can exploit common terms to eliminate redundant calculations.

Appendix C
Carlson-Schmidt Square Root Filter

If $\mathbf{P}$ represents the state covariance, we begin by taking the Cholesky factor, $\mathbf{S}$ satisfying

$$\mathbf{P} = \mathbf{S}\mathbf{S}^T \tag{78}$$

The observation vector $\boldsymbol{\delta\varphi}_k$ is formed from the two image exposures in the k-th telemetry frame. We can show that

$$\mathbf{H}_k = \left.\frac{\partial \boldsymbol{\delta\varphi}_k}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k} \tag{79}$$

Carlson's update procedure assumes the sequential processing of scalar measurements. Our arclength measurements are not independent and must be decorrelated before use. We can employ a Cholesky factorization of the measurement covariance matrix, $\mathbf{R}_k$ to transform the errors and measurement matrices. The measurement covariance, $\mathbf{R}_k$ from a single telemetry frame can be factored as

$$\mathbf{R}_k = \Xi_k \Xi_k^T \tag{80}$$

To decorrelate the measurements we must solve the two triangular systems

$$\Xi_k \boldsymbol{\delta\varphi}'_k = \boldsymbol{\delta\varphi}_k \tag{81}$$

$$\Xi_k \mathbf{H}'_k = \mathbf{H}_k \tag{82}$$

We summarize the filter equations in a manner similar to Potter's original square root formulation, but implement the calculations using Carlson's fast triangular update [25]

$$\widehat{\mathbf{x}}^{+} = \widehat{\mathbf{x}}^{-} + \frac{\mathbf{b}\delta\varphi'_{ij}}{\alpha} \tag{83}$$

$$\mathbf{S}^{+} = \mathbf{S}^{-}\left(\mathbf{I} - \mathbf{f}\mathbf{f}^T\right)^{1/2} \tag{84}$$

where

$$\mathbf{b} = \mathbf{S}\mathbf{f} \tag{85}$$

$$\alpha = r + \mathbf{f}^T\mathbf{f} \tag{86}$$

$$\mathbf{f} = \mathbf{S}\mathbf{h}'_{ij} \tag{87}$$

In these relations, the matrix $\mathbf{h}'_{ij}$ is a single row taken from $\mathbf{H}'_k$; the error $\delta\varphi'_{ij}$ in Section IV is a single element of $\boldsymbol{\delta\varphi}'_k$; and $r = 1$ as a consequence of the factorization of $\mathbf{R}_k$.

We assume that the model parameters are constants, so there are no dynamics to affect state propagation from measurement to measurement. Artificial process noise can be added to prevent premature convergence of the filter. If the conventional covariance is represented by $\mathbf{Q} = \mathbf{M}\mathbf{M}^T$, the updates for our SQRF estimator are:

$$\mathbf{S}^{-}_{k+1} = \mathbf{S}^{+}_{k} + \mathbf{M}_k \tag{88}$$

Acknowledgments

The authors would like to thank Mr. Doug Sinclair from Sinclair Interplanetary for providing engineering model ST-16s for research purposes. Mr. Sinclair was also instrumental in arranging for flight data collection.

References

[1] C. C. Liebe, "Accuracy performance of star trackers—A tutorial," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 587–599, Apr. 2002.

[2] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognit.*, vol. 41, no. 2, pp. 607–615, 2008.

[3] J. Weng, P. Cohen, and M. Herniou, “Camera calibration with distortion models and accuracy evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.

[4] J. Sedlak, G. Welter, and N. Ottenstein, “Towards automating spacecraft attitude sensor calibration,” in *Proc. 54th Int. Astron. Congr. Int. Astron. Fed.*, Bremen, Germany, 2003, pp. 29–30.

[5] M. E. Pittelkau, “Autonomous on-board calibration of attitude sensors and gyros,” in *Proc. 20th Int. Symp. Space Flight Dyn.*, Annapolis, MD, USA, Sep. 2007.

[6] J. Enright, I. Jovanovic, and B. Vaz, “On-orbit star tracker recalibration: A case study,” in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2015, pp. 1–13.

[7] T. Sun, F. Xing, and Z. You, “Optical system error analysis and calibration method of high-accuracy star trackers,” *Sensors*, vol. 13, no. 4, pp. 4598–4623, 2013.

[8] X. Wei, G. Zhang, Q. Fan, J. Jiang, and J. Li, “Star sensor calibration based on integrated modelling with intrinsic and extrinsic parameters,” *Measurement*, vol. 55, pp. 117–125, Sep. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263224114001845

[9] D. S. Bayard and B. H. Kang, “A high-order Kalman filter for focal plane calibration of NASA’s space infrared telescope facility (SIRTF),” in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, 2003, pp. 11–14.

[10] A. Y. Lee and G. Hanover, “Cassini spacecraft attitude control system flight performance,” in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, San Francisco, CA, USA, Aug. 2005, p. 6269.

[11] T. Brady, S. Buckley, and M. Leammukda, “Space validation of the inertial stellar compass,” in *Proc. AIAA/USU Conf. Small Satell.*, Aug. 2007, Paper SSC07-X-6.

[12] M. A. Samaan, T. Griffith, P. Singla, and J. L. Junkins, “Autonomous on-orbit calibration of a star tracker,” in *Proc. Core Technol. Space Syst. Conf. Commun. Navigat. Session*, 2001.

[13] P. Singla, D. T. Griffith, J. Crassidis, and J. Junkins, “Attitude determination and autonomous on-orbit calibration of star tracker for GIFTS mission,” *Adv. Astron. Sci.*, vol. 112 Part I, pp. 19–38, 2002.

[14] D. T. Griffith, P. Singla, and J. L. Junkins, “Autonomous on-orbit calibration approaches for star tracker cameras,” *Adv. Astron. Sci.*, vol. 112, pp. 39–57, Jan. 2002.

[15] M. Pal and M. S. Bhat, “Star camera calibration combined with independent spacecraft attitude determination,” in *Proc. Amer. Control Conf. (ACC)*, Jun. 2009, pp. 4836–4841.

[16] H.-B. Liu, J.-Q. Wang, J.-C. Tan, J.-K. Yang, H. Jia, and X.-J. Li, “Autonomous on-orbit calibration of a star tracker camera,” *Opt. Eng.*, vol. 50, no. 2, p. 023604, 2011.

[17] J. Shen, G. Zhang, and X. Wei, “Star sensor on-orbit calibration using extended Kalman filter,” in *Proc. 3rd Int. Symp. Syst. Control Aeronaut. Astronaut.*, Jun. 2010, pp. 958–962.

[18] T. Dzamba *et al.*, “Success by 1000 improvements: Flight qualification of the ST-16 star tracker,” in *Proc. AIAA/USU Conf. Small Satell.*, Aug. 2014, Paper SSC14-XII-1.

[19] J. Enright and T. Dzamba, “Measurement weighting strategies for satellite attitude estimation,” in *Proc. IEEE Aerosp. Conf.*, Mar. 2014, pp. 1–8.

[20] T. Dzamba and J. Enright, “Ground testing strategies for verifying the slew rate tolerance of star trackers,” *Sensors*, vol. 14, no. 3, pp. 3939–3964, 2014.

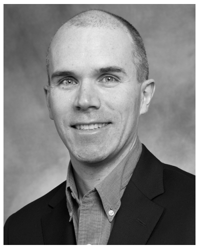
[21] J. Enright and T. Dzamba, “Rolling shutter compensation for star trackers,” in *Proc. AIAA Guid., Navigat., Control Conf.*, Minneapolis, MN, USA, Aug. 2012, Paper AIAA 2012-4839.

[22] T. Dzamba, J. Enright, and D. Sinclair, “Characterizing chromatic effects in small star trackers,” in *Proc. AIAA/USU Conf. Small Satell.*, Aug. 2016, Paper SSC16-P2-01. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2016/Poster2/14

[23] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, “Averaging quaternions,” *J. Guid., Control, Dyn.*, vol. 30, no. 4, pp. 1193–1196, 2007.

[24] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*. Hoboken, NJ, USA: Wiley, 2001, pp. 239–244.

[25] N. A. Carlson, “Fast triangular formulation of the square root filter,” *AIAA J.*, vol. 11, no. 9, pp. 1259–1265, 1973. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/3.6907



John Enright (M’05) received the B.A.Sc. degree in engineering science (aerospace) from the University of Toronto in 1997 and the M.S. and Ph.D. degrees in aerospace systems from MIT, in 1999 and 2002, respectively. He is currently an Associate Professor of Aerospace Engineering with Ryerson University, Toronto. He is the Principal Investigator of the Space Avionics and Instrumentation Laboratory (SAIL). Dr. Enright and his SAIL researchers have demonstrated expertise in instrument calibration, attitude estimation, and sensor processing algorithms. His research interests include spacecraft avionics and sensor processing, systems engineering, and flight software. He is a member of AIAA and CASI.

Ilija Jovanovic is currently pursuing the Ph.D. degree in the field of aerospace engineering with Ryerson University. His research focuses on the calibration of star trackers, computer modeling of pulsar navigation systems, and nighttime planetary navigation systems.

Brendon Vaz received the B.Eng. and M.A.Sc. degrees in aerospace engineering from Ryerson University, in 2011 and 2013, respectively. While working towards the M.A.Sc. degree, he conducted research on On-Orbit Calibration Techniques for Star Trackers at Ryerson’s Space Avionics and Instrumentation Laboratory (SAIL). His research interests include non-linear estimation, controls, and optimization.