Robust GPS-INS Outlier Accommodation using a Sliding Window Filter: A Comparative Study

Paul F. Roysdon[†]   Jay A. Farrell[‡]

***Abstract*— Global Navigation Satellite systems (GNSS) and inertial navigation systems (INS) are important in many applications involving control and planning that require reliable state estimation with high bandwidth and accuracy. Typical environments for consumer applications, such as connected and autonomous vehicles, challenge these requirements because of the many factors that can cause anomalous GNSS measurements (i.e., outliers). A common approach is to use a single-epoch Extended Kalman Filter (EKF) for sensor fusion combined with the Receiver Autonomous Integrity Monitoring (RAIM) for GNSS outlier detection. However, the number of degrees-of-freedom (DOF) available per epoch for outlier accommodation is limited, which affects the ability to properly classify measurements as valid or not. False alarms result in good information not being incorporated into the state and covariance estimates. Missed detections result in incorrect information being incorporated into the state and covariance estimates. Either case can cause subsequent incorrect decisions, possibly causing divergence, due to the state and covariance now being incorrect. This article considers outlier accommodation within a sliding-window estimator containing multiple GNSS epochs. The sliding-window approach increase the dimension of the residual vector resulting in more DOF's, which enables enhanced outlier accommodation. The method solves the full-nonlinear *Maximum A Posteriori* problem in real-time. The article derives both the hypotheses testing and Least Soft-thresholded Squares approaches. Experimental sensor data is used to analyze each algorithms performance.**

I. Introduction

The past decades have seen the rapid rise and adoption of ubiquitous navigation systems. This has been driven by the availability of low cost sensors (e.g., GNSS and cameras), inertial instruments, and computation. With sensor fusion, these systems can exhibit very good accuracy (e.g. sub-meter error). However, further improvements in the reliability and continuity of this accuracy are required to fully support autonomous vehicle operations, especially in urban environments, where variations in the operating conditions can have critical effects. In GNSS applications such outlier measurements can be caused by multi-path, non-line of sight signals, or foliage. In the design of a reliable, high-performance system, it is critical to remove the effects of outlier measurements before they degrade performance.

RAIM is a set of techniques designed to detect and remove GNSS receiver outlier measurements [1], [2], [3], [4]. Successful accommodation requires measurement redundancy [1]. Many RAIM implementations assume that only a single outlier occurs in any epoch. Multiple outlier detection has also been well developed [2], [3], [5], The authors of [6] included an inertial measurement unit (IMU) and a Kalman filter to "extend" the RAIM capabilities. Their method is called eRAIM. Both RAIM and eRAIM are based on measurements from a single epoch, limiting data redundancy. Furthermore, the residual generation algorithm in RAIM and eRAIM assumes a linear system.

Data redundancy, quantified by the number of degrees-of-freedom (DOF), is critical to successful outlier accommodation. Redundancy can be enhanced both by adding additional sensors or by solving the estimation problem using all sensor data within a sliding temporal window. Herein we consider the sliding-window approach. We build on theoretical and computational methods developed within the control [7], robotics [8], simulataneous localization and mapping (SLAM) [9], [10], [11], and receding horizon estimation [12], [13], [14], [15] literatures. The resulting full nonlinear *Maximum A Posteriori* (MAP) estimator, without outlier accommodation, is presented in [16]. This article extends [16] with two methods to accommodate outlier measurements within the temporal window. The first method builds on traditional residual space methods which also form the basis for RAIM and eRAIM [17], [18], [19], [4]. Multiple outlier detection has also been well developed [20]. An earlier version of the residual-space sliding-window approach was presented in [21]. The second method is motivated by the Least Soft-thresholded Squares (LSS) approach, building on l_1-regularization, that was recently presented in the computer vision literature [22], [23], [24], [25], [26]. An earlier version of the LSS approach was presented in [27]. The contributions of the present paper relative to [21], [27] are more extensive presentation and theoretical analysis of each approach, in particular the first incorporation of the LSS approach in a real-time aided inertial vehicle state estimator. This paper also contains an extensive implementation section with detailed comparison of the two approaches.

The theoretical approach presented in Sections III–V is general. It applies to any set of aiding sensors (e.g., cameras, GNSS, RADAR, LIDAR) combined with kinematic integration for high-bandwidth based on data from an IMU or encoders. The focus on GNSS and INS herein should be thought of as a particular example, both to demonstrate the application of the theory and to allow experimental analysis of performance. Both theoretical and computational trade-off's are discussed. Both estimators are evaluated using both simulated data and real-world experimental data involving urban canyons and overhead foliage.

[†]Ph.D. graduate, [‡]Professor at the Dept. of Electrical & Computer Engineering, UC Riverside. {proysdon, farrell}@ece.ucr.edu.

II. Background and Notation

This section introduces Global Positioning System (GPS) aided inertial navigation system (INS) background [28].

A. Aided Inertial Navigation

Let $\boldsymbol{x} \in \mathbb{R}^{n_s}$ denote the rover state vector, where

$$\boldsymbol{x}(t) = [\mathbf{p}^\intercal(t), \mathbf{v}^\intercal(t), \mathbf{q}^\intercal(t), \mathbf{b}_a^\intercal(t), \mathbf{b}_g^\intercal(t)]^\intercal \in \mathbb{R}^{n_s},$$

where $\mathbf{p}$, $\mathbf{v}$, $\mathbf{b}_a$, $\mathbf{b}_g$ each in $\mathbb{R}^3$ represent the position, velocity, accelerometer bias and gyro bias vectors, respectively, and $\mathbf{q} \in \mathbb{R}^4$ represents the attitude quaternion ($n_s = 16$).

The kinematic equations for the rover state are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \mathbf{u}(t)), \tag{1}$$

where $\boldsymbol{f} : \mathbb{R}^{n_s} \times \mathbb{R}^6 \mapsto \mathbb{R}^{n_s}$ represents the kinematic model, and $\mathbf{u} \in \mathbb{R}^6$ is the vector of specific forces and angular rates. The function $\boldsymbol{f}$ is accurately known (see eqns. 11.31-11.33 in [28], derivations specific to this article are provided in Section 1 of [29]). The user applies forces and torques causing accelerations and angular rates which determine $\mathbf{u}(t)$. The kinematic integration of $\mathbf{u}(t)$ through eqn. (1) determines $\boldsymbol{x}(t)$.

Let τ_i denote the time instant of the i^{th} IMU measurement of $\mathbf{u}$. Each IMU measurement is modeled as

$$\tilde{\mathbf{u}}(\tau_i) = \mathbf{u}(\tau_i) - \mathbf{b}(\tau_i) - \boldsymbol{\omega}_u(\tau_i),$$

with white random errors $\boldsymbol{\omega}_u(\tau_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{Qd})$ and time-correlated random errors $\mathbf{b} = [\mathbf{b}_a^\intercal, \mathbf{b}_g^\intercal]^\intercal$.

Given the initial condition $\boldsymbol{x}(t_0) \sim \mathcal{N}(\boldsymbol{x}_0, \mathbf{P}_0)$ and measurements $\tilde{\mathbf{u}}$, an inertial navigation system propagates an estimate of the vehicle state $\hat{\boldsymbol{x}}(t)$ as the solution of

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \tilde{\mathbf{u}}(t)), \tag{2}$$

where $\hat{\boldsymbol{x}}(0) = \boldsymbol{x}_0$.

Let $\boldsymbol{x}_i$ and $\mathbf{u}_i$ denote $\boldsymbol{x}(\tau_i)$ and $\mathbf{u}(\tau_i)$. The solution of eqn. (2) over the interval $t \in [\tau_{i-1}, \tau_i]$ from the initial condition $\hat{\boldsymbol{x}}_{i-1}$ is $\hat{\boldsymbol{x}}_i \doteq \phi(\hat{\boldsymbol{x}}_{i-1}, \hat{\mathbf{u}}_{i-1})$, where $\hat{\mathbf{u}}_{i-1} \doteq \tilde{\mathbf{u}}_{i-1} - \hat{\mathbf{b}}_{i-1}$ and

$$\phi(\boldsymbol{x}_{i-1}, \mathbf{u}_{i-1}) = \boldsymbol{x}_{i-1} + \int_{\tau_{i-1}}^{\tau_i} \boldsymbol{f}(\boldsymbol{x}(\tau), \mathbf{u}(\tau))\, d\tau. \tag{3}$$

In this article, the equality symbol '=' will be used in its normal sense. The symbol '$\doteq$' will be used to indicate computations that are implemented in software.

Define $\mathbf{U}_{k-1} = \{\tilde{\mathbf{u}}(\tau_i) \text{ for } \tau_i \in [t_{k-1}, t_k]\}$. The integral operator in (3) can be iterated for all IMU measurements in $\mathbf{U}_{k-1}$ to propagate the state from t_{k-1} to t_k. Denote this iterative application of eqn. (3) as $\hat{\boldsymbol{x}}_k \doteq \boldsymbol{\Phi}(\hat{\boldsymbol{x}}_{k-1}, \mathbf{U}_{k-1})$.

B. GPS Model

Let $t_k = kT$ denote the time instants at which GPS measurements are valid, and $\boldsymbol{x}_k$ denote the state $\boldsymbol{x}(kT)$ at t_k. Typically there are numerous IMU measurements available between GPS epochs: $T \gg [\tau_i - \tau_{i-1}]$.

For $(m{+}1)$ satellites, $\mathbf{y}_k$ represents the double-differenced code (pseudorange) and Doppler measurement vector, as defined in Section 8.8 of [28]. For notational simplicity, it is assumed that the double difference approach removes all common-mode errors (e.g., ionosphere, troposphere, satellite clock and ephemeris errors), as well as the receiver clock biases. The double-differenced measurement model at t_k is

$$\mathbf{y}_k = \mathbf{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_{yk} + \mathbf{s}_k, \tag{4}$$

where $\boldsymbol{\eta}_{y,k} = [\boldsymbol{\eta}_{\rho,k}, \boldsymbol{\eta}_{d,k}]$, and $\mathbf{y}_k,\ \boldsymbol{\eta}_{y,k} \in \mathbb{R}^{2m}$. The symbol $\boldsymbol{\eta}_{\rho,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\rho,k})$ represents the pseudorange measurement noise, and $\boldsymbol{\eta}_{d,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{d,k})$ represents the Doppler measurement noise. Depending on receiver design, environmental factors and the performance of multipath mitigation techniques, the noise level $\mathbf{R}_\rho$ and $\mathbf{R}_d$ can vary for each available satellite. Using MATLAB syntax, the measurement noise covariance $\mathbf{R} = \text{blkdiag}(\mathbf{R}_\rho, \mathbf{R}_d) \in \mathbb{R}^{2m\times 2m}$.

The symbol $\mathbf{s}_k = [\mathbf{s}_{\rho,k}^\intercal, \mathbf{s}_{d,k}^\intercal]^\intercal \in \mathbb{R}^{2m}$ represents the error due to outliers, where $\mathbf{s}_{\rho,k} = [s_{\rho,1}, \dots, s_{\rho,m}]^\intercal$ and $\mathbf{s}_{d,k} = [s_{d,1}, \dots, s_{d,m}]^\intercal$. Throughout the paper, the ability to accomodate outliers will be considered from different perspectives. In Section III, the estimation approach is presented in the ideal case where $\mathbf{s}_k = \mathbf{0}$. In Section IV, the ability to detect and remove outliers is considered from the hypothesis testing point-of-view. Alternatively, in Section V, the outliers will be directly accommodated from the Least Soft-thresholded Squares (LSS) perspective.

Using the state estimate, the GPS measurements at t_k are predicted to be $\hat{\mathbf{y}}_k \doteq \mathbf{h}_k(\hat{\boldsymbol{x}}_k)$. The GPS measurement residual vector is computed as $\delta\mathbf{y}_k \doteq \mathbf{y}_k - \hat{\mathbf{y}}_k$.

C. INS Error Model

Due to initial condition errors, system calibration errors, and measurement noise, estimation error develops over time such that $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ are not equal. The error state vector is

$$\delta\boldsymbol{x} = [\delta\mathbf{p}^\intercal,\ \delta\mathbf{v}^\intercal,\ \delta\boldsymbol{\theta}^\intercal,\ \delta\mathbf{b}_a{}^\intercal,\ \delta\mathbf{b}_g{}^\intercal]^\intercal \in \mathbb{R}^{n_e},$$

where $\delta\mathbf{p}$, $\delta\mathbf{v}$, $\delta\boldsymbol{\theta}$, $\delta\mathbf{b}_a$, and $\delta\mathbf{b}_g$ each in $\mathbb{R}^3$ are the position, velocity, attitude, accelerometer bias and gyro bias error vectors, respectively. The fact that $n_s = 16$ and $n_e = 15$ is discussed in Section 3 of [29].

The error state $\delta\boldsymbol{x}(t)$ is related to $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ by $\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) \ominus \hat{\boldsymbol{x}}(t)$. The symbol '$\ominus$', which is discussed in detail in Section 2 of [29], represents the subtraction operation for position, velocity and bias states, and the multiplication operation of the attitude states. The dynamics and stochastic properties of this estimation error vector are well understood, and can be found in Section 11.4 of [28].

The linear state transition error model over $t \in [t_k, t_{k-1}]$ is defined as $\delta\boldsymbol{x}_k = \boldsymbol{\Upsilon}_{k-1}\delta\boldsymbol{x}_{k-1} + \mathbf{w}_{k-1}$ where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Qd})$ is white process noise. The derivation of this model from eqn. (3), along with the definitions of $\boldsymbol{\Upsilon}_{k-1}$ and $\mathbf{Qd}$, are presented in [29].

III. Estimation Theory

For a known linear system with white, normally distributed, and mutually uncorrelated process and measurement noise vectors with known covariance, the Kalman filter (KF) is the optimal estimator [30]. When the time propagation or measurement models are nonlinear, a variety of methods (e.g., the extended Kalman filter [31]) are available to solve the sensor fusion problem over a single GPS epoch.

This section reviews the MAP estimator [32] solved over a sliding temporal window in real-time. The problem has a long history [7]. While the theoretical motivations are distinct, the approach is also closely related to receding horizon estimation [15], [13], [12], [14]. This approach has been developed extensively in the Simultaneous Localization and Mapping (SLAM) research community [10], [11], [9], [33], [34], [35]. The approach developed for GPS-INS in [16] is referred to as a Contemplative Real Time (CRT) method due to the potential ability to evaluate and consider alternative outlier hypotheses for all data within the sliding window. That ability is demonstrated in [21] for CRT with Hypothesis Testing (CRT-HT), and for CRT with Least Soft-thresholding Squares (CRT-LSS) in [27]. The practical and theoretical comparison of CRT-HT versus CRT-LSS is developed and demonstrated herein.

A. Theoretical Solution

Let $\mathbf{X} = [\boldsymbol{x}(t_{k-L})^\intercal, \dots, \boldsymbol{x}(t_k)^\intercal]^\intercal \in \mathbb{R}^{n_s(L+1)}$ denote a state trajectory over a sliding time window that contains L GPS measurements: $[\mathbf{y}_{k-L+1}, \dots, \mathbf{y}_k]$. The window will slide one epoch upon arrival of each new GPS measurement. For presentation purposes only, we assume that each GPS epoch aligns with an IMU measurement time. The results in the experimental section relax this assumption.

Estimation of the vehicle trajectory $\mathbf{X}$ can be formulated as a MAP problem (see eqn. 11.18 in [32]):

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}} \left\{ p(\mathbf{X}, \mathbf{U}, \mathbf{Y}) \right\}, \tag{5}$$

where within the time window $\mathbf{U} = \{\mathbf{U}_l \,|\, l \in [k-L, k-1]\}$, and $\mathbf{Y} = \{\mathbf{y}_j \mid j \in [k-L+1, k]\}$ is the set of GPS measurements over the time window for satellites $1, ..., m$. Both $\mathbf{U}$ and $\mathbf{Y}$ will be treated as concatenated vectors in the analysis that follows. The joint probability for the GPS-INS problem, $p(\mathbf{X}, \mathbf{U}, \mathbf{Y})$, can be factored as

$$\begin{aligned} p(\mathbf{X}, \mathbf{U}, \mathbf{Y}) &= p(\mathbf{X}, \mathbf{U}) p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U}) \qquad (6) \\ &= p(\boldsymbol{x}_{k-L}) \prod_{l=k-L}^{k-1} p(\boldsymbol{x}_{l+1} | \boldsymbol{x}_l, \mathbf{U}_l) \prod_{j=k-L+1}^{k} p(\mathbf{y}_j | \boldsymbol{x}_j), \end{aligned}$$

where $p(\boldsymbol{x}_{k-L})$ is the distribution of the initial condition for the time window.

B. Numerical Solution

Assume that $\boldsymbol{x}(t_{k-L})$, $\mathbf{w}_{k-1}$, and $\boldsymbol{\eta}_y$ have Gaussian distributions with positive definite covariance matrices $\mathbf{P}_{(k-L)}$, $\mathbf{Q_D}$, and $\mathbf{R}$, respectively. Finding $\mathbf{X}$ that maximizes eqn. (6) is equivalent to minimizing the negative of its natural logarithm. This yields the equivalent nonlinear cost function:

$$\begin{aligned} \mathcal{C}(\mathbf{X}) &= \|\hat{\boldsymbol{x}}_{k-L} - \boldsymbol{x}(t_{k-L})\|^2_{\mathbf{P}_{(k-L)}} \\ &+ \sum_{l=k-L}^{k-1} \|\boldsymbol{\Phi}\big(\boldsymbol{x}(t_l), \mathbf{U}_l\big) - \boldsymbol{x}(t_{l+1})\|^2_{\mathbf{Q_D}} \\ &+ \sum_{j=k-L+1}^{k} \|\mathbf{y}(t_j) - \mathbf{h}_j\big(\boldsymbol{x}(t_j)\big)\|^2_{\mathbf{R}}, \end{aligned} \tag{7}$$

which can be compactly expressed as $\mathcal{C}(\mathbf{X}) = \|\mathbf{v}(\mathbf{X})\|^2_{\mathbf{W}}$, where $\mathbf{v}(\mathbf{X}) \in \mathbb{R}^{(n_s + n_s L + 2mL)\times 1}$ is defined as

$$\begin{aligned} &\mathbf{v}(\mathbf{X}) \\ &= \Big[\hat{\boldsymbol{x}}^\intercal_{k-L} - \boldsymbol{x}(t_{k-L})^\intercal, \boldsymbol{\Phi}(\mathbf{X}, \mathbf{U})^\intercal - \mathbf{X}^\intercal, \mathbf{Y}^\intercal - \mathbf{h}(\mathbf{X})^\intercal \Big]^\intercal. \end{aligned}$$

In this expression, $\boldsymbol{\Phi}(\mathbf{X}, \mathbf{U}) - \mathbf{X}$ represents the concatenation of the vector terms $\boldsymbol{\Phi}\big(\boldsymbol{x}(t_l), \mathbf{U}_l\big) - \boldsymbol{x}(t_{l+1})$, and $\mathbf{h}(\mathbf{X})$ represents the concatenation of the vector $\mathbf{h}_j(\boldsymbol{x}(t_j))$ terms. The matrix $\mathbf{W} \in \mathbb{R}^{(n_s+n_sL+2mL)\times(n_s+n_sL+2mL)}$ is the positive definite block diagonal matrix formed by the positive definite submatricies $\mathbf{P}_{(k-L)} \in \mathbb{R}^{n_s \times n_s}$, $\bar{\mathbf{Q}}_\mathbf{D} = \text{blkdiag}(\mathbf{Q_D}) \in \mathbb{R}^{n_s L \times n_s L}$, and $\bar{\mathbf{R}} = \text{blkdiag}(\mathbf{R}) \in \mathbb{R}^{2mL \times 2mL}$. The matrix $\mathbf{W}$ can be represented as $\mathbf{W} = \text{blkdiag}(\mathbf{P}_{(k-L)}, \bar{\mathbf{Q}}_\mathbf{D}, \bar{\mathbf{R}})$, and $\|\mathbf{v}\|^2_{\mathbf{W}} = \mathbf{v}^\intercal \mathbf{W}^{-1} \mathbf{v}$ represents the squared Mahalanobis norm with matrix $\mathbf{W}$.

The cost function in eqn. (7) can be normalized using Cholesky Decomposition [10], [11], [33]. Let $\boldsymbol{\Sigma}^\intercal_\mathbf{W} \boldsymbol{\Sigma}_\mathbf{W} = \mathbf{W}^{-1}$. Then, $\mathbf{r} \triangleq \boldsymbol{\Sigma}_\mathbf{W} \mathbf{v}$ is the weighted residual, and $\|\mathbf{v}\|_\mathbf{W} = \|\mathbf{r}\|_2$. With this notation the cost function of eqn. (7) reduces to

$$\mathcal{C}(\mathbf{X}) = \|\mathbf{a} - \mathbf{b}(\mathbf{X})\|^2_2, \tag{8}$$

where

$$\mathbf{a} \doteq \boldsymbol{\Sigma}_\mathbf{W} [\hat{\boldsymbol{x}}^\intercal_{k-L}, \mathbf{0}^\intercal, \mathbf{Y}^\intercal]^\intercal \tag{9}$$

represents the terms that are known at each iteration, and

$$\mathbf{b}(\mathbf{X}) = \boldsymbol{\Sigma}_\mathbf{W} [x(t_{k-L})^\intercal, \boldsymbol{\Phi}(\mathbf{X}, \mathbf{U})^\intercal - \mathbf{X}^\intercal, \mathbf{h}(\mathbf{X})^\intercal]^\intercal$$

represents the terms that are computed based on $\mathbf{X}$.

The optimal value of $\mathbf{X}$ in eqn. (5) is

$$\mathbf{X}^* = \underset{\mathbf{X}}{\mathit{arg\,min}} \left\{ \|\mathbf{a} - \mathbf{b}(\mathbf{X})\|^2_2 \right\}. \tag{10}$$

Given an initial trajectory vector

$$\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}(t_{k-L})^\intercal, \dots, \hat{\boldsymbol{x}}(t_k)^\intercal]^\intercal \in \mathbb{R}^{n_s(L+1)},$$

this optimization can be solved iteratively using the Taylor series expansion to approximate $\mathbf{b}(\mathbf{X})$:

$$\mathbf{b}(\mathbf{X}) = \mathbf{b}(\hat{\mathbf{X}}) + \mathbf{B}\delta\mathbf{X} + h.o.t's, \tag{11}$$

where the Jacobian $\mathbf{B} \doteq \frac{\partial \mathbf{b}(\mathbf{X})}{\partial \mathbf{X}}\Big|_{\mathbf{X}=\hat{\mathbf{X}}}$, and

$$\delta\mathbf{X} = [\delta\boldsymbol{x}(t_{k-L})^\intercal, \dots, \delta\boldsymbol{x}(t_k)^\intercal]^\intercal \in \mathbb{R}^{n_e(L+1)}.$$

Substituting eqn. (11) into eqn. (10) and ignoring the higher order terms ($h.o.t's$), yields

$$\begin{aligned}\delta\mathbf{X}^* &= \mathop{\mathit{arg\,min}}_{\delta\mathbf{X}} \|\mathbf{a} - (\mathbf{b}(\hat{\mathbf{X}}) + \mathbf{B}\delta\mathbf{X})\|_2^2 \\ &= \mathop{\mathit{arg\,min}}_{\delta\mathbf{X}} \|\mathbf{r} - \mathbf{B}\delta\mathbf{X}\|_2^2, \end{aligned} \tag{12}$$

where

$$\mathbf{r} \doteq \mathbf{a} - \mathbf{b}(\hat{\mathbf{X}}). \tag{13}$$

The optimal $\delta\mathbf{X}^*$ is found by solving

$$\mathbf{B}\delta\mathbf{X}^* = \mathbf{r} \tag{14}$$

in the least squares sense. The matrix $\mathbf{B} \in \mathbb{R}^{(n_e+n_eL+2mL)\times(n_e+n_eL+2mL)}$ is sparse; therefore, eqn. (14) can be solved efficiently by many methods [10], [11], [9].

The optimal $\delta\mathbf{X}$ is the solution of the normal equation:

$$\mathbf{B}^\intercal\mathbf{B}\delta\mathbf{X}^* \doteq \mathbf{B}^\intercal\mathbf{r}. \tag{15}$$

Eqn. (15) can be compactly expressed as

$$\boldsymbol{\Xi}\delta\mathbf{X}^* = \boldsymbol{\xi}, \tag{16}$$

where $\boldsymbol{\Xi} = \mathbf{B}^\intercal\mathbf{B}$ is the information matrix, $\boldsymbol{\xi} = \mathbf{B}^\intercal\mathbf{r}$ is the information vector [10], [11], [9].

Given $\delta\mathbf{X}^*$, the update to the trajectory after each iteration is

$$\hat{\mathbf{X}}^+ \doteq \hat{\mathbf{X}}^- \oplus \delta\mathbf{X}^*. \tag{17}$$

The optimal solution is obtained by iterating eqns. (9), (11), (13), (14), and (17) to convergence for some user defined stopping conditions. The superscript symbol '$-$' denotes the trajectory estimate at the start of the iteration, and superscript '$+$' is the updated trajectory estimate. The symbol '$\oplus$', which is discussed in Section 2 of [29], represents the addition operation for position, velocity and bias states, and the multiplication operation of the attitude states. A line search is implemented in the direction of $\delta\mathbf{X}^*$ from eqn. (14) to determine the magnitude of the update for eqn. (17) (see [11], [36]).

Solving the full MAP problem results in the optimal estimate of the vehicle trajectory $\mathbf{X}$. When the structure of $\boldsymbol{\Xi}$ is sparse, the computational load increases linearly with the length L of the trajectory (see [11]). The computational complexity of the algorithm is discussed in Section 4 of [29].

C. Optimization: Iterated Solution

Consider the l^{th} iteration of the optimization. Given an estimate of the solution $\hat{\mathbf{X}}^l$, the optimization algorithm computes an error vector $\delta\mathbf{X}^l$, which corrects $\hat{\mathbf{X}}^l$ using eqn. (17) to yield an improved solution $\hat{\mathbf{X}}^{l+1}$ to eqn. (7).

The process of re-integrating the trajectory $\mathbf{X}$ for each iteration, is discussed in Section 4.3 of [16]. The iterative process is terminated when either $\|\delta\mathbf{X}\|_2 \leq \epsilon$ or $l \geq l_{\max}$, where $\epsilon > 0$ and $l_{\max}$ are user-defined parameters.

The derivation of eqn. (12) shows that

$$\mathbf{r}(\mathbf{X}) = \mathbf{B}(\hat{\mathbf{X}})\delta\mathbf{X} + \boldsymbol{\eta}_\mathbf{r}, \tag{18}$$

where $\mathbf{B}(\hat{\mathbf{X}})$ is the Jacobian of $\mathbf{r}(\mathbf{X})$ evaluated at $\hat{\mathbf{X}}$, with $\boldsymbol{\eta}_\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This fact will be used in Section IV.

IV. Residual Space Outlier Detection

This section examines residual space methods for outlier detection and removal. The discussion in this section uses the standard notation in the hypothesis testing (HT) literature [37], [38], [39]. The mapping between the HT and CRT variables, comparing eqns. (18) and (19), is $\mathbf{y} = \mathbf{r}$, $\mathbf{H} = \mathbf{B}$, $\boldsymbol{x} = \delta\mathbf{X}$, and $\boldsymbol{\eta} = \boldsymbol{\eta}_\mathbf{r}$. By eqn. (18), $\boldsymbol{\eta}_\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_\eta^2\mathbf{I})$ with $\boldsymbol{\sigma}_\eta^2 = 1$. Complete derivations for this section are provided in Section 5 of [29]. The method developed in this section will be referred to herein as CRT-HT.

Before the optimization process discussed in Section III-C, the state vector is integrated from the prior (optimized) state $\hat{\mathbf{X}}^0(t_{k-L})$ to the current GPS epoch t_k, giving an initial trajectory $\hat{\mathbf{X}}^0$, then the GPS residual is computed. Note we assume the prior and IMU to be outlier free, and therefore only consider GPS outliers. Consider the following hypotheses related to the GPS residual $\mathbf{r}(\hat{\mathbf{X}}^0)$:

- Null Hypothesis, $\mathcal{H}_0$:

$$\mathbf{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{\eta}, \tag{19}$$

- Alternative Hypothesis, $\mathcal{H}_i$:

$$\mathbf{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{\eta} + \mu_i\mathbf{e}_i. \tag{20}$$

For $\mathcal{H}_i$, corresponding to (4), the outlier model is $\mathbf{s} = \mu_i\mathbf{e}_i$ and since the prior and IMU are assumed outlier free: $i = n_e(L+1)+1, \cdots, n_e(L+1)+2m$. When the magnitude of the outlier μ_i is nonzero, the i^{th} measurement is called an *outlier*. The magnitude μ_i will affect the ability to detect such outliers. The null-hypothesis assumes no outliers, i.e., $\mu_i = 0$.

In this section, $\mathbf{y} \in \mathbb{R}^{p\times 1}$ is the measurement vector, $\mathbf{H} \in \mathbb{R}^{p\times q}$, $p > q$, $\text{rank}(\mathbf{H}) = q$ is the measurement matrix, and $\boldsymbol{x} \in \mathbb{R}^{q\times 1}$ is the vector to be estimated. For $\mathcal{H}_i$, $\mathbf{e}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^\intercal \in \mathbb{R}^{p\times 1}$. To simplify notation in the following equations, let $\boldsymbol{\varepsilon}_i \triangleq \mu_i\mathbf{e}_i$.

A. Null-hypothesis, $\mathcal{H}_0$

Corresponding to eqn. (19), the minimum-variance unbiased estimator (MVUE) [32] for $\boldsymbol{x}$ is

$$\hat{\boldsymbol{x}} = (\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal\mathbf{y}. \tag{21}$$

Consider the residual $\mathbf{r}_0 \triangleq \mathbf{y} - \hat{\mathbf{y}}$, where $\hat{\mathbf{y}} = \mathbf{H}\hat{\boldsymbol{x}} = \mathbf{H}(\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal\mathbf{y}$. Then

$$\mathbf{r}_0 = (\mathbf{I} - \mathbf{P})\boldsymbol{\eta}, \tag{22}$$

where $\mathbf{P} \triangleq \mathbf{H}(\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal \in \mathbb{R}^{p\times p}$ is the projection matrix onto the range-space of $\mathbf{H}$. The matrix $\mathbf{P}$ is symmetric, idempotent, and $\text{rank}(\mathbf{P}) = q$. Similarly, the matrix $\mathbf{Q} \triangleq (\mathbf{I} - \mathbf{P}) \in \mathbb{R}^{p\times p}$ is a real, symmetric, and idempotent matrix. The matrix $\mathbf{Q}$ is a projection matrix onto the left null-space of $\mathbf{H}$. Importantly, $\text{rank}(\mathbf{Q}) = \text{trace}(\mathbf{Q}) = (p - q)$.

The mean and covariance of the residual are

$$\mathrm{E}\left\langle \mathbf{r}_0 \right\rangle = \mathrm{E}\left\langle \mathbf{Q}\boldsymbol{\eta} \right\rangle = \mathbf{0} \tag{23}$$

$$\mathrm{Cov}\left\langle \mathbf{r}_0 \right\rangle = \mathbf{Q}\mathrm{E}\left\langle \boldsymbol{\eta}\boldsymbol{\eta}^\intercal \right\rangle \mathbf{Q}^\intercal = \boldsymbol{\sigma}_\eta^2 \mathbf{Q}. \tag{24}$$

The final step is valid because $\mathbf{Q}$ is idempotent, and $\mathrm{E}\left\langle \boldsymbol{\eta}\boldsymbol{\eta}^\intercal \right\rangle = \boldsymbol{\sigma}_\eta^2 \mathbf{I}$. The mean square error (MSE) is

$$\begin{aligned} \mathrm{E}\left\langle \|\mathbf{r}_0\|^2 \right\rangle &= \mathrm{E}\left\langle \mathbf{r}_0^\intercal \mathbf{r}_0 \right\rangle \\ &= \mathrm{E}\left\langle \mathrm{trace}(\mathbf{Q}\boldsymbol{\eta}\boldsymbol{\eta}^\intercal\mathbf{Q}^\intercal) \right\rangle \\ &= (p-q)\boldsymbol{\sigma}_\eta^2. \end{aligned} \tag{25}$$

The standard test statistic $\boldsymbol{\Gamma}_{\hat{\mathbf{X}}}$ for the validity of $\mathcal{H}_0$ [37], [38], [39] is:

$$\boldsymbol{\Gamma}_{\hat{\mathbf{X}}} = \frac{\|\mathbf{r}_0(\hat{\mathbf{X}})\|^2}{\mathrm{E}\left\langle \|\mathbf{r}_0(\hat{\mathbf{X}})\|^2 \right\rangle} = \frac{\|\mathbf{r}_0(\hat{\mathbf{X}})\|^2}{(p-q)\boldsymbol{\sigma}_\eta^2}. \tag{26}$$

Under $\mathcal{H}_0$, the statistic $\boldsymbol{\Gamma}_{\hat{\mathbf{X}}}$ is a reduced chi-square [40], with expected value equal to 1, that can be used to test the validity of hypothesis $\mathcal{H}_0$ (i.e., detect the existence of outliers). The test statistic calculated by eqn. (26) is evaluated relative to a threshold computed using the one-tailed Chi-square distribution with significance level α, normalized by the number of degrees-of-freedom (DOF) [40], [41]:

$$\boldsymbol{\Gamma}_{\hat{\mathbf{X}}} < \frac{\boldsymbol{\chi}^2_{\alpha/2,(p-q)}}{(p-q)}. \tag{27}$$

The threshold $\boldsymbol{\chi}^2_{\alpha/2,(p-q)}$ is determined from a look-up table for α versus DOF. The significance level α is chosen by the designer for some probability of success. For example, $\alpha = 0.05$ indicates a 95% confidence level.

If the test succeeds, then $\mathbf{r}_0(\hat{\mathbf{X}})$ is assumed outlier-free and the optimization process of Section III-B is performed. Otherwise, outlier identification is performed.

B. Alternate-hypothesis, $\mathcal{H}_i$

To analyze the effect of the outlier $\boldsymbol{\varepsilon}_i$ on the state error, substitute eqn. (20) into eqn. (21)

$$\begin{aligned} \hat{\boldsymbol{x}} &= ((\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal)(\mathbf{H}\boldsymbol{x} + \boldsymbol{\eta} + \boldsymbol{\varepsilon}_i) \\ &= \mathbf{I}\boldsymbol{x} + (\mathbf{H}^\intercal\mathbf{H})^{-1}\mathbf{H}^\intercal(\boldsymbol{\eta} + \boldsymbol{\varepsilon}_i). \end{aligned} \tag{28}$$

Therefore, $\delta\boldsymbol{x} = \mathbf{H}^*(\boldsymbol{\eta}+\boldsymbol{\varepsilon}_i)$. The expected value of the state error due to the outlier is $\mathrm{E}\left\langle \delta\boldsymbol{x} \right\rangle = \mathbf{H}^*\boldsymbol{\varepsilon}_i$.

To analyze the effect of the outlier on the residual, substitute eqns. (20) and (28) into the definition of $\mathbf{r}$ yields:

$$\mathbf{r} = \mathbf{Q}(\boldsymbol{\eta} + \boldsymbol{\varepsilon}_i). \tag{29}$$

Note that the residual still lies in the left-null-space of $\mathbf{H}$. The mean and covariance of $\mathbf{r}$ due to the outlier, are

$$\mathrm{E}\left\langle \mathbf{r} \right\rangle = \mathrm{E}\left\langle \mathbf{Q}(\boldsymbol{\eta} + \boldsymbol{\varepsilon}_i) \right\rangle = \mathbf{Q}\boldsymbol{\varepsilon}_i \tag{30}$$

$$\mathrm{Cov}\left\langle \mathbf{r} \right\rangle = \mathrm{E}\left\langle (\mathbf{r} - \mathbf{Q}\boldsymbol{\varepsilon}_i)(\mathbf{r} - \mathbf{Q}\boldsymbol{\varepsilon}_i)^\intercal \right\rangle = \boldsymbol{\sigma}_y^2\mathbf{Q}. \tag{31}$$

Comparing eqn. (24) with eqn. (31), we see that the covariance is the same both with and without the outlier; however, the means are different, as shown in eqns. (23) and (30), which provides a method for identifying outliers. The decision statistic under the alternate-hypothesis $\mathcal{H}_i$, is based on the distribution of $\mathbf{r} \sim \mathcal{N}(\mathbf{Q}\boldsymbol{\varepsilon}_i, \boldsymbol{\sigma}_y^2\mathbf{Q})$.

Consider the parity vector [1], [4]

$$\mathbf{p} \triangleq \mathbf{U}_2^\intercal \mathbf{r} \in \mathbb{R}^{(p-q)},$$

where the columns of $\mathbf{U}_2 \in \mathbb{R}^{p\times(p-q)}$ form an orthonormal basis for the null-space of $\mathbf{H}^\intercal$ and $\mathbf{U}_2\mathbf{U}_2^\intercal = \mathbf{Q}$ (see Section 4 of [29]). Thus, using eqn. (29)

$$\begin{aligned} \mathbf{p} &= \mathbf{U}_2^\intercal(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{x}}) \\ &= (\mathbf{U}_2^\intercal\mathbf{e}_i)\mu_i + \mathbf{U}_2^\intercal\boldsymbol{\eta}; \end{aligned} \tag{32}$$

therefore, $\mathbf{p} \sim \mathcal{N}(\mu_i\mathbf{U}_2^\intercal\mathbf{e}_i,\ \boldsymbol{\sigma}_\eta^2\mathbf{I}_{p-q})$ (see [29]).

Then the magnitude of the outlier μ_i in eqn. (20) can be estimated as (see Section 5.6 of [40]),

$$\begin{aligned} \hat{\mu_i} &= \left((\mathbf{U}_2^\intercal\mathbf{e}_i)^\intercal(\boldsymbol{\sigma}_y^2\mathbf{I})^{-1}(\mathbf{U}_2^\intercal\mathbf{e}_i)\right)^{-1}(\mathbf{U}_2^\intercal\mathbf{e}_i)^\intercal(\boldsymbol{\sigma}_y^2\mathbf{I})^{-1}\mathbf{p} \\ &= \boldsymbol{\sigma}_y^2\left(\mathbf{e}_i^\intercal\mathbf{U}_2\mathbf{U}_2^\intercal\mathbf{e}_i\right)^{-1}\frac{1}{\boldsymbol{\sigma}_y^2}\mathbf{e}_i^\intercal\mathbf{U}_2\mathbf{p} \\ &= \frac{\mathbf{e}_i^\intercal\mathbf{Q}^\intercal\mathbf{r}}{\mathbf{e}_i^\intercal\mathbf{Q}^\intercal\mathbf{e}_i}, \end{aligned}$$

where the covariance of $\hat{\mu_i}$ is $\mathrm{Cov}\left\langle \hat{\mu_i} \right\rangle = (\mathbf{e}_i^\intercal\mathbf{Q}^\intercal\mathbf{e}_i)^{-1}$.

Each time that $\mathcal{H}_0$ is rejected, outlier identification is executed iteratively for each i in the alternative hypothesis set, assuming a single outlier. Each μ_i is compared against a threshold $\boldsymbol{\gamma}$, such that any $\mu_i > \boldsymbol{\gamma}$ is considered an outlier. After completion of the identification process, if an outlier is identified, it's measurement is removed from the measurement-set. When $\mathcal{H}_0$ is accepted, then and the optimization process of Section III-B is performed.

C. Complexity

Given $2m$ GPS measurements at each time over a window of length L, there are $(2mL)$ GPS residuals in each CRT window. Therefore, there are

$$\sum_{k=1}^{2mL}\binom{2mL}{k} = \sum_{k=1}^{2mL}\frac{(2mL)!}{((2mL)-k)!\ k!},$$

ways that any number of satellite measurements could have outliers in any combination at one epoch (see Section 3 of [5]). For the EKF or IEKF (with $L = 1$ and $m = 9$) this results in 262,143 hypotheses, which is too large for full consideration. For the CRT with $L = 20$, consideration of all hypothesis is infeasible for real-time implementation. Therefore, computationally simplified approaches are required. An example approach is to only consider single outlier occurrences, as discussed in the previous section, but to remove all measurements with residuals greater than a user defined threshold before completing the optimization.

D. Comparison of Degrees of Freedom

This section compares the DOF (i.e., $(p-q)$) available for outlier detection between algorithms, where q is the total number of real variables to be estimated, p denotes the total number of available constraints, n_e is the error state dimension, and m_k is the number of satellite pseudoranges available at epoch k.

The EKF at any time step has $q = n_e$ variables to estimate (one state vector) and $p = n_e + 2m$ constraints (GPS and prior); therefore, the DOF is $2m$. The DOF of the Iterated EKF (IEKF) is the same. The advantage of the IEKF is its ability to perform nonlinear iterative corrections [31], [42].

For the CRT algorithm with window length L, the number of variables to be estimated is $q = (L+1)n_e$. The number of constraints is $p = (L+1)n_e + 2mL$. The DOF is therefore, $2mL$. Both the outlier detection capability and the amount of required computation increase with the DOF, which increase with L. Also, the ability to detect outliers and the potential number of outliers both increase with L.

V. Theory for CRT with Soft-Thresholding

This section develops the theory to adapt the soft-thresholding operator [23], [24] to the CRT problem in Section III, we denote this algorithm as CRT-LSS.

A. Outlier Model

For the LSS approach, each component of $\mathbf{s}_k$ in eqn. (4) is assumed to have $i.i.d.$ Laplacian distributions [23], [24] and are assumed to be independent of $\boldsymbol{x}_k$, $\boldsymbol{\eta}_{y,k}$, $\mathbf{w}_{k-1}$:

$$p_{s_{\rho,k}}(s_{\rho,k}) = \frac{1}{2\nu_\rho} \exp\left[-\frac{|s_{\rho,k}|}{\nu_\rho}\right]$$

for pseudorange measurements and

$$p_{s_{d,k}}(s_{d,k}) = \frac{1}{2\nu_d} \exp\left[-\frac{|s_{d,k}|}{\nu_d}\right],$$

for Doppler measurements. Therefore

$$p_{\mathbf{s}_k}(\mathbf{s}_k) = \left(\frac{1}{4\nu_\rho\nu_d}\right)^m \prod_{j=1}^{m} \exp\left(-\left[\frac{|s_{\rho,j}|}{\nu_\rho} + \frac{|s_{d,j}|}{\nu_d}\right]\right).$$

To simplify notation later, let $\lambda_\rho = \frac{1}{\nu_\rho}$ and $\lambda_d = \frac{1}{\nu_d}$. The distribution of

$$p_{\mathbf{S}}(\mathbf{S}) = \prod_{n=k-L+1}^{k} p_{\mathbf{s}_n}(\mathbf{s}_n),$$

where $\mathbf{S} = \{\mathbf{s}_j \mid j \in [k-L+1, k]\}$ and $\mathbf{S} \in \mathbb{R}^{2mL\times 1}$.

B. Theoretical Solution

Modifying eqn. (5) to account for $\mathbf{S}$, the MAP estimate of the vehicle trajectory $\mathbf{X}$ is:

$$\hat{\mathbf{X}} = \underset{\mathbf{X},\mathbf{S}}{\operatorname{argmax}} \left\{p(\mathbf{X}, \mathbf{U}, \mathbf{Y}, \mathbf{S})\right\}.$$

The GPS-INS joint probability of eqn. (6) is modified as:

$$\begin{aligned} p(\mathbf{X}, \mathbf{U}, \mathbf{Y}, \mathbf{S}) &= p_{\mathbf{S}}(\mathbf{S}) p(\mathbf{X}, \mathbf{U}) p(\mathbf{Y} \mid \mathbf{X}, \mathbf{S}) \\ &= p_{\mathbf{S}}(\mathbf{S}) p(\mathbf{X}, \mathbf{U}) \prod_{j=k-L+1}^{k} p(\mathbf{y}_j | \boldsymbol{x}_j, \mathbf{s}_j) \qquad (33) \\ &= p_{\mathbf{S}}(\mathbf{S}) p(\boldsymbol{x}_{k-L}) \prod_{l=k-L}^{k-1} p(\boldsymbol{x}_{l+1}|\boldsymbol{x}_l, \mathbf{U}_l) \prod_{j=k-L+1}^{k} p(\mathbf{y}_j|\boldsymbol{x}_j, \mathbf{s}_j). \end{aligned}$$

C. Numerical Solution

Finding $\mathbf{X}$ that maximizes eqn. (33) is equivalent to minimizing the negative of its natural logarithm. This yields the equivalent nonlinear cost function:

$$\begin{aligned} \mathcal{C}(\mathbf{X}, \mathbf{S}) = {} & \|\hat{\boldsymbol{x}}_{k-L} - \boldsymbol{x}(t_{k-L})\|^2_{\mathbf{P}_{(k-L)}} \\ & + \sum_{l=k-L}^{k-1} \|\boldsymbol{\Phi}(\boldsymbol{x}(t_l), \mathbf{U}_l) - \boldsymbol{x}(t_{l+1})\|^2_{\mathbf{Q_D}} \\ & + \sum_{j=k-L+1}^{k} \left(\frac{1}{\nu_\rho}\|\mathbf{s}_\rho(t_j)\|_1 + \frac{1}{\nu_d}\|\mathbf{s}_d(t_j)\|_1\right) \\ & + \sum_{j=k-L+1}^{k} \|\mathbf{y}(t_j) - \mathbf{h}_j(\boldsymbol{x}(t_j)) - \mathbf{s}(t_j)\|^2_{\mathbf{R}}. \qquad (34) \end{aligned}$$

To simplify notation later, let $\boldsymbol{\psi}_{\rho,j} = \frac{\mathbf{s}_\rho(t_j)}{\nu_\rho} \in \mathbb{R}^m$, $\boldsymbol{\psi}_{d,j} = \frac{\mathbf{s}_d(t_j)}{\nu_d} \in \mathbb{R}^m$, and $\boldsymbol{\psi}_j = \boldsymbol{\psi}(\mathbf{s}(t_j)) = [\boldsymbol{\psi}^\intercal_{\rho,j}, \boldsymbol{\psi}^\intercal_{d,j}]$. Also, let $\boldsymbol{\Psi}(\mathbf{S}) = \{\boldsymbol{\psi}_j \mid j \in [k-L+1, k]\}$ and $\boldsymbol{\Psi} \in \mathbb{R}^{2mL\times 1}$.

Define $\boldsymbol{\Sigma}^\intercal_{\mathbf{R}}\boldsymbol{\Sigma}_{\mathbf{R}} = \mathbf{R}^{-1}$ and $\boldsymbol{\Sigma}_{\mathbf{S}} = \text{blkdiag}(\mathbf{0}, \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{R}})$, where $\boldsymbol{\Sigma}_{\mathbf{R}}$ is upper-triangular. The cost function $\mathcal{C}(\mathbf{X}, \mathbf{S})$ can be normalized using Cholesky Decomposition, such that the minimization problem of eqn. (34) reduces to the optimization of

$$\mathcal{C}(\mathbf{X}, \mathbf{S}) = \|\mathbf{z} - \mathbf{g}(\mathbf{X}) - \boldsymbol{\Sigma}_{\mathbf{S}}\mathbf{S}\|_2^2 + \|\boldsymbol{\Psi}(\mathbf{S})\|_1, \qquad (35)$$

where $\mathbf{z} = \boldsymbol{\Sigma}_{\mathbf{W}}[\hat{\boldsymbol{x}}^\intercal_{k-L}, \mathbf{0}^\intercal, \mathbf{Y}^\intercal]^\intercal$ represents the terms that are known at each iteration and

$$\mathbf{g}(\mathbf{X}) = \boldsymbol{\Sigma}_{\mathbf{W}}[\boldsymbol{x}(t_{k-L})^\intercal, \boldsymbol{\Phi}(\mathbf{X}, \mathbf{U})^\intercal - \mathbf{X}^\intercal, \mathbf{h}(\mathbf{X})^\intercal]^\intercal$$

represents the terms that are computed based on $\mathbf{X}$.

Eqn. (35) will be minimized iteratively, by the algorithm discussed in Section VI.

VI. Solution of CRT with Soft-Thresholding

The discontinuity of the l_1-norm in eqn. (35) complicates its minimization. The authors of [23], [24] provide the derivation of the Soft-Thresholding Operator for the linear case. Eqn. (35) can be optimized by methods similar to those of [23], [24], as derived below. Because the l_1-norm is not differentiable at its minimum, the optimal solution to $\mathcal{C}(\mathbf{X}, \mathbf{S})$ is solved in two parts.

A. Part 1: Estimate $\mathbf{X}$, holding $\hat{\mathbf{S}}$ constant

Given $\hat{\mathbf{S}}$, the optimal value of $\mathbf{X}$ is

$$\begin{aligned} \mathbf{X}^* &= \underset{\mathbf{X}}{\mathit{arg\,max}}\ \mathcal{C}(\mathbf{X}, \hat{\mathbf{S}}) \\ &= \underset{\mathbf{X}}{\mathit{arg\,max}} \left\{-\frac{1}{2}\|\mathbf{z} - \mathbf{g}(\mathbf{X}) - \boldsymbol{\Sigma}_{\mathbf{S}}\hat{\mathbf{S}}\|_2^2 + \|\hat{\boldsymbol{\Psi}}\|_1\right\} \\ &= \underset{\mathbf{X}}{\mathit{arg\,max}} \left\{-\frac{1}{2}\|(\mathbf{z} - \boldsymbol{\Sigma}_{\mathbf{S}}\hat{\mathbf{S}}) - \mathbf{g}(\mathbf{X})\|_2^2\right\}, \qquad (36) \end{aligned}$$

where $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}(\hat{\mathbf{S}})$. This optimization can be solved iteratively, starting from an initial value $\hat{\mathbf{X}}$, using a Taylor series expansion to approximate $\mathbf{g}(\mathbf{X})$,

$$\mathbf{g}(\mathbf{X}) = \mathbf{g}(\hat{\mathbf{X}}) + \mathbf{G}\delta\mathbf{X} + h.o.t's, \qquad (37)$$

where the Jacobian $\mathbf{G} \doteq \frac{\partial \mathbf{g}(\mathbf{X})}{\partial \mathbf{X}}\Big|_{\mathbf{X}=\hat{\mathbf{X}}}$, and $\delta\mathbf{X} \in \mathbb{R}^{n_e(L+1)}$. Substituting eqn. (37) into eqn. (36) and ignoring the higher order terms ($h.o.t's$), yields

$$\begin{aligned}\delta\mathbf{X}^* &= \mathop{arg\,min}_{\delta\mathbf{X}} \|(\mathbf{z} - \mathbf{\Sigma_S}\hat{\mathbf{S}}) - (\mathbf{g}(\hat{\mathbf{X}}) + \mathbf{G}\delta\mathbf{X})\|_2^2 \\ &= \mathop{arg\,min}_{\delta\mathbf{X}} \|\mathbf{b} - \mathbf{G}\delta\mathbf{X}\|_2^2\end{aligned}$$

where in this section $\mathbf{b}$ is defined as

$$\mathbf{b} \doteq \mathbf{z} - \mathbf{\Sigma_S}\hat{\mathbf{S}} - \mathbf{g}(\hat{\mathbf{X}}). \tag{38}$$

Then, the optimal $\delta\mathbf{X}$ solves

$$\mathbf{G}\delta\mathbf{X}^* = \mathbf{b} \tag{39}$$

in the least squares sense. The update to the trajectory is performed using (17).

Optimization with respect to $\mathbf{X}$ is obtained by iterating eqns. (37), (38), and the solution of (39), to convergence for some user defined stopping conditions. A line search is implemented in the direction of $\delta\mathbf{X}^*$ from eqn. (39) to determine the magnitude of the update step in eqn. (17).

B. Part 2: Estimate $\mathbf{S}$, holding $\hat{\mathbf{X}}$ constant

Given an estimate $\hat{\mathbf{X}}$, the optimal value of $\mathbf{S}$ is

$$\begin{aligned}\mathbf{S}^* &= \mathop{arg\,max}_{\mathbf{S}}\ \mathcal{C}(\hat{\mathbf{X}}, \mathbf{S}) \\ &= \mathop{arg\,max}_{\mathbf{S}} \left\{-\frac{1}{2}\|\mathbf{z} - \mathbf{g}(\hat{\mathbf{X}}) - \mathbf{\Sigma_S}\mathbf{S}\|_2^2 + \|\mathbf{\Psi}(\mathbf{S})\|_1\right\}. \\ &= \mathop{arg\,max}_{\mathbf{S}} \left\{-\frac{1}{2}\|\mathbf{c} - \bar{\mathbf{S}}\|_2^2 + \|\mathbf{\Psi}(\mathbf{S})\|_1\right\}, \\ &= \mathop{arg\,min}_{\mathbf{S}} \sum_i \left[\frac{1}{2}\left(c_i - \bar{s}_i\right)^2 + \|\psi(s_i)\|_1\right],\end{aligned} \tag{40}$$

where $\bar{\mathbf{S}} = \mathbf{\Sigma_S}\mathbf{S}$, and $\mathbf{c} = \mathbf{z} - \mathbf{g}(\hat{\mathbf{X}})$ is independent of $\mathbf{S}$. In (40), the non-boldface quantities c_i, $\bar{s}_i$, and $\psi(s_i)$ denote the scalar elements of the vectors in the previous equation.

Each term in the summation in eqn. (40) only depends on c_i, $\bar{s}_i$, and $\psi(s_i)$, thus each term can be optimized independently[1]:

$$\hat{s}_i = \mathop{arg\,min}_{s_i} \left\{\frac{1}{2}\left(c_i - \bar{s}_i\right)^2 + |\psi(s_i)|\right\}. \tag{41}$$

The closed-form solution of the optimization problem in eqn. (41) is the soft-thresholding operation (see eqn. 16 of [24]) resulting in two equations, one for pseudorange and one for Doppler:

$$\hat{s}_{\rho,i} = \sigma_\rho\ \mathrm{sgn}(c_i)\ \ \max\left(|c_i| - \frac{\sigma_\rho}{\nu_\rho}, 0\right), \tag{42}$$

$$\hat{s}_{d,i} = \sigma_d\ \mathrm{sgn}(c_i)\ \ \max\left(|c_i| - \frac{\sigma_d}{\nu_d}, 0\right), \tag{43}$$

where $\mathrm{sgn}(\cdot)$ represents the *signum* function. The proof is provided in Section 7 of [29].

C. CRT-LSS Summary

Implementation of the method in Sections VI-A and VI-B is summarized in Algorithm 1.

Algorithm 1 Least Soft-threshold Squares Regression

```
1: Given initial $\hat{\mathbf{X}}$ and $\hat{\mathbf{S}}$. Initialize $i = 0$.
2: while $(((\|\delta\mathbf{X}\|_2$ & $\|\delta\hat{\mathbf{S}}\|_2) > 1 \times 10^{-3})$ & $(i < 20))$
3:     $i = i + 1$
4:     Compute $\delta\mathbf{X}$ and $\hat{\mathbf{X}}^+ = \hat{\mathbf{X}}^- \oplus \delta\mathbf{X}$ (eqns. (38), (39)).
5:     if $(\|\delta\mathbf{X}\|_2 < 1 \times 10^{-3})$
6:         Compute $\hat{\mathbf{S}}$ using eqns. (42) & (43).
7:         (Optional) Compute $\delta\mathbf{X}$ and $\hat{\mathbf{X}}^+ = \hat{\mathbf{X}}^- \oplus \delta\mathbf{X}$.
8:     end
9: end
```

In Section VI-A the value of the cost function $\mathcal{C}(\mathbf{X}, \mathbf{S})$ is decreased by changing only $\mathbf{X}$, even if multiple nonlinear least squares iterations occur. In Section VI-B the value of the cost function $\mathcal{C}(\mathbf{X}, \mathbf{S})$ is decreased by changing only $\mathbf{S}$, only a single iteration is required, and $\delta\hat{\mathbf{S}}$ is the final $\mathbf{S}$ minus the initial $\mathbf{S}$. Therefore, at each iteration, the cost function is decreased and is bounded below. Ultimately at least a local minimum of the cost function will be attained. Note that line 7 of Algorithm 1 is not necessary, but is useful for analyzing the convergence of the cost function $\mathcal{C}(\mathbf{X}, \mathbf{S})$.

The combined operation of Sections VI-A and VI-B are equivalent to the Huber Loss function (see [24], [25]).

VII. Illustrative Example

This section evaluates the CRT-HT and CRT-LSS algorithm performance using two data sets:

1) Synthetic-sensor data provides controlled conditions where ground truth and outliers are known.
2) Real-sensor data is used to demonstrate algorithm performance under real-world conditions, where ground truth is estimated.

A. Data Descriptions

The synthetic-data is a simulated 515 second automobile trajectory generated by a signal generator incorporating a 6DOF kinematic model to produce both ground truth and noise-corrupted IMU and GPS "measurements." The simulated trajectory is similar to the real-world trajectory discussed below.

IMU measurements $\tilde{u}_i(t)$ for the i^{th} axis of the gyroscope and accelerometer were generated at $200Hz$ according to the sensor model (Ch. 11.6 of [28]):

$$\tilde{u}_i(t) = (1 + \epsilon_i)[u_i(t) + b_i(t) + \eta_{RW,i} + \eta_{MA,i}],$$

where $u_i(t)$ is actual acceleration (or angular rotation rate) for axis i, $\eta_{RW,i}$ is white noise, ϵ_i is the scale factor error, and $\eta_{MA,i}$ is the misalignment error. The time-correlated error (i.e., bias) $b_i(t)$ has a random initial value and performs a random walk over time with driving noise $\eta_{RRW,i}$. The accelerometer has $\sigma_{RW} = 0.05\ (m/s)/\sqrt{Hr}$, bias variation $=$ 15 *milli-g*, and bias

[1] When $\mathbf{R}$ and hence $\mathbf{\Sigma_R}$ are diagonal, this is obvious. When $\mathbf{R}$ is not diagonal, the fact that $\mathbf{\Sigma_R}$ is upper-diagonal allows (40) to be solved one component at a time using back-substitution. For brevity, the algorithm below only presents the diagonal case.

stability $= 0.1$ *milli-g*. The constant along-axis scale-factor errors are 200 *parts-per-million (ppm)*, and cross-axis scale-factor errors are 150 *ppm*. The gyro was modeled with angle random walk parameter $\sigma_{RW} = 0.1\ deg/\sqrt{Hr}$, bias variation $= 3.0\ deg/Hr$, and bias stability $= 1.0\ deg/Hr$. The constant along-axis g-sensitivity errors are $2\ deg/Hr/g$, along-axis scale-factor errors are 200 *ppm*, and cross-axis scale-factor errors are 150 *ppm*.

The GPS signal generator provided L1 pseudorange and Doppler measurements at $1Hz$ for both the vehicle and base station using the models in Sections 8.2 and 8.3 of [28]. Satellite vehicle orbits were produced from Receiver Independent Exchange (RINEX) ver. 2.10 [43] files downloaded from a Continuously Operating Reference Station (CORS) server [44], and valid for January 10, 2017, between 9:00-10:00 am local time.

Real-world performance is evaluated using a 515 second drive-test data around the University of California, Riverside campus. While driving, the sensor data is time-stamped and stored. The sensor data includes: a MEMS IMU (Nav Technology Co., Ltd. NV-IMU100) which outputs specific force and angular rate measurements at $200Hz$, and an L1/L2 GPS receiver (NovAtel OEMV3) which outputs pseudorange, Doppler, and carrier phase measurements at $1Hz$. A GPS antenna (Leica AT502-667126) is mounted on the vehicle roof. Differential corrections were obtained in real-time, via cellular connection, from the UC Riverside NTRIP [45] caster (`ntrip.engr.ucr.edu:2101`), which broadcasts raw dual frequency GPS measurements (Message 1004 in RTCM3.1 standard) and the base position (Message 1006 in RTCM3.1 standard) publicly over the internet at $1Hz$ and $0.1Hz$, respectively [46]. The base station uses the same model of GPS receiver and antenna as the vehicle, and runs both the NTRIP Server and NTRIP Caster (both developed in C++) in Linux (Ubuntu 12.02) on a mini-desktop computer with an Ethernet connection. The trajectory contains a variety of real-world conditions that can adversely affect GPS receiver performance, e.g. tall buildings and trees, thereby producing measurement outliers. Fig. 1 shows snapshots along this trajectory.

For the experimental data, the *ground truth* trajectory is found by solving a nonlinear MAP optimization problem over the entire (515 second) trajectory using the L1/L2 integer-resolved carrier phase DGPS and IMU measurements to achieve centimeter level ground truth accuracy [47].

To allow direct comparison of the performance of various algorithms, using identical input data, the results of this section are computed during post-processing. Even though running in post-processing for this evaluation, each algorithm is written in C++ to run in real-time, using only the data and prior as would be applicable for each approach. These real-time navigation algorithms only use L1 GPS with differential corrections, and IMU data.

At the start of the experiment, the vehicle is stationary and pointed approximately north. The estimated yaw angle is initialized to 180° (i.e., south). During the first one second of this stationary period, the initial roll and pitch are computed from the direction of the specific-force (i.e, gravity) vector. The initial position is obtained by least-squares using the first GPS double-differenced L1 pseudorange measurement. The initial biases and velocity values are all zero.

B. CRT-HT Performance

1) Synthetic Data: Using the simulated trajectory, 216 simulations were run consisting of: six injected outlier magnitudes ($O = \{1.0, 1.5, 2.0, 2.5, 5.0, 10.0\}m$), six CRT window lengths ($L = \{1, 5, 10, 20, 30, 40\}$), and six threshold values ($\gamma = \{0.75, 1.00, 1.25, 1.50, 2.00\}$).

To simulate driving near trees or past a tall building, *multiple simultaneous outliers* were added at three known trajectory locations. The outliers persisted for five seconds. The outliers were added to three of the ten available pseudorange measurements representing satellites that were visible in one quadrant of the sky. The minimum outlier magnitude of $1.0m$ was selected such that the outlier is similar in magnitude to double-differenced pseudorange measurement noise.

Fig. 2 presents the receiver operating characteristic (ROC) curve for several algorithms, e.g. IEKF and CRT-HT with length L. Each ROC curve plots the Probability of Correct Detection for one setting of O and L (for CRT-HT), versus Probability of False Alarm, as a function of γ.

Since this is simulation, the correct outlier decisions are known for each satellite at each epoch. To construct the ROC curve for each algorithm and each outlier magnitude, the procedure discussed in the next paragraph is repeated once for each specified value of γ. The outlier detection decisions for each algorithm (IEKF and CRT-HT with length L) and each value of γ are compared with the known correct decisions to compute P_{CD} and P_{FA}. Each simulation for a single value of γ generates one point on the ROC curve.

For example, to evaluate the CRT-HT $L = 5$, a significance threshold α is chosen and γ is computed, then the CRT-HT is run for the entire trajectory. At the completion of the trajectory, the P_{CD} (of outliers) is calculated, as well as the P_{FA}, and the values are recorded. This provides one point on the CRT-HT $L = 5$ ROC curve. The trajectory estimation is repeated for a vector of α values. This set of values provides the CRT-HT $L = 5$ ROC curve. This process is performed for each outlier magnitude, and each estimator, e.g. IEKF, and CRT-HT with $L = \{5, 10, 20, 30, 40\}$. For clarity of the graph, only IEKF and CRT-HT $L = \{5, 10, 40\}$ are shown here, however the trends for $L = \{20, 30\}$ are similar. The outlier detection algorithm evaluates the entire residual vector before optimization of the trajectory. If outliers are detected, the corresponding measurements are removed, and the optimization is performed. Upon sliding the window, the outliers from the previous time-window are ignored and the detection procedure searches the entire residual vector.

Each curve in Fig. 2 displays P_{CD} vs. P_{FA}. The trends for γ are as expected. As γ increases, the next point on each curve will be below and to the left of the last point, as P_{CD} and P_{FA} both decrease as γ increases. As Fig. 2 shows, the rates of decrease are very different for each algorithm.

Curves closer to the upper left corner show improved ability to detect outliers with lower probability of injecting false alarms.

The advantage of increasing the CRT-HT window length is demonstrated by the increase in P_{CD} and decrease in P_{FA} as L increases. For example in Fig. 2, consider points P_1 and P_2. Both points have outlier magnitude of $2.0m$ and $\gamma = 1.25$. Point P_1 for the IEKF has $P_{CD} = 48\%$ and $P_{FA} = 12\%$, whereas point P_2 for the CRT-HT $L = 5$ has $P_{CD} = 75\%$ and $P_{FA} = 8\%$. While both algorithms perform nonlinear optimization, it is the window length that contributes to the 27% increase in P_{CD} and a 4% decrease in P_{FA}. As CRT-HT window length increases, for a given value of γ P_{CD} increases and P_{FA} decreases.

This confirms the claim that the ability of the CRT-HT algorithm to discriminate outliers from valid data is enhanced with the length of the sliding window, which increases the redundancy as quantified by the number of degrees of freedom. The reliability of achieving a specified level of accuracy increases with the ability to remove outliers.

2) Experimental Data: For each algorithm (EKF, IEKF and CRT-HT with length L), Fig. 3 shows the cumulative distribution function (CDF) of the position error norm $\|\hat{\mathbf{p}}_k - \mathbf{p}_k\|$ at 1 Hz where the ground truth trajectory is used for $\mathbf{p}_k$ and $\hat{\mathbf{p}}_k$ is the result of the optimization at the first time when the k-th epoch enters the sliding window. For outlier removal, $\gamma = 1.25$. The CRT-HT algorithm curves are included for various window lengths L.

Fig. 3 shows that the percentage of occurrences where the EKF position error is less than $0.1m$, is roughly 18%. Roughly 90% of the EKF trajectory has errors less than $1.0m$. Figs. 3 also indicate that accuracy improves from the EKF to the IEKF to the CRT-HT. Also, CRT-HT performance (generally) improves with the window length L. For the CRT-HT with $L > 5$, 100% of the position errors are less than $1.0m$. CRT-HT algorithms with $L > 20$ each achieve $0.6m$ position accuracy on 100% of the trajectory. The EKF and IEKF CDF plots do not reach 100% until the position accuracy is greater than $3.0m$.

Fig. 4 plots variables related to the CRT-HT ($L = 20$) estimator and its optimization process. The top plot shows the total number of iterations, which is typically two, except at initialization and immediately after $t = 20$ seconds. Because the vehicle is stationary for the first 20 seconds, the state is not fully observable until after that time. At the start of the trajectory it takes more iterations to converge from its initial value to an estimate $\hat{\mathbf{X}}$ such that the remaining error $\delta\mathbf{X}$ lies within the unobservable space. That unobservable space includes the yaw error and linear combinations of the bias and attitude errors. Fig. 5 plots the actual and estimated yaw angle. To produce the data for Figs. 4 and 5, the yaw estimate is intentionally initialized to be in the opposite direction from the actual yaw angle. With this worst case initialization, the EKF diverges (not shown) after $t = 20$ when the vehicle accelerates, whereas the CRT yaw estimate converges rapidly. This demonstrates the ability of the CRT to correct large initialization errors that the EKF cannot. The second graph of Fig. 4 shows the optimal cost (i.e., value when the iteration stops) at each time instant. Note that without GPS measurements, the cost will be zero. As the number of GPS measurements in the window increases, the optimal solution requires tradeoffs between the prior, the INS, and the GPS terms in the cost function; therefore, the optimal cost typically increases with the number of measurements available within the window. The cost is also affected by the geometric arrangement of the available satellites and their multipath and measurement noise. The number of satellites available at each time is plotted in the third graph along with the number of rejected satellites. The number used from each epoch is the difference between the number available and rejected. The fourth graph shows the horizontal position error (relative to ground truth) in blue and the prediction of two times the error standard deviation in black, computed based on the covariance extracted from the estimator. To compute the position error, we use the position $\hat{\mathbf{p}}(t)$ from $\hat{\boldsymbol{x}}(t)$ at time t (i.e., the first time instant when $\mathbf{p}(t)$ enters the sliding window). The final graph plots two curves. The blue curve is the GDOP of only the satellites available at the current epoch. The red curve is the GDOP computed by stacking the line-of-sight vectors available over all epochs within the window. The red curve shows that in the $L = 20$ second window, for this data set, for all time instants except one (near 175 s), the sliding window maintains a set of measurements from satellites with diverse geometry, even though the blue curve shows that there are many time instants with insufficient geometric diversity of satellites for a pointwise solution. As the available satellites within the window become less recent, the position error, and the prediction of that error by its standard deviation both increase. This ability to predict growth in the state estimation error is essential for safety-of-life applications.

A few time intervals deserve additional discussion. The approach removes satellite measurements during three different time intervals. For $t \in [33, 53]$, there are 7 satellites available and the algorithm removes 2. During this time, the vehicle is making two consecutive right turns around a police station onto University Avenue and then onto Linden Avenue. There is significant tree cover along Linden. During this time interval, the satellites available within the window have a sufficiently diverse geometry (i.e., GDOP) such that both the position error and its predicted standard deviation remain small. At $t = 186$ all 5 available satellites are removed. This is immediately following the interval $t \in [165, 178]$ when the vehicle is on North Campus Dr. between Winston Chung Hall ($23.2m$ tall) and a hill with tall trees. During this time interval, the time elapsed since the satellite geometry was sufficient for position observability is growing; therefore, both the position error and its predicted standard deviation grow with the elapse time, until a sufficiently diverse set of satellites return to view. The figure shows similar growth for $t \in [231, 242]$ when the vehicle drives on East Campus Dr. between the Chemical Sciences building ($21.6m$ tall) and Orbach Library ($17.9m$ tall); for $t \in [381, 410]$, when the vehicle is on West Campus Dr. under overhanging trees and adjacent to Hinderaker Hall ($17.3m$ tall); and for $t \in [430, 440]$, when the vehicle is adjacent to the Fine Arts building ($20.1m$ tall) on West Campus Drive with trees and a church across the street. In this last interval, the algorithm removes two satellites.

C. CRT-LSS Performance

1) Specification of Laplacian Parameters: The distribution of the outliers is not known. The CRT-LSS derivation of Section V-A assumed a Laplacian distribution. For implementation, reasonable values for the parameters of that distribution need to be specified. Our approach is to specify the standard deviation of the Laplacian outlier to match a multiple of the standard deviation assumed for the Gaussian noise. For the Laplacian distribution, the standard deviation $\sigma_{\mathcal{L}}$ is related to the parameter ν according to $\sigma_{\mathcal{L}} = \sqrt{2}\nu$.

Let $\sigma_{\mathcal{L}_\rho}$ and $\sigma_{\mathcal{L}_d}$ denote the standard deviation of the pseudorange and Doppler outliers. Let $\sigma_{\mathcal{N}_\rho}$ and $\sigma_{\mathcal{N}_d}$ denote the standard deviation of the pseudorange and Doppler measurements. Setting $\sigma_{\mathcal{L}_\rho}, = \lambda\sigma_{\mathcal{N}_\rho}$ and $\sigma_{\mathcal{L}_d} = \lambda\sigma_{\mathcal{N}_d}$ yields $\nu_\rho = \frac{\lambda}{\sqrt{2}}\sigma_{\mathcal{N}_\rho}$ and $\nu_d = \frac{\lambda}{\sqrt{2}}\sigma_{\mathcal{N}_d}$. This selection of the Laplace distribution parameters will cause residuals $\frac{\lambda}{\sqrt{2}}$larger that the expected Normal standard deviation to be soft-thresholded.

2) Synthetic data: Similar to the process discussed in Section VII-B.1, 216 simulations were performed, wherein the value of λ was selected and held constant for the entire trajectory. This process is repeated for a set of values of λ, for each estimator and outlier magnitude to evaluate sensitivity of P_{CD} vs. P_{FA} versus λ for each algorithm. Due to space constraints, only position error data are discussed herein. The velocity and attitude error data are provided in Section 6 of [29].

The CRT-LSS algorithm does not perform detection in the same sense as the CRT-HT algorithm. Therefore the values for P_{CD} and P_{FA} are computed relative to ground truth using the indices where $\hat{\mathbf{S}}$ is non-zero, e.g. measurements which required soft-thresholding (see line 6 of Alg. 1).

Each ROC curve in Fig. 6 displays the expected P_{CD} vs. P_{FA} trends versus λ, meaning that the next point on each curve will be below and to the left of the last point as λ is increased. This is because, with increasing the λ value, larger residuals will not be soft-thresholded, therefore both P_{CD} and P_{FA} will decrease. Curves closer to the upper left corner show improved ability to accommodate outliers with lower probability of injecting false alarms.

The advantage of increasing the CRT-LSS window length is demonstrated in the increase in P_{CD} and decrease in P_{FA} for a given value of λ. For example in Fig. 6, consider two points P_1 and P_2 each with the outlier magnitude of $2.0m$ and $\lambda_\rho = 1.0$. For the IEKF $P_{CD} = 52\%$ and $P_{FA} = 14\%$, whereas for the CRT-LSS $L = 5$ the $P_{CD} = 70\%$ and $P_{FA} = 12\%$. While both algorithms perform nonlinear optimization, it is the window length that contributes to the 18% increase in P_{CD} and a 2% decrease in P_{FA}. As CRT-LSS window length increases, the P_{CD} increase and P_{FA} decrease.

3) Experimental Data: Figs. 7 shows the CDF of the norm of the position error $\|\hat{\mathbf{p}}_k - \mathbf{p}_k\|$ for various estimation algorithms. For outlier accommodation in this section, $\lambda = 1.00$. The EKF curves use hypothesis testing while the IEKF and CRT curves use LSS. Fig. 7 shows that the CRT-LSS performance improves with the window length L. For example, in the CRT-LSS approach with $L > 5$, 97% of the position errors are less than $1.0m$. The CRT-LSS algorithms with $L > 10$ each achieve $1.0m$ position accuracy on 100% of the trajectory. Alternatively, the IEKF CDF plots do not reach 100% until the position accuracy is greater than $3.0m$.

Fig. 8 graphs several key variables plotted versus GPS epoch for the CRT-LSS estimator with $L = 20$. The top graph shows the total number of iterations. Comments similar to those for Fig. 4 apply. The second graph shows the cost function (blue) and $10\|\mathbf{s}\|_1$ (red), where the 10-fold scaling is used to make the non-zero values near $t = 325s$ clearly visible. The third graph plots the number of satellites available (blue) and the number removed (i.e., $\|\mathbf{s}\|_0$ in red). The fourth graph plots the horizontal position error (blue) and its predicted standard deviation (red), both computed as discussed relative to Fig. 4. The final graph plots two curves. The blue curve is the GDOP of only the satellites available at the current epoch. The red curve is the GDOP computed by stacking the line-of-sight vectors available over all epochs within the window.

The same time periods identified and described in the discussion of Fig. 4 are again of interest. The approach soft-thresholds satellites during two different time intervals. For $t \in [33, 53]$, there are 7 satellites available and the algorithm initially soft-thresholds one, then two satellites. During this time, the vehicle is making two consecutive right turns around a police station onto University Avenue and then onto Linden Avenue. There is significant tree cover along Linden. During this time interval, the satellites available within the window have a sufficiently diverse geometry (i.e., GDOP) such that both the position error and its predicted standard deviation remain small. During the interval $t \in [318, 335]$, the vehicle is on North Campus Dr. at the top of the hill where tall overhanging trees are present. In this interval, only one satellite is soft-thresholded. However, since the satellite geometry was sufficient for position observability, both the position error and its predicted standard deviation remain nearly unchanged.

D. Comparison: CRT-LSS vs. CRT-HT

As shown in the preceding figures and discussion, both CRT-HT and CRT-LSS clearly show that (with the exact same data) both accuracy and reliability are enhanced by increasing L.

Both CRT-HT and CRT-LSS successfully accommodate outliers. CRT-HT does so with explicit detection and removal, while CRT-LSS does so through soft-thresholding of the residuals through an optimization process. The ROC curves show each approach may exhibit superior performance depending on parameter settings. Where the implementation of CRT-HT is a compromise between theory and practicality, the CRT-LSS is a direct numerical implementation of the theory. As discussed in Section IV-C, the complete alternate hypothesis testing procedure is infeasible in real-time with CRT-HT window $L > 1$, therefore a modified approach is implemented numerically. Alternatively, the soft-thresholding approach of [23], [24] as adapted to the CRT problem in Section VI does not remove any measurements. Instead, CRT-LSS automatically determines which residuals should be soft-thresholded in each time-window, accounting for outliers in the MAP optimization process.

Due to the modified approach used in CRT-HT, the computational cost is similar to CRT-LSS (see Section 4 of [29]). Both methods have similar performance with respect to P_{FA} and P_{CD}, as well as position and attitude error for a given window length L.

The CRT estimator (HT or LSS) has several advantages over the EKF which are summarized here:

1) CRT has the ability to change the linearization point of the trajectory within the CRT window. For the EKF the linearization point is the prior. Errors in the prior if large relative to the higher order terms of the linearization can cause the EKF to diverge.
2) In the CRT approach, multiple iterations, each with re-linearization, are possible to fully address the non-linearities in the MAP optimization, whereas the EKF performs a single iteration per measurement.
3) The improved performance demonstrated in Figs. 2–8 is attributed to solving the full nonlinear optimization over a longer window with outlier accommodation. The longer window enhances the redundancy and allows reconsideration of fault decisions, as long as the measurement data is within the sliding window. The standard EKF utilizes a single epoch. Thus an incorrect fault decision can be catastrophic and the redundancy available is often insufficient to make confident fault decisions.

This enhanced ability to accommodate outliers to achieve reliable performance is one of the major motivations of the CRT approach.

VIII. Conclusion

[2] **This article presented two methods to enhance the level of redundancy in a GNSS and IMU based navigation system to facilitate the accommodation of outlier measurements. Over a multiple epoch sliding window of data the algorithm performs MAP estimation within a nonlinear optimization framework, while maintaining a real-time estimate as necessary for control and planning purposes. Increasing the duration L of the sliding window enhances redundancy at the expense of increased computation. Enhancing redundancy improves the reliability of achieving any given accuracy specification, by better outlier removal. The MAP framework, through real-time nonlinear optimization, achieves optimal state estimation without linearization assumptions. The enhanced performance of these methods is demonstrated through direct comparisons of both the accuracy and outlier detection abilities of various algorithms using experimental data from a challenging environment. Moreover, while the CRT-HT and CRT-LSS have similar outlier accommodation performance, the CRT-HT is a compromise of theory and practicality, whereas the CRT-LSS is a direct numerical implementation of the theory.**

Related areas of interest for future research include accommodation of time correlated errors (e.g., multipath), and adaptation of the window length L again trading off risk and performance.

IX. Acknowledgments

The authors thank Seiko-Epson USA for providing the M-G320 Quartz-MEMS IMU for this research.

References

[1] R. G. Brown, “A Baseline RAIM Scheme and a Note on the Equivalence of Three RAIM Methods,” *Proc. Nat. Tech. Mtng. ION, San Diego, CA*, pp. 127–137, 1992.
[2] ——, “Solution of the Two-Failure GPS RAIM Problem Under Worst Case Bias Conditions: Parity Space Approach,” *Navigation, J. of ION*, vol. 44, no. 4, 1997-98.
[3] P. Y. Hwang and R. G. Brown, “From RAIM to NIORAIM: A New Integrity Approach to Integrated Multi-GNSS Systems,” *Inside GNSS*, May-June, 2008.
[4] M. A. Sturza, “Navigation System Integrity Monitoring Using Redundant Measurements,” *Navigation, J. ION*, vol. 35, no. 4, 1988-89.
[5] J. Angus, “RAIM with Multiple Faults,” *J. ION*, vol. 53, no. 4, 2006.
[6] S. Hewitson and J. Wang, “Extended Receiver Autonomous Integrity Monitoring (eRAIM) for GNSS/INS Integration,” *J. of Surveying Engineering*, vol. 136, no. 1, pp. 13–22, 2010.
[7] A. H. Jazwinski, “Limited Memory Optimal Filtering,” *IEEE T. Automatic Control*, vol. 13, no. 5, pp. 558–563, 1968.
[8] P. E. Moraal and J. W. Grizzle, “Observer design for nonlinear systems with discretetime measurements,” *IEEE TAC*, vol. 40, p. 1995, 395404.
[9] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental Smoothing and Mapping,” *IEEE T. Rob.,*, vol. 24, no. 6, pp. 1365–1378, 2008.
[10] F. Dellaert and M. Kaess, “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing,” *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.
[11] R. Eustice, H. Singh, and J. Leonard, “Exactly Sparse Delayed-State Fiters for View-Based SLAM,” *IEEE T. Rob.,*, vol. 22, no. 6, pp. 1552–3098, 2006.
[12] K. R. Muske, J. B. Rawlings, and J. H. Lee, “Receding horizon recursive state estimation,” *American Control Conference*, pp. 900–904, 1993.
[13] D. Q. Mayne and H. Michalska, “Adaptive receding horizon control for constrained nonlinear systems,” *IEEE CDC*, pp. 1286–1291, 1993.
[14] G. Zimmer, “State observation by on-line minimization,” *Int. J. Control*, vol. 60, no. 4, pp. 595–606, 1994.
[15] E. L. Haseltine and J. B. Rawlings, “A critical evaluation of extended kalman filtering and moving horizon estimation,” March 12, 2003.
[16] S. Zhao, Y. Chen, H. Zhang, and J. A. Farrell, “Differential GPS Aided Inertial Navigation: A Contemplative Realtime Approach,” *19th IFAC World Congress*, pp. 8959–8964, 2014.
[17] A. S. Willsky, “A Survey of Design Methods for Failure Detection in Dynamic Systems,” *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.
[18] A. S. Willsky and E. Y. Chow, “Issues in the development of a general design algorithm for reliable failure detection,” 1980, pp. 1006–1012.
[19] ——, “Analytical Redundancy and the Design of Robust Failure Detection Systems,” *IEEE TAC*, vol. AC-29, no. 7, pp. 603–614, 1984.
[20] R. J. Patton, “Fault detection and diagnosis in aerospace systems using analytical redundancy,” *Computing and Control Engineering Journal*, vol. 2, no. 3, pp. 127–136, 1991.

[2] JAF to edit

[21] P. F. Roysdon and J. A. Farrell, “GPS-INS Outlier Detection and Elimination using a Sliding Window Filter,” *IEEE, In Proc. of American Control Conf.*, 2017.

[22] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust Face Recognition via Sparse Representation,” *IEEE T. PAMI*, vol. 31, no. 2, 2009.

[23] X. Mei and H. Ling, “Robust Visual Tracking using L-1 Minimization,” *ICCV*, 2009.

[24] D. Wang, H. Lu, and M. Yang, “Robust Visual Tracking via Least Soft-threshold Squares,” *IEEE T. on Circuits and Systems for Video Technology*, 2015.

[25] P. Huber, *Robust Statistics*. NY: John Wiley and Sons Inc., 1986.

[26] P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, New York: John Wiley and Sons Inc., 1987.

[27] P. F. Roysdon and J. A. Farrell, “Robust GPS-INS Outlier Accomodation using a Sliding Window Filter,” *22th IFAC World Congress*, 2017.

[28] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[29] P. F. Roysdon and J. A. Farrell. (2017, May) “Technical Note: CRT with Hypothesis Testing and Least Soft-thresholded Squares”. [Online]. Available: www.escholarship.org/uc/item/4b0289qj

[30] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *T. of the ASMEJ. of Basic Eng.*, vol. 82, no. D, pp. 35–45, 1960.

[31] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press: Mathematics in Science and Engineering, 1979.

[32] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory*. Prentice Hall PTR, 2013.

[33] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *Intl. J. Rob. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[34] A. Mourikis and S. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation,” in *ICRA*, 2007, pp. 3565–3572.

[35] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual–inertial odometry,” *Int. J. of Rob. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[36] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, *Bundle Adjustment - A Modern Synthesis Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Berlin / Heidelberg, 2000, vol. 1883.

[37] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3, pp. 591–611, 1965.

[38] T. Anderson and D. Darling, “A Test of Goodness-of-Fit,” *Journal of the American Statistical Association*, vol. 49, no. 765-769, 1954.

[39] N. Smirnov, “Table for estimating the goodness of fit of empirical distributions,” *Ann. Math. Stat.*, vol. 19, pp. 279–281, 1948.

[40] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. II - Detection Theory*. Prentice Hall PTR, 1998.

[41] R. A. Fisher, “Statistical Methods for Research Workers,” *Edinburgh, UK: Oliver and Boyd*, p. 43, 1925.

[42] A. H. Jazwinski, *Stochastic Processes and Filtering*. Academic Press: Mathematics in Science and Engineering, 1970.

[43] W. Gurtner. (2007, Dec) “RINEX: the receiver independent exchange format, Version 2.10”. [Online]. Available: http://igscb.jpl.nasa.gov/igscb/data/format/rinex210.txt

[44] R. A. Snay and M. Tomás Soler, “Continuously operating reference station (CORS): History, applications, and future enhancements,” *J. of Surveying Engineering*, vol. 134, no. 4, 2008.

[45] P. Gary and C. E. Fly, “RTCM Standard 10410.1, Networked Transport of RTCM via Internet Protocol (Ntrip), Version 2.0,” Radio Technical Commission for Maritime Services, Tech. Rep., 2011.

[46] RTCM Special Committee No. 104, “RTCM Standard 10403.1, Differential GNSS Services, Version 3,” Radio Technical Commission for Maritime Services, Tech. Rep., 2011.

[47] A. Vu and J. A. Farrell, “Feature mapping and state estimation for highly automated vehicles,” *J. of Control and Decision*, vol. 2, no. 1, pp. 1–25, 2015.

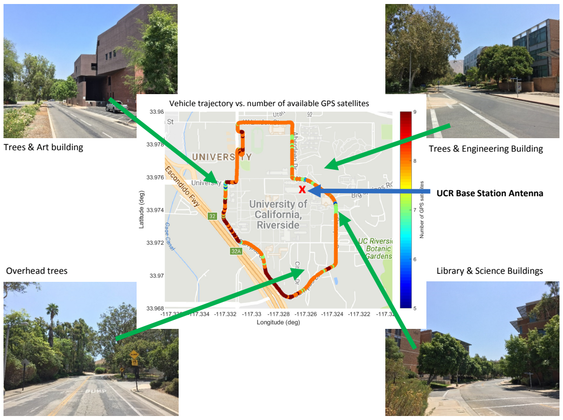

Fig. 1. Test trajectory around UCR campus. The corner photos identify a variety of real-world environmental factors which adversely affect the performance of a GPS receiver, e.g. trees and tall buildings. The center image shows the number of satellites *available* along the trajectory for estimation and outlier detection. The UCR base station antenna position is shown with a red ‘X’.

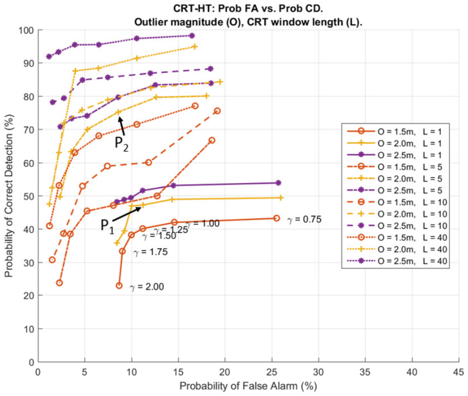

Fig. 2. Receiver Operating Characteristic curves for each CRT-HT algorithm, outlier magnitude and γ. Values for γ correspond to curve ($O = 1.5m$, $L = 1$), however the same values apply to the corresponding tick marks on the other curves. Outlier magnitudes $> 2.5m$ have $P_{CD} = 100\%$. Increasing window length L improves P_{CD} regardless of outlier magnitude. Tick-mark style (e.g. circle, star, etc.) and line color correspond to outlier magnitude. Line style (e.g. solid, dashed, etc.) correspond to CRT window length.

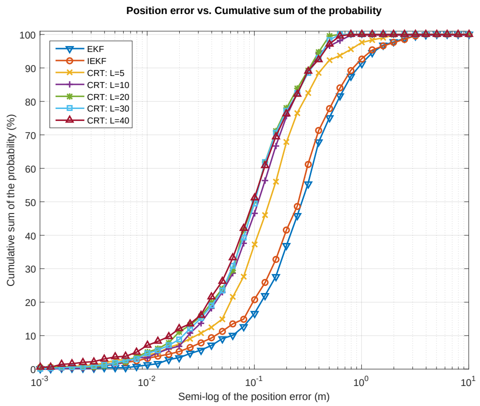

Fig. 3. Position error cumulative distributions for each HT algorithm. Based on synthetic data results, the value $\gamma = 1.25$ was selected for all curves to guarantee minimum $P_{CD} > 50\%$.

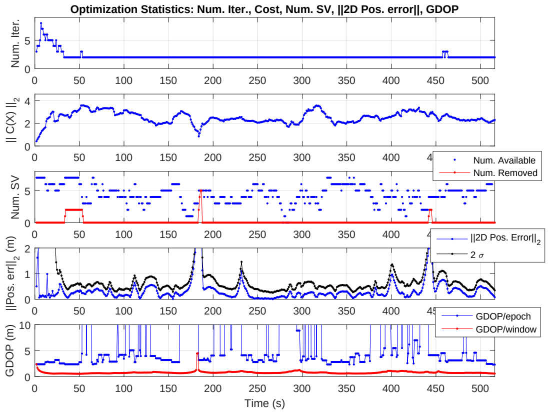

Fig. 4. Optimization Statistics for experimental data trajectory using CRT-HT with $L = 20$ and $\gamma = 1.00$, including: total number of iterations, final $\|\mathcal{C}(\hat{\mathbf{X}})\|_2$, final $\|\boldsymbol{\xi}\|_2$, and final $\|\delta\hat{\mathbf{X}}\|_2$.

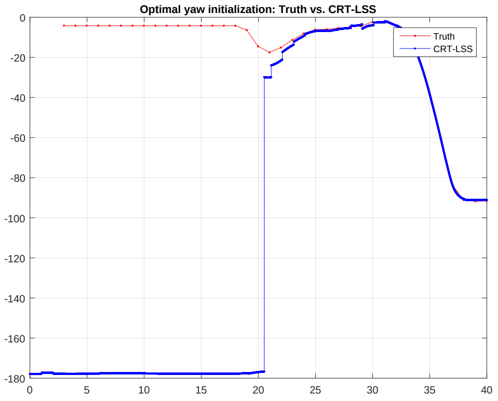

Fig. 5. CRT-LSS optimal yaw estimation.

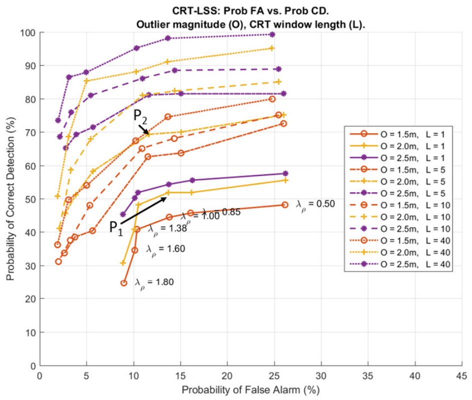

Fig. 6. Receiver Operating Characteristic curves for each CRT-LSS algorithm, outlier magnitude, and λ_ρ. Values for λ_ρ correspond to curve ($O = 1.5m$, $L = 1$), however the same values apply to the corresponding tick marks on the other curves. Outlier magnitudes $> 2.5m$ have $P_{CD} = 100\%$. Increasing window length L improves P_{CD} regardless of outlier magnitude. Tick-mark style (e.g. circle, star, etc.) and line color correspond to outlier magnitude. Line style (e.g. solid, dashed, etc.) correspond to CRT window length.

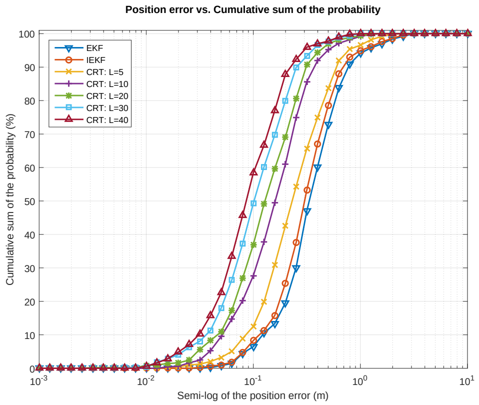

Fig. 7. Position error cumulative distributions for each LSS algorithm. Based on synthetic data results, the value $\lambda_\rho = 1.00$ was selected for all curves to guarantee minimum $P_{CD} > 50\%$.

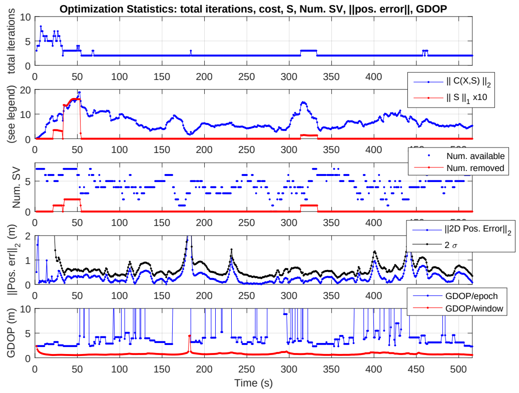

Fig. 8. Optimization Statistics for experimental data trajectory using CRT-LSS with $L = 20$ and $\lambda_\rho = 1.00$, including: total number of iterations, final $\|\mathcal{C}(\hat{\mathbf{X}}, \hat{\mathbf{S}})\|_2$, final $\|\hat{\mathbf{S}}\|_1$, $\|\mathbf{b}\|_2$, and final $\|\delta\hat{\mathbf{X}}\|_2$.