SPACE OBJECT MANEUVER DETECTION ALGORITHMS USING TLE DATA

Dr. Mark Pittelkau
Solers, Inc.

ABSTRACT

An important aspect of Space Situational Awareness (SSA) is detection of deliberate and accidental orbit changes of space objects. Although space surveillance systems detect orbit maneuvers within their tracking algorithms, maneuver data are not readily disseminated for general use. However, two-line element (TLE) data is available and can be used to detect maneuvers of space objects. This work is an attempt to improve upon existing TLE-based maneuver detection algorithms. Three adaptive maneuver detection algorithms are developed and evaluated:

The first is a fading-memory Kalman filter, which is equivalent to the sliding-window least-squares polynomial fit, but computationally more efficient and adaptive to the noise in the TLE data.

The second algorithm is based on a sample cumulative distribution function (CDF) computed from a histogram of the magnitude-squared $|\Delta\mathrm{V}|^2$ of change-in-velocity vectors (ΔV), which is computed from the TLE data. A maneuver detection threshold is computed from the median estimated from the CDF, or from the CDF and a specified probability of false alarm.

The third algorithm is a median filter. The median filter is the simplest of a class of nonlinear filters called order statistics filters, which is within the theory of robust statistics. The output of the median filter is practically insensitive to outliers, or large maneuvers. The median of the $|\Delta\mathrm{V}|^2$ data is proportional to the variance of the ΔV, so the variance is estimated from the output of the median filter. A maneuver is detected when the input data exceeds a constant times the estimated variance.

1. INTRODUCTION

One of the key operations in space situational awareness is space event monitoring, one function of which is timely detection of maneuvers of orbiting objects. Although most satellites execute maneuvers for orbit maintenance, some execute maneuvers outside of their normal or anticipated operations. Some objects thought to be inert have unexpectedly “come alive” and others thought to be placed in orbit for a particular purpose change their orbit to serve another purpose. Maneuvers can also result from system failure or operator error and present a hazard. On another extreme, a collision is detected as a maneuver. A false detection of a maneuver can occur because of tracking and data association (cross-tagging) error.

The purpose of this paper is to define and analyze algorithms to process two-line-element (TLE) data to detect maneuvers of space objects, and to improve upon the performance of existing algorithms.

There are essentially two main categories of maneuver detection algorithms: tracking filters and polynomial-fit algorithms. Tracking filters estimate orbital motion from tracking sensor data and can be sensitive maneuver detectors, particularly when gravity and perturbations are modeled accurately. Unfortunately the tracking data are not readily available. However, space surveillance tracking systems provide only two-line element (TLE) data to most end-users. The TLE data are readily available from Space Track [1], which is operated by the Joint Space Operations Center (JSpOC), and from CelesTrak [2].

Most of the published maneuver detection algorithms compute a least-squares fit of a low-order polynomial to a sliding window of orbital parameters derived from TLE data, typically semimajor axis and inclination [19–26]. These algorithms require manual adjustment of parameters (window length, polynomial order, and threshold) for each space object to obtain acceptable results. They exhibit an inherent lag, low probability of detection, and high false-alarm rate. The polynomial algorithms require some information about the error in the TLE data, which is computed from a long span (a year or more) of data, though details are not published. A second algorithm [26] removes trends and harmonics from the TLE data prior to detecting maneuvers. The algorithm in [27] is a second-order filter with an elaborate treatment of residuals and a computationally intensive startup procedure.

The least-squares polynomial-fitting algorithms require significant manual adjustment of parameters (number of TLEs fitted, polynomial order, and threshold) for each space object to obtain acceptable results. A fixed set of parameters reported in [20, 25] were obtained by adjusting the parameters and comparing maneuver detection results for two satellites against a history of known maneuvers. The fixed parameters result in a large number (13%) of false alarms and missed detections (17%) [20, 25]. It is not clear which or how many space objects with known maneuver histories were used to determine the false alarm rate and missed detection rate, and the detection threshold was not stated.[1]

Some desired capabilities of a maneuver detection algorithm are as follows:

1. The maneuver detection algorithm should be computationally efficient to process daily TLE updates, and reprocess corrected TLE data, for a single space object within a few seconds.
2. The maneuver detection algorithm should be able to initialize itself in a few minutes for all objects in the TLE catalog.
3. The algorithm can be started for any object at any time.
4. The algorithm should run autonomously. Therefore it should require little to no re-tuning, and the tuning should work for all objects at all times. Different tunings are permitted for a small number of classes of objects (GEO, HEO, MEO, LEO).
5. The false alarm rate and the missed detection rate should be small (ideally less than 1% (TBR) false alarm rate and less than 5% (TBR) missed detection rate).

Item 4 is important because the error statistics of the TLE data are nonstationary and can differ amongst various space objects. It is not practical to manually tweak the parameters of a maneuver detection algorithm as conditions change, and it is not practical to manually select parameters for each space object. Requirement 5 is particularly important because in a catalog of over 22 000 objects in which perhaps 10% of the objects are of particular interest, so an operator having to give special attention to only 1% (say) of the objects of interest is time consuming and tedious. In addition, an operator should not have to be an expert on maneuver detection algorithms.

An algorithm based on TLE data cannot be optimal (or even nearly optimal) because little is known (publicly) about how the TLE data are computed and how accurate they are. In addition, maneuver detection algorithms based on TLE data essentially filter data that are the output of a filter. The TLE data are estimated from either a continuous stream of measurement data or from overlapping intervals of measurement data, so there can be significant correlation of the errors in the TLE data. We can assume that upon detection of a maneuver, the TLE estimator (operated by JSPOC) detects maneuvers and "forgets" previous measurement data. For example, When a maneuver occurs, the orbit determination filter is re-initialized or its covariance is opened-up (increased) when a maneuver is detected. Errors in the TLE data are therefore sequentially correlated until a maneuver occurs, after which the correlation resumes. Characteristics of TLE data and some difficulties encountered are discussed in [26, §II.A].

[1] The performance results reported in [25] may be substantiated in [20], but [20] was not available when this article was written.

Despite the foregoing limitations of using TLE data to detect maneuvers, it appears there is still room for improvement upon existing TLE-based algorithms. In this paper we define and analyze algorithms to process two-line-element (TLE) data to detect maneuvers of space objects and to improve upon the performance and computational requirements of the algorithms cited above. Three new maneuver detection algorithms are introduced:

Fading-Memory Filter: The recursive fading-memory polynomial filter with a prescribed memory length is nearly equivalent to but computationally more efficient than the sliding-window fixed-memory least-squares algorithm. The fading-memory filter is adaptive to the noise in the data.

Histogram Method: A histogram of ΔV data derived from the TLE data approximates the probability distribution function of the data. Maneuvers are detected at the tail of the distribution with a probability threshold computed from the data. A fixed threshold can be used instead. The histogram gives significant insight into the statistical nature of the TLE data.

Median Filter: The median filter is one of a class of robust statistics filters and detects outliers in the ΔV data. It is faster than the polynomial filters, has fewer fiddle parameters, and is easier to “tune”.

The histogram method gives insight into the statistical nature of the data and is related to the median filter. A recursive fixed-memory or fading-memory histogram may be effective but was not investigated. The fading-memory filter and the median filter are temporally adaptive in that the detection threshold depends on the data as the data arrive. The histogram method is adaptive, but the detection threshold is not determined until all the data over a fixed window is processed.

When a maneuver is detected, the ΔV and the orbital elements can be examined to determine the nature of the maneuver, eg., orbit adjust or orbit change, and altitude or inclination change. Maneuver characterization is beyond the scope of this report.

1.1. Organization

The fading-memory polynomial filter in Section 2.1 and the median filter in Section 2.3 Test and evaluation are reported in Section 2.5 and Section 3. Results for a selection of space objects are reported in Section 3.

2. MANEUVER DETECTION ALGORITHMS

2.1. Fading-Memory Polynomial Filter

TLE-based maneuver detectors are typically based on fitting a short window of TLE elements (notably the semimajor axis data) to a low-order polynomial. The maneuver detection is based on comparing a residual with a threshold. The parameters chosen in [25] are a window length of 11 TLEs, order 3 polynomial, and a 3σ threshold. The residual is the difference in the element and the fitted element, either at the center or at the end of the interval. The threshold is a function of the error statistics of the TLE data. The error statistics are computed from the TLE data with some type of compensation for maneuvers. The polynomial fitting is performed by using batch least-squares fit to the data. The batch least-squares solution is recomputed for each new TLE data. The error statistics are typically compiled from the catalog of TLE data [19, 22–26].

The sliding-window fixed-memory least squares algorithm can be reformulated as a fixed-memory Kalman filter [4, Ch. 7, §10, p. 255–258]. However, it requires significant data storage and its numerical stability would have to be evaluated. A recursive fading-memory least-squares estimator with a Cholesky-factor update algorithm was initially considered and was found to perform about as well as a fixed-memory least-squares estimator. A fading-memory Kalman filter [4, Ch. 8, §10, pp. 307–311] is essentially equivalent but has computational advantages, and is simpler to implement. A factorized-covariance formulation of the fading-memory filter, which is equivalent to the Cholesky factor update algorithm for the fading-memory least-squares estimator, is not presented here. The fading-memory polynomial filter is described in what follows.

Define the state vector

$$\boldsymbol{x}_k = \begin{bmatrix} x_k^{(0)} \\ x_k^{(1)} \\ \vdots \\ x_k^{(n)} \end{bmatrix} \tag{1}$$

where $x_k^{(i)}$ is the ith derivative of the orbital element $x(t)$ at time $t = t_k$ and let $\boldsymbol{y}_k$ be a "measurement" of an orbital element from the TLE data at epoch t_k. The filter equations are as follows.

$$\begin{aligned}
P_{k|k-1} &= \alpha_k \Phi_k P_{k-1|k-1} \Phi_k^T && \text{covariance prediction} && \text{(2a)} \\
V_k &= H_k P_{k|k-1} H_k^T + R_k && \text{residual covariance} && \text{(2b)} \\
K_k &= P_{k|k-1} H_k^T V_k^{-1} && \text{Kalman gain} && \text{(2c)} \\
\boldsymbol{z}_k &= \boldsymbol{y}_k - H_k \hat{\boldsymbol{x}}_{k-1|k-1} && \text{residual} && \text{(2d)} \\
\hat{\boldsymbol{x}}_{k|k-1} &= \Phi_k \hat{\boldsymbol{x}}_{k-1|k-1} && \text{state estimate prediction} && \text{(2e)} \\
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + K_k \boldsymbol{z}_k && \text{state estimate update} && \text{(2f)} \\
P_{k|k} &= P_{k|k-1} - K_k V_k K_k^T && \text{covariance update} && \text{(2g)}
\end{aligned}$$

The various matrices are

Φ_k	state transition matrix
H_k	observation matrix
V_k	residual covariance matrix
R_k	measurement covariance matrix
K_k	Kalman gain
$P_{k\|k-1}$	prior covariance matrix
$P_{k\|k}$	posterior covariance matrix

The state transition matrix for a second-order polynomial filter ($n = 2$) is

$$\Phi_k = \begin{bmatrix} 1 & T_k & \frac{1}{2}T_k^2 \\ 0 & 1 & T_k \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $T_k = t_k - t_{k-1}$ is the time between observations y_k. The observation matrix is constant,

$$H_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4}$$

The scalar α_k in (2a) is the fading factor (or forgetting factor) given by

$$\alpha_k = \exp(T_k/\tau) \tag{5}$$

where τ is the effective memory length (in the same units as T_k). It can be shown that when compared to a fixed-memory least-squares estimator with a memory length of N samples (with equal time steps) that $\tau \simeq N - 1/2$.

The measurement update is skipped when $T_k = 0$ because there is likely no new information in the TLE data. It is possible that the TLE was corrected and reissued. If that is the case, the state of the filter prior to the previous update should restored and the update should be made with the corrected TLE.

There is no process noise covariance in the covariance prediction equation, (2a). We can write (2a) as $P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + (\alpha_k - 1)\Phi_k P_{k-1|k-1} \Phi_k^T$ from which we obtain $Q_k = (\alpha_k - 1)\Phi_k P_{k-1|k-1} \Phi_k^T$. Since $\alpha_k > 1$, the multiplier $\alpha_k - 1 \geq 0$. We must guard against $\alpha_k = 1$, which happens when $T_k = 0$, since that would cause the subsequently updated covariance $P_{k|k}$ to become singular.

The state is initialized to

$$\hat{\boldsymbol{x}}_1 = \begin{bmatrix} \boldsymbol{y}_2 \\ (\boldsymbol{y}_2 - \boldsymbol{y}_1)/T_2 \\ \boldsymbol{0} \end{bmatrix} \tag{6}$$

with $T_2 = t_2 - t_1$. The first update of the filter following initialization is with the third measurement, unless $T_3 = 0$ in which case the third measurement is skipped. The covariance is initialized to

$$P_1 = \text{diag}\begin{bmatrix} R_1 & 2R_1/T_1^2 & (\sigma_a s^2/m)^2 \end{bmatrix} \tag{7}$$

where $\sigma_a = 0.0001$ m/s^2 is the standard deviation of the acceleration process noise, $s = 86400$ scales the time units from seconds to days, and $m = 1000$ scales the distance units from meters to km. When the measurement $\boldsymbol{y}_k$ is the semimajor axis from the TLE, which is given in km, the units of R_k are conveniently specified in meters but should be scaled to km. Note that the covariance of error between the first two states in $\boldsymbol{x}_1$ is zero. Normally we would want to calculate this, but the TLE data are already highly correlated in an unknown way. It is safer to over-estimate the covariance of error in the TLE data so that the filter does not become too optimistic.

A residual edit test is usually used to detect outliers in the measurements. In this case, we use the residual edit test to detect maneuvers. The residual edit test is

$$\chi^2 = z_k^T V^{-1} z_k > \kappa^2 \tag{8}$$

where $\kappa > 0$ is the test threshold. If the test is true, the state and covariance are reinitialized.

The measurement $\boldsymbol{y}_k$ is a scalar y_k and so the measurement covariance $R_k = \sigma_k^2$ is a scalar. The variance of the measurement error can be estimated by averaging the square residuals with an averaging filter

$$\hat{\sigma}_k^2 = \hat{\sigma}_k^2 + g(|z_k|^2 - \hat{\sigma}_k^2) \tag{9}$$

where g is a gain. This averaging step is skipped when a maneuver is detected via (8). The initial convergence rate of $\hat{\sigma}^2$ is improved, and the effect of an erroneous initial estimate mitigated, by computing the gain as

$$g = 1/\min(j\,, j_{\max}) \tag{10}$$

where j is the number of iterations since initialization of the filter and $j_{\max}$ sets the minimum gain. The initial value for $\hat{\sigma}_k$ should be larger than the anticipated error in the TLE element and should be chosen so that it works well for TLE data from a wide variety of space objects. This determination requires extensive experimentation with the data. Alternatively, we can compute σ^2 from the median of $|z_k|^2$, similar to the method in Section 2.3. This has not yet been tried with the fading memory filter.

The foregoing variance estimator turns the Kalman filter into an adaptive filter. In [19, 22–26] the variance is estimated by analyzing an entire record of TLE data and, evidently, is fixed for the duration of the data processing. The method of computing the variance is not clear in those publications.

2.2. Histogram Method

If the probability density function of the TLE errors were known, one could base a maneuver detection algorithm on a hypothesis test and establish a detection threshold based on specified probabilities of false alarm and missed detection. Even if the form of the probability density function were known, typically we do not know the parameters of the density function. In addition, the TLE errors are not stationary and outliers (due to maneuvers) make it difficult to determine the parameters of the density function. In lieu of exact knowledge of the probability density function, we can create a histogram of the data to approximate the probability density function (PDF) and similarly to approximate the cumulative distribution function (CDF). The CDF is approximated as the cumulative sum of the histogram that approximates the PDF. Theoretical statements regarding the accuracy of these approximations versus the number of samples are not provided in this report, but should be investigated.

Histograms of the position and velocity changes between TLE epochs are computed over the history of a space object to obtain a statistical characterization of the orbit prediction errors of the space object. Large maneuvers are identified as position and velocity changes in an upper percentile (say the 97th percentile) of the histograms.

The TLE data at epoch t_k is converted to an orbital position vector $\boldsymbol{r}_{k,k-1}$ and velocity vector $\boldsymbol{v}_{k,k-1}$ at the epoch t_{k-1} of a previous TLE.[2] The prediction algorithm includes a general perturbation model. The TLE data at the epoch t_{k-1} from the previous epoch t_{k-1} is converted to an orbital position vector $\boldsymbol{r}_{k-1,k-1}$ and velocity vector $\boldsymbol{v}_{k-1,k-1}$ at the epoch t_{k-1}. We then compute the magnitude-square of the change in position and velocity[3]

$$(\Delta r_k)^2 = |\Delta \boldsymbol{r}_k|^2 = |\boldsymbol{r}_{k,k-1} - \boldsymbol{r}_{k-1,k-1}|^2 \tag{11a}$$

$$(\Delta v_k)^2 = |\Delta \boldsymbol{v}_k|^2 = |\boldsymbol{v}_{k,k-1} - \boldsymbol{v}_{k-1,k-1}|^2 \tag{11b}$$

It turns out that a maneuver detection algorithm based on $(\Delta v_k)^2$ is more sensitive to maneuvers than one based on $(\Delta r_k)^2$, so we will focus on analyzing $(\Delta v_k)^2$ in what follows.

The histogram $\boldsymbol{N}(\ell)$, $1 \le \ell \le L$, of $(\Delta v_k)^2$ is computed over a range of bin edges $\boldsymbol{X}(\ell)$. Maneuvers then appear as counts in the bins above a specified percentile, for example, the 97th percentile. The last bin edge, $\boldsymbol{X}(L)$, is ideally less than the minimum detectable $(\Delta v_k)^2$. (The bin edge is the right edge of the bin. The left edge of the first bin is zero.) The tally $\boldsymbol{N}(\ell)$ in each bin is normalized by the total number K of samples to obtain the sample density function,[4]

$$f(\ell) = \boldsymbol{N}(\ell)/K, \qquad 1 \le \ell \le L \tag{12}$$

Observe that

$$K = \sum_{\ell=1}^{L} \boldsymbol{N}(\ell) \tag{13}$$

The sample probability distribution function, or CDF, is the cumulative sum of the $\boldsymbol{N}(\ell)$,

$$F(\ell) = \sum_{m=1}^{\ell} f(m) = F(\ell-1) + f(\ell), \qquad 1 \le \ell \le L, \quad F(0) = 0 \tag{14}$$

[2] It is assumed that the backward prediction is within the fit interval of the TLE data and therefore should be more accurate than a forward prediction.

[3] The position and velocity vectors are in the True-Earth Mean-Equator (TEME) reference frame [3], which differs from Earth-Centered Inertial coordinates (ECI J2000) coordinates by precession and nutation, which changes slowly and negligibly over the prediction interval. The TEME frame is therefore a quasi-inertial frame.

[4] More precisely, $f(\ell)$ is a probability mass function (PMF). To obtain a density, we should compute $f(\ell) = (N(\ell)/K)/\delta X(\ell)$, where $\delta X(\ell) = X(\ell) - X(\ell-1)$ is the bin size. But then we would have to multiply $f(\ell)$ by the bin size to compute the sample cumulative distribution function in (14).

Equivalently,

$$\boldsymbol{M}(\ell) = \sum_{m=1}^{\ell} \boldsymbol{N}(m) = \boldsymbol{M}(\ell - 1) + \boldsymbol{N}(\ell), \qquad 1 \le \ell \le L, \quad \boldsymbol{M}(0) = 0 \tag{15a}$$

$$F(\ell) = \boldsymbol{M}(\ell)/K, \qquad 1 \le \ell \le L \tag{15b}$$

The bin size is chosen to be sufficiently small to obtain a fairly smooth histogram, but not so small that the bins contain too few samples. The bins do not have to be uniformly spaced. The main difficulty in implementing the histogram method is in choosing the number of bins, bin edges, and the maximum bin, all of which are data-dependent. A one-size-fits-all set of parameters may be chosen for particular classes of space objects or perhaps for all space objects. The feasibility and success of this method has to be determined empirically by evaluating its performance for a variety of space objects.

Assume that $\boldsymbol{v}_{k,k}$ and $\boldsymbol{v}_{k,k-1}$ are the true velocity vectors plus independent Gaussian random error vectors with equal variances for each component of error. Then $(\Delta v_k)^2 = |\Delta \boldsymbol{v}_k|^2$ is a χ^2 random variable with $d = 3$ degrees of freedom (DOF).[5] Assume that each element of $\boldsymbol{v}_{k,k}$ has variance σ_0^2 and that each element of $\Delta \boldsymbol{v}_1$ has variance $\sigma_{k,k-1}^2$. Then assuming that the errors in $\boldsymbol{v}_{k,k}$ and in $\boldsymbol{v}_{k,k-1}$ are uncorrelated,[6] each element of $\Delta \boldsymbol{v}_k$ has variance $\sigma^2 = \sigma_0^2 + \sigma_1^2$. Then the mean value of $\chi^2 = (\Delta v_k)^2$ is

$$\mu = d\sigma^2 \tag{16}$$

the median is (to a close approximation)

$$m = d(1 - 2/9d)^3 \sigma^2 \tag{17}$$

and the variance of χ^2 is

$$\sigma_{\chi^2}^2 = 2d\sigma^4 \tag{18}$$

The variance is easily estimated from the sample mean or sample median once one of the latter is estimated. The mean is statistically more efficient to compute from the data than is the median. The mean of $(\Delta v_k)^2$ can be computed from the histogram as a discrete integration by multiplying each normalized bin value by the value of $(\Delta v_k)^2$ that it represents, and summing the results. The sample mean and sample median may not be representative of the actual mean or median for small sample sizes. The mean can be adversely influenced by outliers (too-frequent maneuvers or a few large maneuvers), and for this reason the median is preferred. The median is the bin location where the sample CDF is approximately 0.5, which is where there are approximately equal numbers of samples in lower and higher bins. The approximation is due to bin quantization and the finite (and usually small) number of samples.

Maneuvers are declared for any $(\Delta v_k)^2$ that lies above some specified threshold τ. The threshold can be computed by one of two methods. (1) Assuming that $(\Delta v_k)^2$ is chi-square distributed, the threshold can be computed as a multiple of the variance for some dimensionless multiplier κ,

$$\tau = \kappa\sigma^2 \tag{19}$$

where κ is chosen for a given probability of false alarm. This threshold is valid if the distribution of the data is close to chi-square distribution with known degrees of freedom. Excessive maneuver detections can occur when the estimated σ^2 is too small or the $(\Delta v_k)^2$ are chi-square distributed (in the absence of maneuvers,

[5] If the errors are independent, then $d = 6$, which would explain why the maneuver detection thresholds, when tested on real TLE data, had to be larger than originally anticipated. This will be examined in an update to this report.

[6] The errors in $\boldsymbol{v}_{k,k}$ and in $\boldsymbol{v}_{k,k-1}$ are not completely uncorrelated, but we do not know the covariance of error in the TLE data.

since the median is used to estimate σ^2). (2) For any probability distribution of $(\Delta v_k)^2$, a threshold can be computed from the sample CDF $F(\ell)$ at the bin value closest to a given probability P_τ of false alarm. Since $F(\ell)$ is a sample cumulative distribution, we find the bin $X(\ell)$ for which $F(\ell)$ is closest to P_τ,

$$\ell_\tau = \arg \min_{1 \leq \ell \leq L} |F(\ell) - P_\tau| \tag{20}$$

The corresponding independent variable is

$$\tau = X(\ell_\tau) \tag{21}$$

Computing the threshold from P_τ makes the histogram method independent of the form of the distribution of the data, but it also guarantees, on average, $\lfloor (1 - P_\tau)K \rfloor$ false maneuver detections.[7]

Other than the bin parameters, the multiplier κ or probability P_τ is the only tuning parameter in the histogram method. A typical value for P_τ is perhaps 0.97 (97th percentile), though a value somewhere between 0.85 and 0.98 may be appropriate for some space objects. A lower threshold increases the false alarm rate but also increases the probability of detection. For any fixed threshold, performance ultimately depends on the noise level in the data. The histogram method works well when the maneuvers are sufficiently distinct from the noise in the TLE data. Since the threshold is computed from the entire record of data, the false alarm rate and probability of detection can vary when the errors in the TLE data are not stationary, that is, the statistics of the data vary over time.

The histogram is easily updated as new observations arrive. A sliding window histogram can be implemented by reducing bin counts as old data exit the window, then re-normalizing to obtain a distribution. It is possible to implement a fading-memory histogram in the same manner as the fading-memory filter. Much greater efficiency may be gained by realizing that only the median and upper percentiles need to be computed. An algorithm similar to the one in [5] may be suitable for this purpose.

2.3. Median Filter

Significant drawbacks to sliding-window least-squares estimation or Kalman filtering of TLE data for maneuver detection are that the TLE data are already filtered quantities and we do not know the accuracy and correlation of the TLE data. A filter has to be run for each TLE element of interest, and each such filter requires its own tuning parameters. A single tuning does not work equally well for all space objects and may require operator intervention.

The histogram method is limited in performance as explained in the previous section, but can be modified to operate as a fading-memory statistic of the data. The median filter in this section is computationally fast and requires only choosing a window width and a threshold, and it is adaptive to non-stationary statistics of the TLE data. The median filter is used in a variety of signal processing applications. It is commonly used in image processing to remove speckles from an image. It was also used in the Space Shuttle's redundant IMU processing algorithm to reject measurements from a possibly failed gyro.

The median filter is the best known filter in a class of filters based on order statistics, which is a branch of robust statistics [11]. The median filter is widely used in signal processing, image processing, and data analysis in various other fields. Its wide applicability is due to its relative immunity from influence of outliers in the data. In this section we apply the median filter to the TLE data to detect maneuvers of space objects.

Some further background and mathematical analysis of the median filter is given in Section 2.3.1 followed by its application to maneuver detection in Section 2.3.2. The performance of the median filter is verified and characterized in Section 2.5.

[7] The false alarms can be mitigated by ignoring the detections when the corresponding $(\Delta v_k)^2$ is below some fixed threshold. This check was not implemented in the histogram method prior to production of results reported in Section 3. A combination of the two thresholds, (20) and (21) may be beneficial.

2.3.1. Background

The median filter was introduced by J. W. Tukey in 1963 [6] but remained unpublished until 1974 [7, Ch. 7 & 16], [8, p. 210] to smooth noisy time-series data, and it was subsequently used to remove impulsive noise from signals and images. Statistical and deterministic analyses of the median filter are given in [9]. An overview of these and other properties of the median filter are given in [10–12] and in the numerous references therein. Some pertinent results are presented here.

The median filter is a nonlinear fixed memory-length filter that selects the middle value y_k from a sliding window of N data values $x_{k-N+1}, \ldots, x_{k-1}, x_k$,

$$y_k = \text{median}(x_{k-N+1}, \ldots, x_{k-1}, x_k) \tag{22}$$

We require N to be odd to simplify analysis and implementation of the median filter. If N were even, the median would be the arithmetic mean (the average) of the two middle values, which incurs additional computation. Using the lower or upper median can bias the average of the medians from the sliding window over the data, but would be acceptable for a large window. Thus we can write $N = 2n + 1$ where $n \geq 1$. For N odd, y_k is one of the input samples. The median filter can be viewed as a type of sliding-window histogram, and in fact some median filters for image processing are implemented as histograms.

Suppose we have a sequence of data $x_1, x_2, \ldots, x_N$ (N odd or even here) and we want to estimate a value ρ that is "closest" in some sense to the data [9]. Define a distance measure, or cost function,

$$D^r(\rho) = \sum_{i=1}^{N} a_i |x_i - \rho|^r \tag{23}$$

where the a_i are weighting coefficients such that $a_i > 0$. The optimal value for ρ is that which minimizes $D^r(\rho)$. In what follows we will set $a_i = 1/N$, so the a_i can be effectively omitted from the distance measure.

It is well known that for $r = 2$ the distance measure $D^2(\rho)$ is the L_2-norm square (the sum of square errors). The L_2-norm is minimized by the Minimum Mean Square Error (MMSE) estimate

$$\rho_2 = \arg\min_{\rho} D^2(\rho) = \text{mean}(x_1, x_2, \ldots, x_N) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{24}$$

This is the maximum likelihood (ML) estimate for Gaussian data, for which the probability density function is

$$f_2(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{25}$$

where μ is the mean and σ^2 is the variance of the data. For $r = 1$, the distance measure $D^1(\rho)$ is the L_1-norm (also known as the Manhattan Distance). The L_1-norm is minimized by the Minimum Absolute Error (MAE) estimate

$$\rho_1 = \arg\min_{\rho} D^1(\rho) = \text{median}(x_1, x_2, \ldots, x_N) \tag{26}$$

This is the ML estimate for Laplacian data, for which the probability density function is the biexponential density function

$$f_1(x) = \frac{1}{2\sigma^2} \exp\left(-\frac{\sqrt{2}}{\sigma}|x-\mu|\right) \tag{27}$$

Typically the distribution is not known, or is a mixture of a known density function (perhaps with unknown parameters) and an unknown density function of additive impulsive noise. For example, the data may be

Gaussian plus impulsive noise, so the density function of the data is a Gaussian density with a heavy tail. The tail may be modeled as a lumped probability mass.

Although the sample mean is a statistically more efficient estimator than the sample median, the sample mean is sensitive to outliers, whereas the sample median is practically insensitive to outliers. The median filter effectively "clips" the tail of the input distribution. It is shown in [9] that the MAE estimator is more robust to large outliers (impulses), and therefore less sensitive to the tail of the distribution, than the MMSE estimator. The median filter has a breakdown point of 50%, which means that the median becomes unreliable when more than 50% of the data are impulses. When 50% of the data are impulses, the probability of missed detection equals the impulse rate for all window widths of the median filter. For lower impulse rates the median is more reliable as the window width increases [11].

For any input distribution, the output distribution is given by [9, Eqs. (10), (11)] and [11, Eqs. (9), (10)], [12, Eqs. (3), (4)]. Weighted median filters [9, 11, 12] exist for unequal a_i, and of course the weighted MMSE estimator is well known. Other statistical and deterministic properties of median filters are given in [9] and [11, §II.A]. One notable deterministic property is that the impulse response of the median filter is zero.

Other types of filters designed to remove impulsive noise, many of which are related to the median filter, are discussed in [9–11, 13, 14]. These filters are primarily designed for image processing but can be applied to other types of signals. Some of these algorithms are impressive in their ability to remove high-density impulsive noise, so they may be candidates for maneuver detection.

2.3.2. Application to Maneuver Detection

Let $x_k = (\Delta v_k)^2$, where $(\Delta v_k)^2$ is defined in 11b. Since x_k is χ^2 distributed with $d = 3$ degrees of freedom (in the absence of maneuvers), the median filter yields the sample median y_k of the χ^2 distribution, which is related to the variance of x_k. The median filter suppresses the tail of the distribution of x_k, with greater suppression for larger windows. The presence of maneuvers creates a tail on the distribution of x_k, but the tail is essentially "clipped off" by the median filter. Theoretical results on the statistics of the sample median are found in the literature within the subject of order statistics. A numerical analysis of the median filter shows that the variance of y_k is essentially the same as that of x_k prior to the addition of outliers, and so y_k is practically insensitive to outliers.

From (17), a rough estimate of the variance of the χ^2 data is

$$\mathrm{v}_k = y_k/[d(1 - 2/9d)^3] \tag{28}$$

The divisor is approximately 2.38 for $n = 3$ and its reciprocal is approximately 0.42. The estimate v_k is averaged by a first-order filter

$$\hat{\sigma}_k^2 = \hat{\sigma}_k^2 + g(\mathrm{v}_k - \hat{\sigma}_k^2) \tag{29}$$

where g is a gain parameter. A maneuver is declared when

$$x_{k-\ell} > \kappa\hat{\sigma}_k^2 \tag{30}$$

where ℓ is a lag such that $0 \leq \ell \leq N - 1$ lag and $\kappa > 0$ sets the detection sensitivity. The lag places the test value $x_{k-\ell}$ within the window of data from which the median is computed. To make the filter causal we set $\ell = 0$ so that the test is based on only past data. Frequent outliers tend to skew the median, so we set $x_{k-\ell} = y_k$ when a maneuver is detected.

The median operation in (22) involves finding the element of a list of N numbers (for odd N) for which there are as many elements greater than the median as there are smaller elements. A brute-force implementation requires $N(N - 1)/2$ comparisons. Wirth's "k-smallest" algorithm [18] is more efficient. Various algorithms and architectures for computing the running median are discussed in [11, §V].

Experiments with several sets of TLE data indicate that $N = 3$ to 9, $g = 0.002$ to 0.005, and $\kappa = 25$ to 35 are suitable parameters for most space objects. Theoretically a wider window increases sensitivity to maneuvers, as shown by the statistical analysis in [11, p. 1896]. If the number of degrees of freedom of are known and the variances in each axis are equal, κ can be chosen based on probabilities from the χ^2 distribution functions. One advantage to the median filter and the variance estimator is that they are scale-invariant. That is, the parameters do not have to be changed if the scaling or units of the data changes.

2.4. A Generalization

An assumption regarding the errors in $\Delta \boldsymbol{v}_k$ is that they have equal variance in each coordinate direction. Consider $\Delta \boldsymbol{v}_k$ in Range-Intrack-Crosstrack (RIC) coordinates defined by $\boldsymbol{r}_{k,k}$ and $\boldsymbol{v}_{k,k}$. Assume that the covariance matrix in RIC coordinates has the form $C = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$, where σ_x^2 is the radial error variance, σ_y^2 is the in-track error variance, and σ_z^2 is the cross-track error variance.[8] Then elements of the vector $C^{-1/2}(\Delta \boldsymbol{v}_k)$ are independent standard normal (zero mean, unit variance, Gaussian) random variables. Thus $\chi_C^2 = (\Delta \boldsymbol{v}_k)^T C^{-1} (\Delta \boldsymbol{v}_k)$ is chi-square distributed with mean $\mu = d$ and median $m = d(1 - 2/9d)^3$. Because the three variances are not known, we would have to estimate them or compute one variance and assume some ratios for the other two variances. We could instead apply the median filter to each (squared) component of $\Delta \boldsymbol{v}_k$ in RIC coordinates, which would help to distinguish various types of maneuvers. Three medians would be computed, from which estimates for σ_x^2, σ_y^2, and σ_z^2 would be obtained.

2.5. Verification and Characterization

The median filter is validated by generating $K + 1$ zero-mean white Gaussian random velocity vectors $\boldsymbol{v}_k$ of dimension $d \times 1$ and standard deviation $\sigma = 0.1$ in each component. Velocity differences are then computed,

$$\Delta \boldsymbol{v}_k = \boldsymbol{v}_k - \boldsymbol{v}_{k-1} \tag{31}$$

and then we compute $(\Delta v_k)^2 = |\Delta \boldsymbol{v}_k|^2$. The $(\Delta v_k)^2$ data are χ^2 distributed random variables with variance parameter $2\sigma^2$ with one-step correlation. A subset of the $\Delta \boldsymbol{v}_k$ vectors are replaced with $M \ll K$ random vectors $e_k \boldsymbol{u}_k$ at unique random indices uniformly spaced over $\mathcal{I} = \{1, 2, \ldots, K\}$. The random maneuver amplitudes e_k are uniformly distributed on the interval $(0, A)$.

If the velocity vectors were replaced by maneuver vectors, the $\Delta \boldsymbol{v}_k$ would contain doublets, which would give twice as many maneuver detections. An alternative approach would be to *add* the maneuvers to the $\Delta \boldsymbol{v}_k$ vectors, but replacement makes it easier to compute the probability that the maneuver exceeds a threshold. Note that the probability distribution of the input data $(\Delta v_k)^2$ is not χ^2, but is a mixed distribution of χ^2 data contaminated with uniformly distributed maneuver data.

The $|\Delta \boldsymbol{v}_k|^2$ data are passed through the median filter with window widths $N = 3, 5, 9, 15$. The median filter output is scaled using (18) to obtain a sample variance. Cases are run with $\sigma = 0.1$, $K = 100\,000$ samples, and $M = 0.2K$, so 20% of samples are maneuvers. The square root of the sample variance computed in (28) is the sample standard deviation $s = \sqrt{v}$, which has a mean value of approximately $\sqrt{2}\sigma$. For $\sigma = 0.1$ we have $s = 0.141$. A threshold κ depending on d is chosen for a specified probability of false alarm. For example, the false alarm rate P_{FA} is 1% for $d = 3$ and $\kappa = 11.34$. Since the component variances of the $\Delta \boldsymbol{v}_k$ data are $2\sigma^2$, a maneuver is detected when the maneuver amplitude exceeds $\sqrt{2\kappa\sigma^2} = 4.76\sigma$. Choosing a maximum maneuver amplitude $A = 20\sigma$, and recalling that the maneuver amplitudes are uniformly distributed, we expect that the probability of missed detection P_{MD} is $(4.76\sigma)/(20\sigma) \times 100 = 24.7\%$, that is, 24.7% of the

[8] Typically the cross-track error is smaller than the radial error, and the radial error is smaller than the in-track error (in general), so $\sigma_y^2 \leq \sigma_z^2 \leq \sigma_x^2$.

Table 1: Verification Test Results

N	**Detections (%)**	P_{FA} **(%)**	P_{MD} **(%)**	s
		0% Impulse Rate		
3	0.4	0.432	n/a	0.153
5	0.6	0.621	n/a	0.15
9	0.8	0.764	n/a	0.145
15	0.9	0.889	n/a	0.144
		5% Impulse Rate		
3	3.8	0.237	28.0	0.162
5	4.1	0.411	25.7	0.153
9	4.3	0.593	25.3	0.149
15	4.4	0.689	24.9	0.147
		20% Impulse Rate		
3	12	0.001	39.8	0.24
5	14	0.021	32.0	0.19
9	14	0.134	28.2	0.165
15	15	0.194	27.2	0.162
		50% Impulse Rate		
3	14	0.002	72.4	0.43
5	18	0.0	63.2	0.37
9	24	0.0	52.7	0.32
15	27	0.0	45.9	0.27

simulated maneuvers will not exceed the detection threshold. In practice, the threshold is computed from the average sample standard deviation $\bar{s}$, so the expected P_{MD} is

$$(4.76\bar{s})/(20\sigma) \times 100 \quad \text{(percent)} \tag{32}$$

Verification test results are shown in Table 1. Although there is some variabilty from run-to-run, these results are typical. It can be seen that the average sample standard deviation increases as the window width decreases. This is due to two effects: (1) the median of the input data is biased (increased) due to the larger tail on the probability distribution with maneuvers, compared to the χ^2 distribution in the absence of maneuvers, and (2) the median of the output of the median filter is biased (increased) for shorter windows. The median filter "clips" the tail of the distribution, effectively removing most of the maneuver information, which mitigates the first effect. The second effect causes the computed value of the average sample standard deviation to increase, thus increasing the average maneuver detection threshold. As is evident in Table 1, the larger detection threshold reduces the number of detections (shown as percentage of the total number of data samples), and it decreases P_{FA} and increases the P_{MD} according to (32). However, the performance of the median filter is not too sensitive to large maneuvers. In all cases, the P_{FA} is less than 1%, which is in accordance with $\kappa = 11.34$. The P_{MD} depends on the probability distribution of the amplitude of the maneuver, increasing for smaller maneuvers and decreasing for larger maneuvers. The performance of the median filter degrades when 50% of the data are maneuvers, but the filter does not "fall apart". The performance results are somewhat improved when the maneuver indices are equally spaced, since random spacing can cause sequential maneuvers whose duration is not less than the width of the window, thus temporarily increasing the detection threshold.

Further testing shows that the average sample standard deviation decreases to about 0.180 for $N = 3$ when the maneuver density is reduced to 10%. In the absence of maneuvers, with purely χ^2 input data, the false alarm rate is less than 1%, and the average sample standard deviation is close to the theoretical value of 0.141 for $N > 9$ and becomes biased for smaller N, growing to about 0.153 for $N = 3$.

3. RESULTS

Vallado [3] provides a set of Matlab functions to convert two-line element strings to orbital elements and to position and velocity vectors. The relevant top-level functions are `twoline2rv.m`, `sgp4init.m`, and `sgp4.m`. These functions are sufficient to demonstrate the maneuver detection algorithms, though improved perturbation force models are desirable and would yield better performance.

The maneuver detection algorithms were evaluated with the fixed set of parameters in Table 2 for all space objects. The same parameters were used to evaluate the algorithms using TLE data from several other space objects with good results (not reported here).

TLE data were obtained from Space Track [1]. Maneuver detection results for three space objects are listed in Table 3. In Table 3, the apogee and perigee are the apogee height and perigee height of the orbits relative to the mean equatorial radius of 6378.137 km. The orbital period, perigee height, apogee height, eccentricity, and inclination in the table are computed from the TLE data at the end of the time span.

Table 2: Maneuver Detection Parameters

Parameter	Symbol	Value	Units
Fading Memory Filter			
Polynomial order	n	3	
Measurement noise	σ_m	1	km
Process noise	σ_a	0.0001	m/s^2
Residual test threshold	κ	3.0	
Histogram Method			
Number of bins	B	200	
Maximum bin value	ΔV_{max}	4	m/s
Degrees of Freedom (DOF)	d	3	
Detection probability	P_τ	0.97	
Detection threshold multiplier*	κ	80.0	
Median Filter			
Window length	N	5	
Variance filter gain	g	0.005	
Variance multiplier	κ	22.68	
Minimum ΔV detection threshold	ΔV_{min}	2.0	m/s

The Histogram method was found to work better if the detection threshold is computed from P_τ in (21), rather than (20), so κ is not used.

Maneuvers often span more than one TLE and so multiple detections can occur. A maneuver reflected in one TLE generates two detections by the HIS and MED algorithms, since the velocity is differenced with the preceding and the following TLE. The FMF algorithm generates fewer detections because the filter covariance is opened up when a maneuver is detected.

The maneuver detection results are plotted in Figures 1 through 3. The maneuver detections are indicated by vertical lines on the graphs of the semimajor axis. The χ values and detection threshold are plotted for the FMF algorithm and the threshold for the HIS algorithm is indicated by a red vertical line on the histogram of

Table 3: Maneuver Detection Results

#	SatID	Name	Date Range		Maneuver Detections			Orbit	Period	Perigee	Apogee	Ecc	Incl
			Start	End	FMF	HIS	MED	Type	min	km	km		deg
3	22076	TOPEX	1993-01-03	1996-12-30	10	53	73	MEO	112	1332	1344	0.0007	66.0
10	23560	ERS 2	2007-01-02	2010-12-31	46	147	293	SS	100	784	785	0.0001	98.5
11	28884	GALAXY 15	2010-01-06	2010-12-31	21	12	61	GEO	1435	35766	35779	0.0002	0.7
13	29499	METOP-A	2011-01-23	2011-08-09	4	10	34	SS	101	820	821	0.0001	98.7
16	33331	GEOEYE 1	2008-09-07	2010-08-06	26	70	88	SS	98	671	686	0.0010	98.1
17	25544	ISS (ZARYA)	2005-01-02	2005-12-30	38	69	227	LEO	92	342	356	0.0011	51.6
21	25273	IRIDIUM 57	2010-01-02	2010-03-31	2	6	9	LEO	100	776	779	0.0002	86.4

the ΔV data. For the MED algorithm, the maneuver detection threshold computed from the running median is plotted in red along with the Δv data. A square plot symbol is shown at the top edge of the plots to indicate the existence of χ values and Δv values that are off-scale.

Results for TOPEX are shown in Figure 1. TOPEX is an Earth-observation satellite in a MEO orbit. TOPEX executed 9 known fine-control maneuvers from 1993 up to 1996 [19]. The FMF detected all of the maneuvers and a few false maneuvers. The HIS method and MED filter reported multiple detections for the same maneuvers. An examination of Figure 1C. b shows increased ΔV at the times of the maneuver detections by the MED filter. Maneuver detection results for TOPEX using a polynomial filter are reported in [19]. The actual maneuver times are shown in [19, Fig. 2a].

Results for ERS 2 are shown in Figure 2. ERS 2 an Earth resources spacecraft in a sun synchronous LEO orbit. Maneuvers from 2007 through 2010 comprise altitude and inclination adjustments. As can be seen in Figures 2, the FMF filter detected every maneuver, the HIS detected all but one maneuver, and the MED filter detected all maneuvers. An examination of the ΔV data in Figure 2 b shows that these all appear to be correct detections. These results compare favorably with those in [21, p. 12].

Results for Iridium 57 are shown in Figure 2 . Iridium 57 is a communications satellite in a high-inclination LEO orbit as part of a constellation. The FMF filter just missed detection of one of the three maneuvers by a small margin. The HIS method and the MED filter detected all three maneuvers. The three maneuvers are seen in the residual data in [25].

4. CONCLUSION

An adaptive fading memory filter, a histogram method, and a median filter were developed to detect maneuvers of space objects. These algorithms are adaptive to the noise level in the TLE data, which is not known and is non-stationary. These algorithms generally performed well with a single set of parameters and TLE data from several space objects in various orbital regimes. The fading memory and median filters do not require that a long history of data be processed prior to being started, as do other TLE-based algorithms.

The fading memory filter was tested using only semimajor axis data. Separate fading memory filters are needed to processes semimajor axis and inclination data, and any other orbital parameters, and each filter requires its own set of tuning parameters.

The histogram method and the median filter process the magnitude-square of the change in velocity computed from TLE data predicted to the same epoch, which essentially incorporates all orbital parameters. The histogram method works best with the detection threshold computed from a fixed probability of false alarm, rather than a threshold computed from a variance estimated from the median.

The median filter is a statistically robust estimator, which makes it suitable for maneuver detection. The median filter is simple and computationally fast, except for the need to compute velocity vectors from the TLE data.

It was found that the ΔV data are noisy, often well above the maneuver levels. Nevertheless, the histogram and median filter maneuver detection algorithms were able to detect most maneuvers. Typically there are multiple detections for the same maneuver. It was also found that the detection threshold κ for the median filter had to be 2 to 3 times larger than expected. It may be that the error in the ΔV data is not Gaussian, though it is zero-mean except for maneuvers.

Variations on the median filter should be considered. The median filter could be used in place of the threshold detection in the fading memory filter. The components of the ΔV vector could be processed individually, without squaring. In RIC coordinates it would detect in-plane and out-of-plane maneuvers. A similar but more efficient algorithm is to input the square of the 1-norm of the ΔV vector, rather than the square of the 2-norm, into the median filter [10, p. 29, Def. 3.1]. The vector median filter [15] may offer some advantages. Other algorithms include the weighted median filter and the adaptive weighted median filter (L-filter) [11, p. 1906], an adaptive order statistics filter [11, p. 1912], and a sliding window adaptive filter similar to a least-mean square adaptive filter [11, 16]. The recursive median filter (a median filter with feedback) [10, p. 8, §1.2] is a simple extension of the median filter introduced in this report and may be more robust.

The histogram method can be implemented much more efficiently by computing only the necessary quantiles, rather than numerous bins that are not used [5]. A sliding-window fading-memory histogram can also be implemented. Techniques such as those in [17] may improve efficiency.

4. REFERENCES

1. Space Track website: https://www.space-track.org/ (pages 1, 13)
2. CelesTrak website: https://celestrak.com/ (page 1)
3. D. Vallado, W. McClain, *Fundamentals of Astrodynamics and Applications*, Microcosm Press, 2nd Ed., 2004. Errata and astrodynamics software are available at https://celestrak.com/software/vallado-sw.asp (pages 6, 13)
4. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Mathematics in Science and Engineering, Vol. 64, Academic Press, New York, 1970. (page 3)

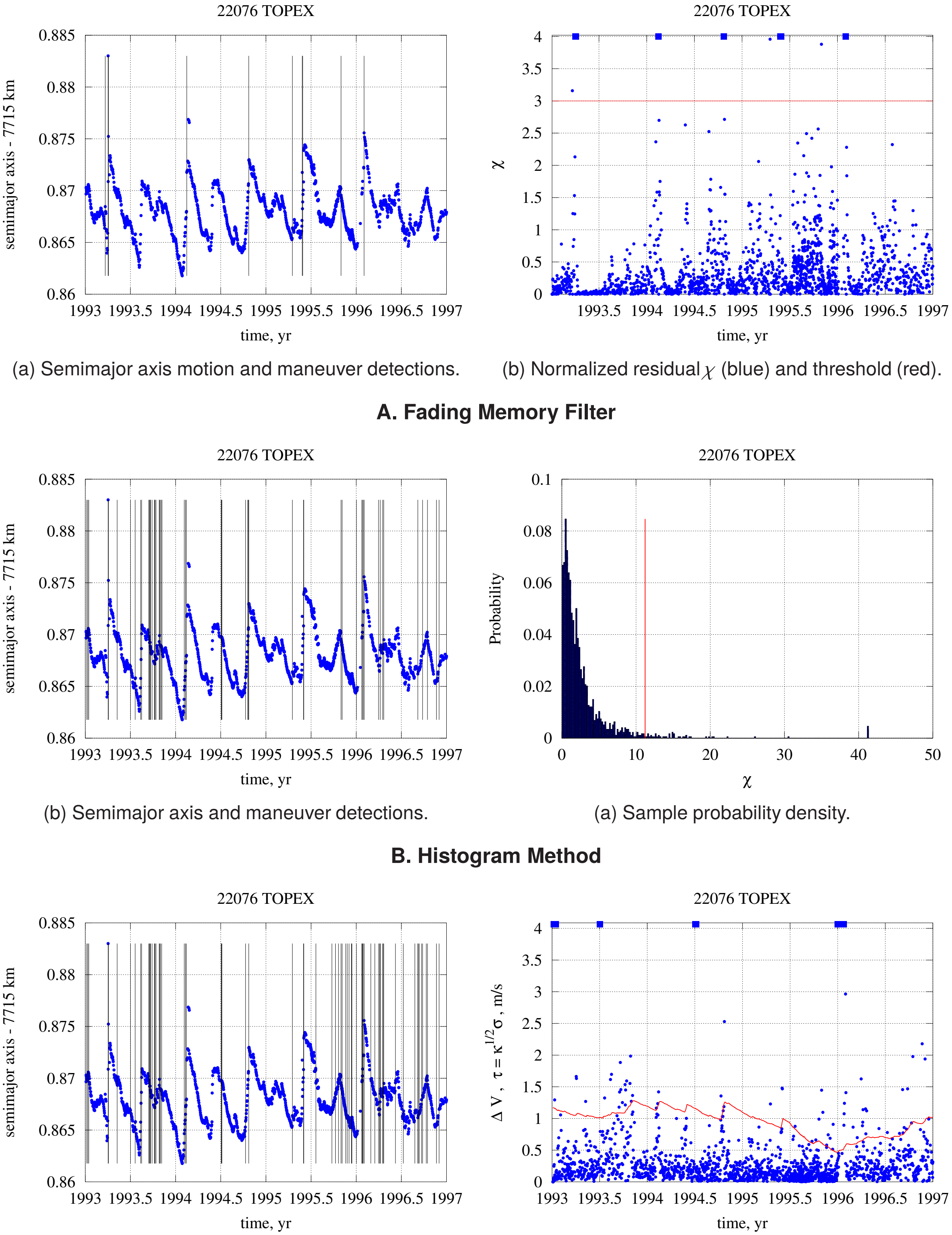
Median Filter

5. R. Jain, I. Chamtac, "The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations", Communications of the ACM: Simulation, Modeling, and Statistical Computing, Vol 28, No 10, Oct 1985, pp 1076–1085. (pages 8, 15)
6. D. R. Brillinger, "John W. Tukey's Work on Time Series and Spectrum Analysis", The Annals of Statistics, Vol 30, No 6, 2002, pp. 1595–1618. https://www.stat.berkeley.edu/~brill/Papers/time.pdf (page 9)
7. J. W. Tukey, "Nonlinear (Nonsuperposable) Methods for Smoothing Data", Proceedings of Congress Record EASCON, Washington DC, 7–9 October 1974, Vol 74, p. 673. (abstract only)
   Also in "The Collected Works of John W. Tukey, Volume II", (CWJWT II), 1985, pp. 837–855. (page 9)
8. J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley: Reading, MA, 1977. (page 9)
9. Barner, K.E., Yeong-Taeg Kim, Arce, G.R., *Order-Statistic Filtering and Smoothing of Time Series: Part 2*, Order Statistics and their Applications, (C.R. Rao and N. Balakrishnan, Eds.), Vol 16 of Handbook of Statistics, pp. 555–602, Amsterdam, The Netherlands: Elsevier Science, 1998. (pages 9, 10)
10. M. Gabbouj, E. J. Coyle, N. C. Gallagher, Jr., "An Overview of Median and Stack Filtering," Circuits, Systems, and Signal Process, Vol 11, Issue 1, March 1992, pp 7–45. DOI: http://dx.doi.org/10.1007/BF01189220 (pages 9, 15)
11. I. Pitas, A. N. Venetsanopoulos, "Order Statistics in Digital Image Processing", Proceedings of the IEEE, Vol 80, No 12, Dec 1992, pp 1893–1921. (pages 8, 10, 11, 15)

12. M. Gabbouj, “Weighted Median Filtering – Striking Analogies to FIR Filters”, Ch 1.11 in *Circuits and Systems Tutorials*, C. Toumazou, N. Battersby; S. Porta (Eds.), Wiley-IEEE Press eBook Chapters, Mar 1996, pp 5–21. DOI: http://dx.doi.org/10.1109/9780470544235.ch2 (pages 9, 10)
13. P. Maragos, R. W. Schafer, “Morphological Filters – Part II: Their Relations to Median, Order-Statistic, and Stack Filters”, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 8, 1987, pp. 1170–1184. (page 10)
14. P. Civicioglu, M. Alci, “Impulsive Noise Suppression from Images with the Noise Exclusive Filter”, EURASIP Journal on Applied Signal Processing, Hindawi Publishing Corporation, Vol 16, 2004, pp 2434–2440. (page 10)
15. J. Astola, P. Haavisto, Y. Neuvo, “Vector Median Filters”, Proceedings of the IEEE, Vol. 78, Issue 4, April 1990, pp. 678–689. DOI: http://dx.doi.org/10.1109/5.54807 (page 15)
16. M. Belge, E.L. Miller, “A sliding window RLS-like adaptive algorithm for filtering alpha-stable noise”, IEEE Signal Processing Letters, Vol 7, Issue 4, April 2000, pp. 86–89. DOI: http://dx.doi.org/10.1109/97.833005 (page 15)
17. B. Weiss, “Fast Median and Bilateral Filtering”, ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH 2006, Vol 25, No 3, pp. 519–526, July 2006. http://doi.acm.org/10.1145/1141911.1141918 (page 15)
18. Niklaus Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood Cliffs NJ, 1976, Prentice-Hall Series in Automatic Computation. (page 10)

Maneuver Detection – TLE Algorithms

19. T. Kelecy, D. Hall, K. Hamada, D. Stocker, “Satellite Maneuver Detection Using Two-line Element (TLE) Data”, Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference, Maui, Hawaii, 12–15 Sep 2007. http://www.amostech.com/TechnicalPapers/2007/Modeling_Analysis_Simulation/Kelecy.pdf (pages 2, 3, 5, 14)
20. J. F. McNeill, J. M. Coggi, R. L. Swartz, W. H. Ailor, T. O. Cooper, “Space Situation Monitoring Laboratory: An Integrated Web-Based Environment for Space Environment Information and Analysis”, Paper No AAS-09-416, AIAA/AAS Astrodynamics Specialist Conference, Pittsburg, PA, *Advances in the Astronautical Sciences: Astrodynamics 2009*, Vol 135, 2010. http://www.univelt.com/book=1343 (page 2)
21. J. Coggi, J. McNeill Jr., R. Nakagawa, “The Space Situation Monitoring Laboratory – What’s Happening in Space?”, The Aerospace Corporation, Presented at the International Symposium on Sustainable Space Development and Space Situational Awareness 2015 in Tokyo, Japan, 27 February 2015. http://www.jsforum.or.jp/debrisympo/2015/pdf/30SSML4JSF_Nakagawa.pdf (page 14)
22. R. P. Patera, “Space Event Detection Method”, AIAA-2006-6513, AIAA/AAS Astrodynamics Specialist Conference, 21–24 August 2006, Keystone, Colorado. DOI: 10.2514/6.2006-6513 http://arc.aiaa.org/doi/abs/10.2514/6.2006-6513 (pages 3, 5)
23. R. P. Patera, “Space Event Detection Method”, AIAA Journal of Spacecraft and Rockets, Vol 45, No 3, May–June 2008, pp. 554–559. DOI: 10.2514/1.30348 http://arc.aiaa.org/doi/abs/10.2514/1.30348?journalCode=jsr
24. L. A. Singh, *Assessment and Validation of the Space Incident Flagging Technique*, Report No. TOR-2008(8581)-8342, The Aerospace Corporation, El Segundo, CA, 2008.
25. R. L. Swartz, J. Coggi, J. McNeill, “A Swift SIFT for Satellite Event Detection”, Paper No AIAA-2010-7527, AIAA/AAS Astrodynamics Specialist Conference, Toronto, Ontario Canada, August 2010. http://arc.aiaa.org/doi/abs/10.2514/6.2010-7527 (pages 2, 3, 14)
26. S. Lemmens, H. Krag, “Two-Line-Elements-Based Maneuver Detection Methods for Satellites in Low Earth Orbit”, AIAA Journal of Guidance and Control, Control, and Dynamics, Vol 37, No 3, 2014, pp. 860–868. http://arc.aiaa.org/doi/abs/10.2514/1.61300 (pages 2, 3, 5)
27. P. Ghosh, “Space Event Detection via Robust Time Series Forecasting”, Paper No. AAS-16-303, Space Flight Mechanics Meeting, 14–18 February 2016, Napa, CA. (page 2)



(a) Semimajor axis motion and maneuver detections.

(b) Normalized residual χ (blue) and threshold (red).

A. Fading Memory Filter

(b) Semimajor axis and maneuver detections.

(a) Sample probability density.

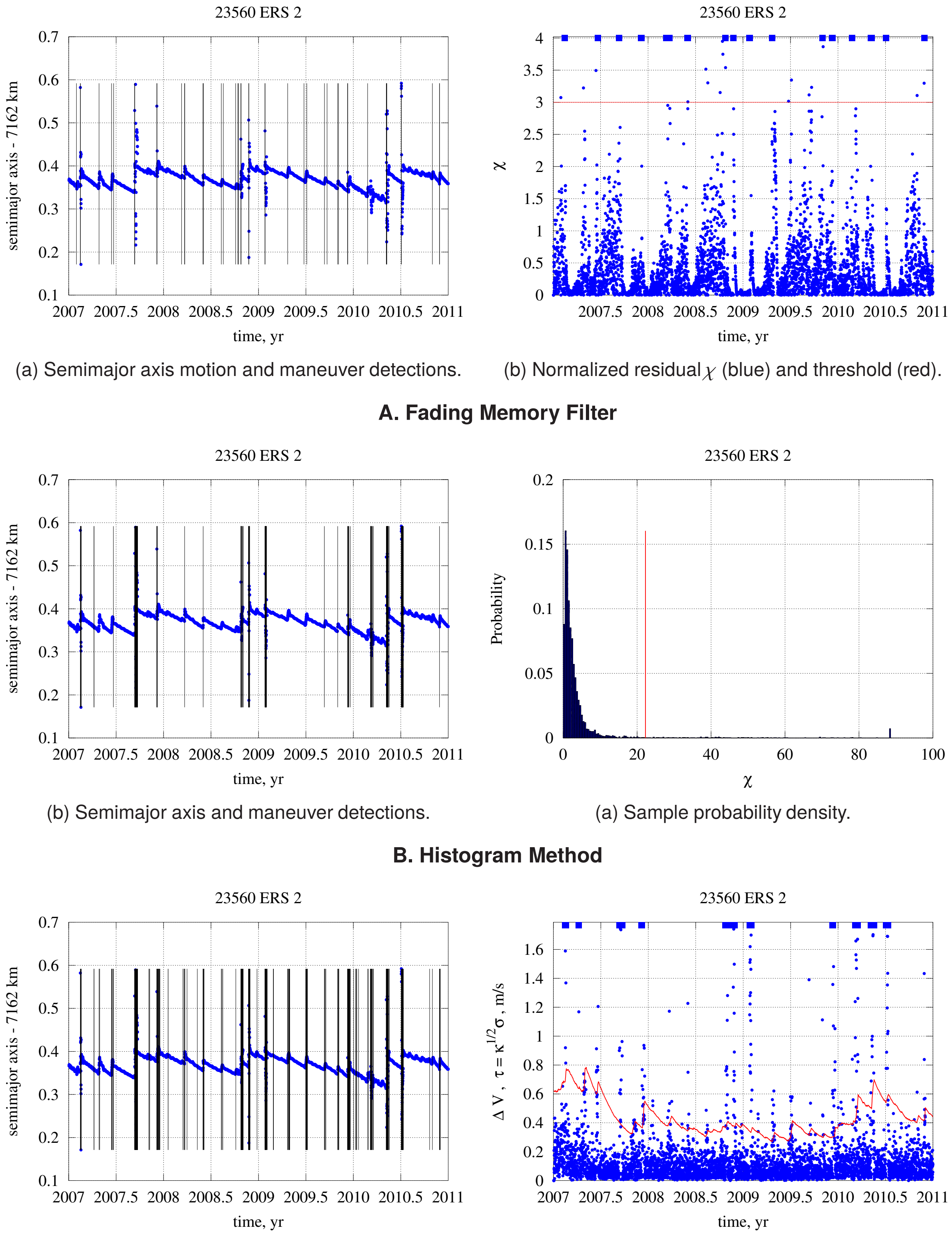
B. Histogram Method

(a) Semimajor axis and maneuver detections.

(b) $\Delta \boldsymbol{v}_k$ (blue) and detection threshold (red).

C. Median Filter

Fig. 1: Maneuver detection results for 22076-TOPEX



(a) Semimajor axis motion and maneuver detections.

(b) Normalized residual χ (blue) and threshold (red).

A. Fading Memory Filter

(b) Semimajor axis and maneuver detections.

(a) Sample probability density.

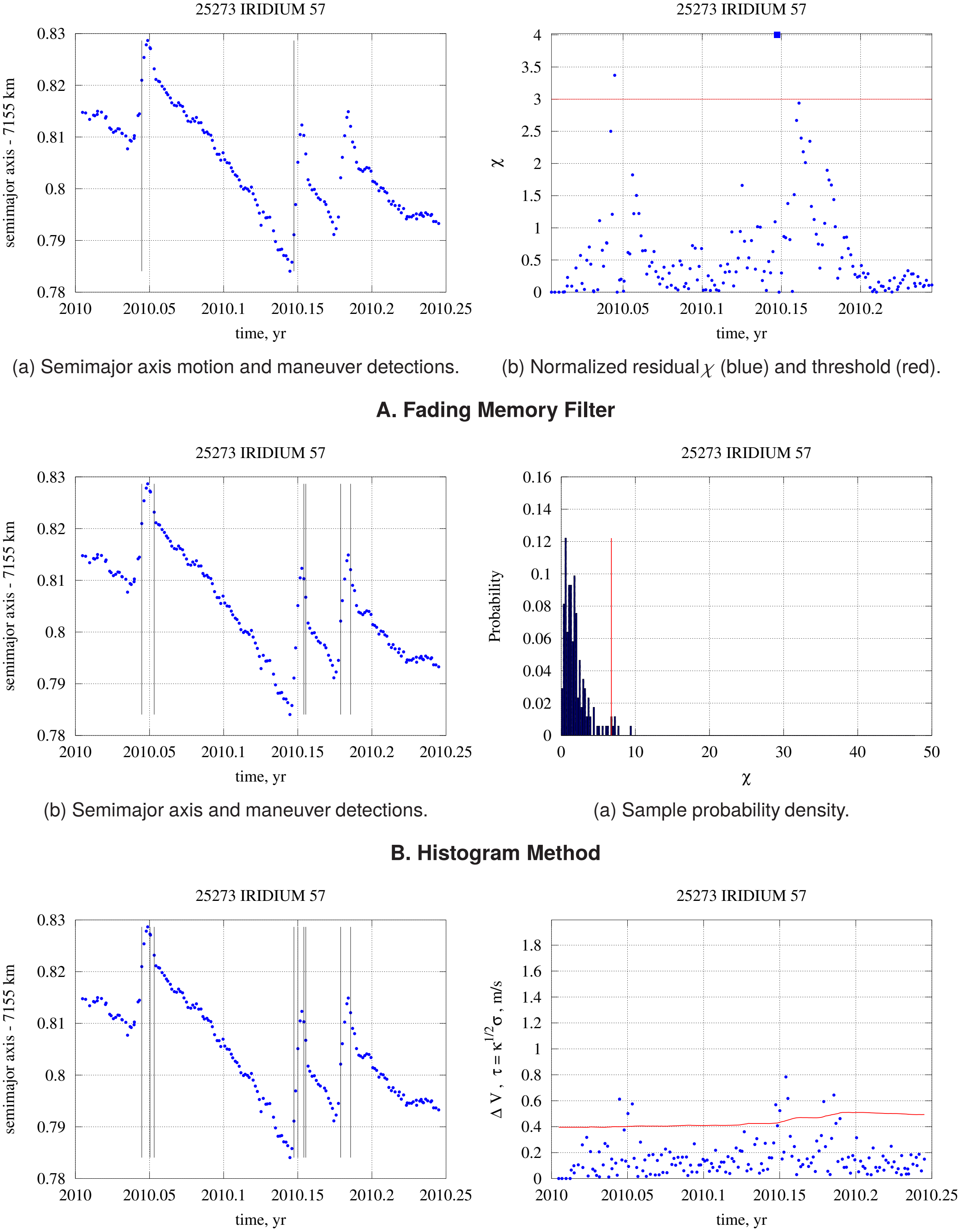
B. Histogram Method

(a) Semimajor axis and maneuver detections.

(b) $\Delta \boldsymbol{v}_k$ (blue) and detection threshold (red).

C. Median Filter

Fig. 2: Maneuver detection results for 23560-ERS-2



(a) Semimajor axis motion and maneuver detections.

(b) Normalized residual χ (blue) and threshold (red).

A. Fading Memory Filter

(b) Semimajor axis and maneuver detections.

(a) Sample probability density.

B. Histogram Method

(a) Semimajor axis and maneuver detections.

(b) $\Delta \boldsymbol{v}_k$ (blue) and detection threshold (red).

C. Median Filter

Fig. 3: Maneuver detection results for 25273-IRIDIUM-57