Learning-Enhanced Adaptive Robust GNSS Navigation in Challenging Environments

Yi Ding[1,2], Paul Chauchat[3], Gaël Pages[2], and Philippe Asseman[1]

***Abstract*—Global Navigation Satellite System (GNSS) is the widely used technology when it comes to outdoor positioning. But it has severe limitations with regard to safety-critical applications involving unmanned autonomous systems. Namely, the positioning performance degrades in harsh propagation environment such as urban canyons. In this paper we propose a new algorithm for GNSS navigation in challenging environments based on robust statistics. M-estimators showed promising results in this context, but are limited by some fixed hyper-parameters. Our main idea is to adapt this parameter, for the Huber cost function, to the current environment in a data-driven manner. Doing so, we also present a simple yet efficient way of learning with satellite data, whose number may vary over time. Focusing the learning problem on a single parameter enables to efficiently learn with a lightweight neural network. The generalization capability and the positioning performance of the proposed method are evaluated in multiple contexts scenarios (open-sky, trees, urban and urban canyon), with two distinct GNSS receivers, and in an airplane ground inspection scenario. The maximum positioning error is reduced by up to 68% with respect to M-estimators.**

***Index Terms*—Localization, Probability and Statistical Methods, Machine Learning for Robot Control.**

I. Introduction

GLOBAL Navigation Satellite System (GNSS) techniques play a decisive role in outdoor navigation. However, it faces a number of critical shortcomings to meet the requirements of new applications, such as autonomous vehicles which evolve in complex environments. Indeed, the presence of multipath or non-line-of-sight (NLOS) measurements can produce severe outliers, which strongly affect the accuracy of standard solutions. Various methods have been proposed to try to tackle this problem [1]–[4].

Robust statistics were first developed for regression problems with outliers [5]–[7]. Since then, robust estimators have been widely applied to various domains, including signal processing [8], [9], and in particular GNSS [10]. Recently, the application of M-estimators has been proved to be much useful to mitigate the impact of signal reflections, working directly on the pseudoranges, inside the extended Kalman filter (EKF) framework [11]–[13]. However, the main drawback is that M-estimators rely on parameters, which encode a trade-off between efficiency and robustness, and are usually fixed. In this work, we propose a data-driven approach to adapt such parameter to the dynamic navigation environment.

In parallel to robust estimation, various machine learning (ML) methods were proposed to detect multipath or to estimate measurement uncertainty, based on support vector machine [14], decision trees [15], or neural networks [16], [17]. The use of ML at the receiver level has also been investigated [18], [19]. All of these studies relate to the GNSS signal. If working at this level is more informative, it nevertheless requires changing the architecture of the receiver itself, which is a complex and time-consuming task. Therefore we focus on GNSS observations (i.e. pseudoranges, Doppler frequency) and other GNSS indicators e.g. the number of visible satellites, constellation geometry.

In the observation domain, ML was used to learn specific three-dimensional corrections to the state estimation, but this highlighted a crucial challenge of this application: the number of visible satellites changes with time, thus making it hard to design an ML application taking them as inputs. Two solutions were proposed: considering filtering estimates as a proxy of the satellite information [20], or using set transformers which are agnostic to changing input dimension [21]. The former degrades the information brought by the satellites, while the latter involves heavy computations.

To overcome these limitations, we propose in this work a data-driven approach, in the observation domain, based on two key ideas, in order to both simplify the learning problem at hand, and improving its generalization potential.

- Only a single parameter is learned, the hyper-parameter related to the M-estimator, here using the Huber loss function.
- The satellite data is preprocessed with a standard statistical tool: the empirical probability density function (PDF) of the residuals is considered, thus removing the dependence to both the number and order of the satellites.

This learning-enhanced adaptive robust methodology is validated on real-world data from a wheeled robot evolving in different environments: multiple contexts on the ISAE-SUPAERO campus (open-sky, trees, urban and urban canyon), and in an aircraft ground inspection scenario. Tests include different hardware and geometries of the GNSS constellation.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Airbus SAS and ANRT, France (PhD Cifre number: 2019/1073).

[1]Yi Ding and Philippe Asseman are with Department of Aircraft Operations, Airbus SAS, 31300 Toulouse, France {yi.y.ding, philippe.asseman}@airbus.com

[2]Yi Ding and Gaël Pages are with DEOS department, ISAE-SUPAERO (Institut Supérieur de l'Aéronautique et de l'Espace), University of Toulouse, 31055 Toulouse, France {yi.ding, gael.pages}@isae-supaero.fr

[3]Paul Chauchat is with Department of Automatic Control, IETR, CentraleSupélec Rennes, 35510 Rennes-Sévigné, France paul.chauchat@centralesupelec.fr

The rest of the paper is organized as follows: Section II recalls robust statistics-based GNSS navigation. Section III introduces the proposed LEAR-EKF, and in particular the considered learning problem. Then, experimental validation is presented in Section IV for all considered navigation scenarios. Section V concludes the paper.

II. Robust Statistics-based GNSS Navigation

A. Considered Models for GNSS Navigation

In the context of GNSS navigation, the state vector is commonly defined as $\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_{u,k}^\top & \mathbf{v}_{u,k}^\top & b_{u,k} & d_{u,k} \end{bmatrix}^\top$, where $\mathbf{p}_{u,k} = \begin{bmatrix} x_{u,k} & y_{u,k} & z_{u,k} \end{bmatrix}^\top$ is the user's position, $\mathbf{v}_{u,k} = \begin{bmatrix} \dot{x}_{u,k} & \dot{y}_{u,k} & \dot{z}_{u,k} \end{bmatrix}^\top$ its velocity, and $b_{u,k}$ and $d_{u,k}$ are referred as the receiver's clock bias and clock drift, respectively [22, p.473], [23, p. 264].

Assuming a single-frequency GNSS receiver in nominal conditions, i.e. without multipaths, we can measure the pseudorange, $\rho_{j,k}$, and the pseudorange rate, $\dot{\rho}_{j,k}$, for the j^{th} satellite at a given instant k, which can be modeled by

$$\underbrace{\begin{bmatrix} \boldsymbol{\rho}_k \\ \dot{\boldsymbol{\rho}}_k \end{bmatrix}}_{\mathbf{z}_k} = \underbrace{\begin{bmatrix} \|\mathbf{p}_{u,k} - \mathbf{p}_{j,k}\|_2 + b_{u,k} + \epsilon_{j,k} \\ \mathbf{e}_{j,k}^\top(\mathbf{v}_{u,k} - \mathbf{v}_{j,k}) + d_{u,k} + \dot{\epsilon}_{j,k} \end{bmatrix}}_{\mathbf{h}_k(\mathbf{x}_k)} + \underbrace{\begin{bmatrix} \mathbf{n}_{\rho,k} \\ \mathbf{n}_{\dot{\rho},k} \end{bmatrix}}_{\mathbf{n}_k}, \tag{1}$$

where $\mathbf{p}_{j,k}$ and $\mathbf{v}_{j,k}$ are the j^{th} satellite's position and velocity vectors respectively; $\mathbf{e}_{j,k}$ is the line-of-sight (LOS) unit vector from satellite j to the receiver; $\epsilon_{j,k}$ and $\dot{\epsilon}_{j,k}$ represent the other elements which are known from broadcast navigation message, i.e. satellite clock bias, clock drift, relativistic bias, instrumental group delay as well as ionospheric and tropospheric propagation delays; measurement noises are denoted by $n_{\rho,k}$ and $n_{\dot{\rho},k}$.

A robot with low dynamics is considered in this work, thus a constant velocity process model is adopted

$$\underbrace{\left[\begin{array}{c} \mathbf{p}_{u,k} \\ \mathbf{v}_{u,k} \\ \hline b_{u,k} \\ d_{u,k} \end{array}\right]}_{\mathbf{x}_k} = \underbrace{\left[\begin{array}{cc|cc} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}_{6\times2}$}} \\ \mathbf{0}_3 & \mathbf{I}_3 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}_{2\times6}$}} & 1 & 0 \\ & & 0 & \Delta t \end{array}\right]}_{\mathbf{F}_{k-1}} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}. \tag{2}$$

The process noise matrix is given by

$$\mathbf{Q} = \sigma_a^2 \mathbf{G}\mathbf{I}_3\mathbf{G}^\top, \tag{3}$$

where σ_a^2 is the acceleration noise variance, $\mathbf{I}_3$ the 3×3 identity matrix, $\mathbf{G} = \begin{bmatrix} \Delta t^2/2 \cdot \mathbf{I}_3 \\ \Delta t \cdot \mathbf{I}_3 \end{bmatrix}$, Δt being the sampling period.

If $\mathbf{w}_{k-1}$ and $\mathbf{n}_k$ are distributed as Gaussians, then the well-known EKF provides an accurate estimate of $\mathbf{x}_k$. However, in the case of challenging GNSS environment, this assumption does not hold and faulty measurements will consequently have a non-negligible impact on the estimation if not accounted for. This is why recent studies relied on filters based on robust statistics [11], [13], which we recall thereafter.

B. Robust M-Estimation

Many estimation problems, such as linear regression, are classically solved by the least-squares (LS) estimator, minimizing the sum of the l_2-norm of residuals $(r_i)_{1\leq i\leq n}$

$$\hat{\mathbf{x}}_{\text{LS}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\frac{r_i(\mathbf{x})}{\sigma_i}\right)^2 = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\overline{r}_i(\mathbf{x})\right)^2, \tag{4}$$

where σ_i is a known or previously estimated scale parameter of the corresponding observation, and $\overline{r}_i(\mathbf{x}) = r_i(\mathbf{x})/\sigma_i$. The LS estimator is optimal under Gaussian assumption (and yields the standard EKF [24]). However, it loses its efficiency when the normality assumption is not satisfied, which is generally the case when dealing with outliers in the measurements. The key idea behind M-estimation is to generalize the maximum likelihood estimator by minimizing the sum of a general loss function $\rho(\cdot)$

$$\hat{\mathbf{x}}_{\text{M}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\overline{r}_i(\mathbf{x})\right). \tag{5}$$

It can be shown [25, p.179] that $\hat{\mathbf{x}}_{\text{M}}$ satisfies

$$\sum_{i=1}^{n} w\left(\overline{r}_i(\hat{\mathbf{x}}_{\text{M}})\right) \overline{r}_i(\hat{\mathbf{x}}_{\text{M}}) \frac{\partial \overline{r}_i(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{\text{M}}} = 0, \tag{6}$$

where $w(\cdot)$ is the weight function derived from $\rho(\cdot)$. Equation (6) can be solved by an iterative reweighted least-square (IRLS) process where the weights are given by $w(\overline{r}_i(\hat{\mathbf{x}}))$, updated at each IRLS iteration. In the following we write

$$\hat{\mathbf{x}}_{\text{M}} \overset{\text{IRLS}}{=} \underset{\mathbf{x}}{\operatorname{argmin}} \ \|\overline{\mathbf{r}}(\mathbf{x})\|^2_{\mathbf{W}(\mathbf{x})^{-1}}, \tag{7}$$

where $\mathbf{W}(\mathbf{x})$ is the weighting diagonal matrix defined for each iteration n of the IRLS as (we drop the dependency on $\mathbf{x}^{(n)}$, redundant with (n), for the sake of readability)

$$\mathbf{W}^{(n)} = \operatorname{diag}\left[w\left(\overline{\mathbf{r}}(\mathbf{x}^{(n)})\right)\right]. \tag{8}$$

In the literature there exists several loss functions [26], we focus on the Huber loss in this paper. It is a compromise between the l_1-norm (less sensitive to outliers, but not differentiable at zero) and the l_2-norm (differentiable everywhere, but highly sensitive to outliers) loss functions, behaving quadratically for small residuals and linearly for large ones. The Huber loss is formulated as

$$\rho^{(\alpha)}(r) = \begin{cases} r^2/2 & \text{if } |r| \leq \alpha \\ \alpha|r| - \alpha^2/2 & \text{if } |r| > \alpha \end{cases}, \tag{9}$$

$$w^{(\alpha)}(r) = \min\left\{1, \frac{\alpha}{|r|}\right\}, \tag{10}$$

where $\alpha \in \mathbb{R}^+$ is a hyper-parameter which controls the transition between l_1-norm and l_2-norm loss. Its standard value is 1.345, which guarantees an efficiency of 95% under Gaussian noise [27]. The Huber loss was used in a number of studies related to robust GNSS navigation, e.g. [11], [28] and [13]. Fig. 1 shows how α impacts the weight function.

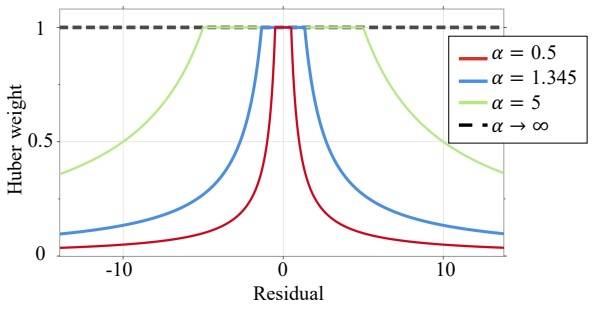

Fig. 1. Huber weight function $w^{(\alpha)}(\cdot)$ for different hyper-parameter values. The limit case is equivalent to a l_2-norm loss (LS).

C. M-Estimation EKF

In order to mitigate the influence of outliers, an M-estimation EKF (M-EKF) was proposed [11], [13]. It does not impact the prediction step, since the process model is linear, given by

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}, \tag{11}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{F}_{k-1}\boldsymbol{\Sigma}_{k-1|k-1}\mathbf{F}_{k-1}^{\top} + \mathbf{Q}_{k-1}. \tag{12}$$

The M-EKF update is based on the M-estimator presented in Section II-B, obtained using an IRLS. As the M-esitmation is defined for LS, the update step is first rewritten in an equivalent augmented batch LS form via linearization [24]

$$\widetilde{\mathbf{z}}_k = \widetilde{\mathbf{H}}_k\mathbf{x}_k + \widetilde{\mathbf{n}}_k, \tag{13}$$

where $\mathbf{H}_k$ is the Jacobian of $\mathbf{h}_k$ about $\hat{\mathbf{x}}_{k|k-1}$

$$\widetilde{\mathbf{z}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \end{bmatrix}, \quad \widetilde{\mathbf{H}}_k = \begin{bmatrix} \mathbf{I} \\ \mathbf{H}_k \end{bmatrix},$$

and the covariance of $\widetilde{\mathbf{n}}_k$ is given by

$$\mathbb{E}\left\{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k^{\top}\right\} = \widetilde{\mathbf{R}}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} = \mathbf{L}_k\mathbf{L}_k^{\top},$$

where $\mathbf{L}_k$ can be obtained by the Cholesky decomposition. Similarly to (4), the regression problem (13) is normalized

$$\underbrace{\mathbf{L}_k^{-1}\widetilde{\mathbf{z}}_k}_{\overline{\mathbf{z}}_k} = \underbrace{\mathbf{L}_k^{-1}\widetilde{\mathbf{H}}_k}_{\overline{\mathbf{H}}_k}\mathbf{x}_k + \underbrace{\mathbf{L}_k^{-1}\widetilde{\mathbf{n}}_k}_{\overline{\mathbf{n}}_k}, \tag{14}$$

so that $\mathbb{E}\left\{\overline{\mathbf{n}}_k\overline{\mathbf{n}}_k^{\top}\right\} = \mathbf{I}$. While the EKF would define the *a posteriori* state through a standard LS in the form of (4), the M-EKF is based on (5), which in turn brings

$$\hat{\mathbf{x}}_{k|k} \overset{\text{IRLS}}{=} \underset{\mathbf{x}_k}{\operatorname{argmin}} \left\|\overline{\mathbf{z}}_k - \overline{\mathbf{H}}_k\mathbf{x}_k\right\|^2_{\mathbf{W}_\alpha(\mathbf{x}_k)^{-1}}, \tag{15}$$

where $\mathbf{W}_\alpha(\mathbf{x}_k)$ is given by (8). Thus for each iteration n

$$\mathbf{W}_\alpha^{(n)} = \operatorname{diag}\left[w^{(\alpha)}\left(\overline{\mathbf{z}}_k - \overline{\mathbf{H}}_k\hat{\mathbf{x}}_k^{(n)}\right)\right]. \tag{16}$$

Equation (15) can thus be solved by the following successive weighted LS estimator

$$\hat{\mathbf{x}}_k^{(n+1)} = \left(\overline{\mathbf{H}}_k{}^{\top}\mathbf{W}_\alpha^{(n)}\overline{\mathbf{H}}_k\right)^{-1}\overline{\mathbf{H}}_k{}^{\top}\mathbf{W}_\alpha^{(n)}\overline{\mathbf{z}}_k, \tag{17}$$

$$\boldsymbol{\Sigma}_k^{(n+1)} = \left(\overline{\mathbf{H}}_k{}^{\top}\mathbf{W}_\alpha^{(n)}\overline{\mathbf{H}}_k\right)^{-1}. \tag{18}$$

The convergence criterion is $\left\|\hat{\mathbf{x}}^{(n+1)} - \hat{\mathbf{x}}^{(n)}\right\| / \left\|\hat{\mathbf{x}}^{(n)}\right\| \leq \delta$, with $\delta > 0$.

A limitation of the Huber loss function in the GNSS navigation framework is that the hyper-parameter α is set as fixed, which may not be suitable for a given environment, making the robust M-EKF perform poorly. Indeed, it could be more appropriate to have a strong aggressive down-weighting in a harsh environment context while fairly tolerant when the environment is just slightly challenging. The adaptive robust concept was inspired with the key idea to automatically tune α allowing the model to adapt the robustness according to the environment in which the robot operates. The achieved performance improvement with such methodology is assessed in section IV through real-world data obtained during measurement campaigns.

III. Machine Learning for Robust GNSS Navigation

A. Methodology: Learning a single parameter

The adaptive robust GNSS navigation method proposed in this work revolves around a main idea: adapting α used in the M-EKF in order to handle changing navigation conditions. As these conditions are hard to model in a dynamic context, we resorted to data-driven methods to infer this parameter. This results in the Learning-Enhanced Adaptive Robust EKF (LEAR-EKF), illustrated in Fig. 2, in which the update parameter α of the M-EKF is inferred at each step from the received information. This means that, compared to (15), the navigation estimate at step k is given by

$$\hat{\mathbf{x}}_{k|k}^{\text{LEAR}} \overset{\text{IRLS}}{=} \underset{\mathbf{x}_k}{\arg\min} \left\|\overline{\mathbf{z}}_k - \overline{\mathbf{H}}_k\mathbf{x}_k\right\|^2_{\mathbf{W}_{\alpha_k}(\mathbf{x}_k)^{-1}}, \tag{19}$$

where α_k is inferred from the available data. Compared to other works trying to directly infer position corrections [20], [21], the rationale of the proposed approach is to get the best from both robust statistics and data-driven methods. As it was already successfully shown in other navigation problems [29], estimating only α simplifies the problem, which should help generalization, while still benefiting from the good properties of the M-EKF.

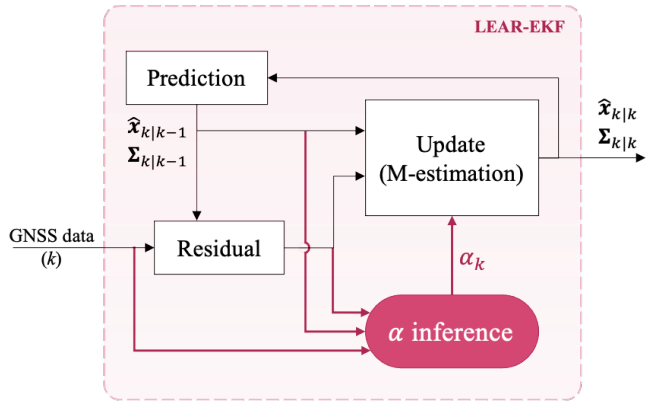

Fig. 2. Diagram of the LEAR-EKF, where the hyper-parameter α is inferred then applied to the Huber loss function within the M-estimation update step.

B. Defining the Learning Problem

This work focuses on supervised methods. However, even if one has access to a reference trajectory, there is no clear reference for the parameter α which should be used at each step. Keeping in mind that it should represent how challenging the environment is, we propose to quantize its value, turning the problem into a classification one. Let $\mathbb{A} = \{\alpha^{(1)}, \dots, \alpha^{(p)}\}$ denote the considered p different values. Considering the role of α in the M-EKF update, a logarithmic scale was considered, so that, for some $\alpha^0, \beta_1, \dots, \beta_p$:

$$\forall 1 \leq m \leq p,\ \alpha^{(m)} = \alpha^0 \exp(\beta_m). \tag{20}$$

The reference value α_k^{GT} is then computed following the process illustrated on Fig. 3. An M-EKF is executed for which, at each step, the best $\alpha \in \mathbb{A}$ is chosen according to

$$\alpha_k^{\text{GT}} = \underset{\alpha \in \mathbb{A}}{\operatorname{argmin}}\, \boldsymbol{\mathcal{J}}(\hat{\mathbf{p}}_{u,k|k,\alpha}, \mathbf{p}_{u,k}^{\text{ref}}, \alpha_{k-1}^{\text{GT}}). \tag{21}$$

The cost function $\boldsymbol{\mathcal{J}}(\cdot)$ focuses on the horizontal accuracy, while ensuring a certain stability of the parameter, so that $\boldsymbol{\mathcal{J}}(\cdot) = \|(\hat{x}_{u,k|k}, \hat{y}_{u,k|k})_\alpha - (x_{u,k}, y_{u,k})_{\text{ref}}\|^2 + \lambda(\log(\alpha/\alpha_{k-1}^{\text{GT}}))^2$, where λ is a regularization parameter and $((\cdot)_{k|k})_\alpha$ represents the solution of (15) for a given α. See Section IV-B for the values used herein.

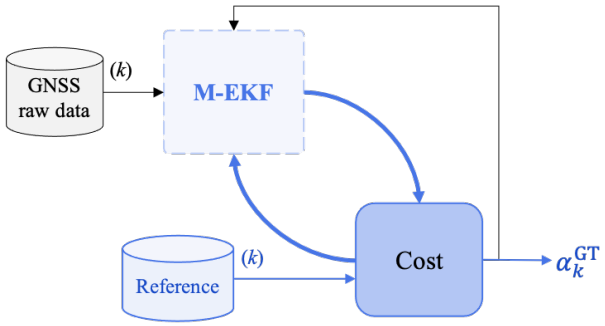

Fig. 3. Process followed to generate the optimal Huber hyper-parameter. A cost computation block to the reference was looped with the M-EKF.

C. Choice and Preprocessing of the Input Data

It was already shown that GNSS standard features can be used to detect degraded conditions [15], [30]. Leveraging learning methods, they can also improve localisation [20], [21]. The latter studies highlight that implementation is not trivial due to the varying structure of the problem: the number of visible satellites varies, and the computations should be invariant to their order. Several solutions were proposed to tackle this problem, such as considering set-valued algorithms [21], which are computationally intensive, or using the output of a filter as a proxy [20], which may destroy information.

In this work, we propose to circumvent the problem by using the empirical PDF of satellite related values. On top of fixing the input size and discarding order issues, this has the advantage to generate inputs in the form of probability vectors, avoiding scale issues [31]. We consider the normalized observation residuals by the corresponding satellite measurement covariance root value, as $\overline{r}_{j,k} = [\mathbf{R}_k]_{j,j}^{-1/2} r_{j,k}$ for the j^{th} satellite. This takes into account the signal quality of each satellite, through their so called carrier-to-noise power density ratios (C/N0), which do not need to be added to the inputs. Their PDF is computed on K manually defined bins, where the first and last ones are unbounded (i.e. gather all values smaller and larger than its threshold respectively). The number of satellites in view (NSV) is considered, as well as the quality of their geometry through the horizontal dilution of precision (HDOP), computed as $\text{HDOP}_k = \sqrt{\text{Tr}[(\mathbf{H}_k^\top \mathbf{H}_k)^{-1}_{1:2,1:2}]}$, where $\text{Tr}[\cdot]$ represents the trace operator. Finally, since α links the predicted state $\hat{\mathbf{x}}_{k|k-1}$ and the observation residuals, we also consider the lower triangular values of the prior covariance $\Sigma_{k|k-1}$ related to the horizontal position. The inputs, along with their domain of definition, are reported in Table I, where the total input dimension is $K+5$. The rationale is as follows: the *empirical* PDF information is completed with the values classically used to *estimate* the residual distribution.

TABLE I
CHOSEN LEARNING INPUTS

Variables	Dimension
Empirical PDF of normalized residuals	K
Prior horizontal position covariance	3
Horizontal dilution of precision (HDOP)	1
Number of satellites in view (NSV)	1

IV. Experimental Validation on a Wheeled Robot

A. Experimental Setup

The proposed LEAR-EKF was evaluated on experimental data acquired by a wheeled inspection robot evolving in challenging and diverse scenarios. For GNSS navigation, the diversity of data comes from three sources: the 3D environment, the hardware, and the satellite configuration (constellation geometry, visible satellites number and IDs). In this regard, measurements from five trajectories were collected: Three trajectories recorded on the ISAE-SUPAERO campus in a multi-contexts scenario recorded at t_0, t_0+30 minutes, t_0+3 hours, denoted "Multi-contexts(A)", "(B)" and "(0)" respectively. This is the red path on Figure 4. One trajectory which loops five times around the same building, marked by "D" on Figure 4, denoted "Loops" and recorded at another date. Finally an aircraft inspection scenario, demonstrated in Fig. 5, with an Airbus A320 parked in an open apron. The aeroplane has a strong reflective curved metal surface with wide wingspan and high tail, which could easily generate multipaths and NLOS conditions. Multi-contexts(B) recorded observations using the MSR, the others relied the u-blox.

Multi-contexts(0) and Loops trajectories were used as training set, the three others formed the test set. This allowed testing generalization against a different constellation geometry (Multi-contexts(A)), a different receiver (Multi-contexts(B)), and a different environment (aircraft).

A positioning reference system was set up consisting of two NovAtel PwrPak7-E1 GNSS+IMU and a dedicated Waypoint offline post-processing software. One of the combined systems was configured as the base station (connected to an antenna

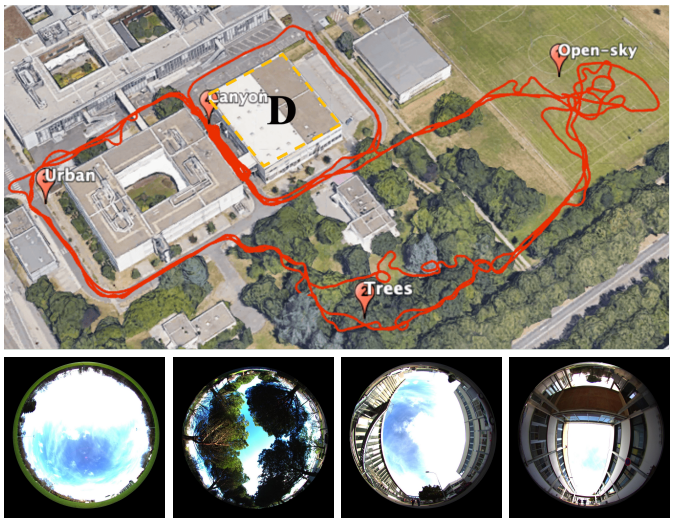

Fig. 4. **Up:** The ISAE-SUPAERO campus on which the measurements were obtained. In red: the multi-contexts scenario. The building marked with "**D**" is the one around which the Loops trajectory was acquired; **Down:** representative fish-eye views of the environments (from the left): open-sky, trees, urban and urban canyon with suspended walkways.

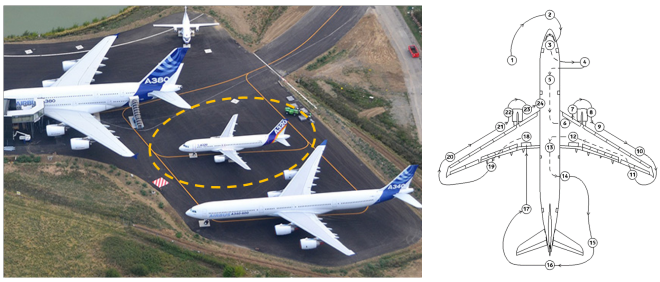

Fig. 5. **Left:** aircraft ground inspection by robot, with a real A320 airplane at the open apron, Aéroscopia, France; **Right:** inspection trajectory with stop points to be performed by the rover.

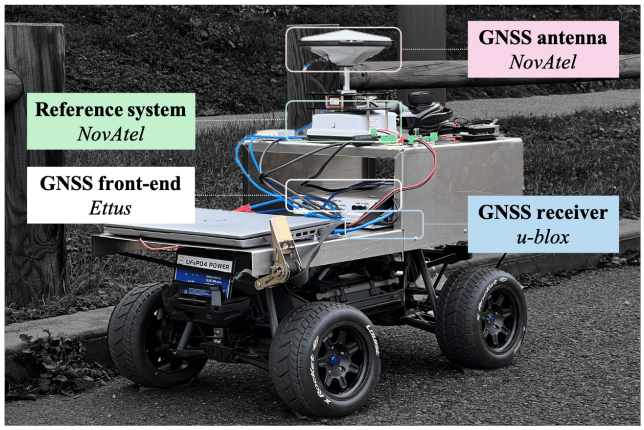

Fig. 6. Rover employed for the measurement campaigns.

mounted on the rooftop of our laboratory and having a well known position) and the second as the mobile station (set up on the rover), thus enabling to perform DGNSS (Differential GNSS) corrections. Both antennas used in the experiments are active and high precision NovAtel antennas. The combination of DGNSS and IMU provides continuously available position, velocity and attitude even through short periods of time when satellite signals are blocked or unavailable, with an accuracy of the solution up to centimeter-level thanks to offline post-processing. The rover, shown in Fig. 6, was equipped with a u-blox M8T commercial receiver, providing GPS and Galileo data (pseudorange, Doppler frequency, C/N0) at 5 Hz and a USRP-X310 (Universal Software Radio Peripheral) from Ettus used as a front-end. The latter provides combined 32-bit in-phase and quadrature samples of the GNSS signal at 4 MHz. These samples are then post-processed by a homemade GNSS software receiver (MSR) [32]. This software receiver is based on a generic bi-constellation (GPS and Galileo) receiver [22] which has been developed in Matlab within our laboratory. The mobile NovAtel PwrPak7-E1, the u-blox M8T and the USRP-X310 are connected to the same antenna through a signal splitter.

B. Implementation Details

Four levels for the value of α were considered around 1.345 , i.e. $p = 4$ in (20), with $\alpha^0 = 1.345$, and $\beta_1, \ldots, \beta_4 = \{-2, -1, 0, 1\}$. In (21), the regularization parameter λ was set to 0.5, meaning that a jump between two consecutive levels of α would require a positioning improvement of about a meter. The PDF of the normalized residuals was computed on $K = 32$ bins, the thirty central ones being equally spaced between -4 and 10.

Data augmentation was carried out on the training set, both on the multi-contexts(0) and the Loops trajectories. A second set of input-output pairs was obtained as follow: at each update, white noise was added to the prior horizontal position and the measurements, which became $\hat{\mathbf{p}}^{\text{2D}}_{u,k|k-1} + \boldsymbol{\varepsilon}^{\mathbf{P}_{\text{2D}}}_k$, and $\mathbf{z}_k + \boldsymbol{\varepsilon}^{\mathbf{z}}_k$ respectively. They were distributed as $\boldsymbol{\varepsilon}^{\mathbf{P}_{\text{2D}}}_k \sim \mathcal{N}(\mathbf{0}, \gamma \boldsymbol{\Sigma}^{\text{2D}}_{k|k-1})$ and $\boldsymbol{\varepsilon}^{\mathbf{z}}_k \sim \mathcal{N}(\mathbf{0}, \gamma \mathbf{R}_k)$, with $\gamma = 0.1^2$. $\boldsymbol{\Sigma}_{k|k-1}$ and $\mathbf{R}_k$ were adjusted accordingly.

A multilayer perceptron (MLP) with 2 hidden layers, of respective size 83 and 248, and hyperbolic tangent (Tanh) hidden layer activations [33] were used as the learning algorithm. The full implementation was done in Matlab.

The acceleration noise variance was set as $\sigma_a^2 = 1.05$ $\text{m}^2 \cdot \text{s}^{-4}$. The variances of $n_{\rho,k}$ and $n_{\dot{\rho},k}$ are based on the sigma-ε model [34], which links the C/N0 measurement to the observation variance, defined by $\sigma^2_{\rho_j} = \sigma^2_\rho \cdot 10^{-(\text{C/N0}_j)/10}$, $\sigma^2_{\dot{\rho}_j} = \sigma^2_{\dot{\rho}} \cdot 10^{-(\text{C/N0}_j)/10}$, where $\sigma_\rho = 200$ m and $\sigma_{\dot{\rho}} = 30$ m/s. The state and covariance estimates were initialized by a LS estimator (weighted by GNSS measurement covariance $\mathbf{R}_k$) after convergence, through a single point positioning procedure in an open-sky environment.

C. Results

1) Classifier performance: Since the LEAR-EKF is fundamentally a recursive algorithm, we considered the following

metric to evaluate the trained classifier: at each update step, the inferred α_k was compared to the one defined by (21) with the current data. The obtained classification performances are reported in Table II, where we can see that the learned model was able to generalize well for the trajectories recorded with the u-blox, with over 74% of correct optimal α class inference. The accuracy is slightly degraded for the MSR receiver, down to 68.9%, but is still satisfactory.

As mentioned at the end of Section III-C, the inputs can be seen a made of two parts, the empirical distribution (the K first inputs of the PDF), and the estimated one (given by the last five inputs). To assess the importance of considering both, we have trained two classifiers, each having only one of the parts as learning inputs. In both cases the classification accuracy dropped under 40%. This thus confirms that the interaction between both parts of the learning inputs defined in Table I plays a key role for this study.

TABLE II
CLASSIFICATION ACCURACY

Type	Dataset	Receiver	# samples	Accuracy
Training validation		u-blox	21548	77.5%
Test	Multi-contexts(A)	u-blox	4719	79.0%
	Multi-contexts(B)	MSR	4386	68.9%
	Aircraft inspection	u-blox	2493	74.2%

2) Positioning performance: We are primarily interested in the output trajectory. Thus the main considerations are: is this classification score enough to ensure positioning as accurate or better than that of the M-EKF? And how does it compare to a commercial off-the-shelf GNSS navigator, such as the u-blox M8T solution? In order to answer these questions, the estimation results of the classical EKF, the standard M-EKF($\alpha^{(3)}$) with fixed $\alpha^{(3)} = \alpha^0 \exp(0) = 1.345$, a more conservative one, denoted, M-EKF($\alpha^{(1)}$) with fixed $\alpha^{(1)} = \alpha^0 \exp(-2) = 0.182$, the u-blox M8T navigation solution and the proposed LEAR-EKF are compared to the reference system. To assess their performances, the two-dimensional root-mean-square error (RMSE) in the North-East frame covering all the trajectory, the horizontal root-square error (RSE) at each step k, and the empirical cumulative distribution function (CDF) of their errors were computed.

The results for the three test trajectories are given respectively in Fig. 7, Fig. 8 and Fig. 9. In each case, the 2D posititon error over time, the inferred α at each step, and the empirical CDF are shown. The overall 2D RMSE, the error at 95% of the CDF and the maximum error are reported in Table III. The standard EKF is strongly degraded in terms of maximum error, showing how challenging these environments are. It is interesting to notice that all the other three methods perform similarly for the first 90% of the CDF, thus showing robustness in most cases. However, the LEAR-EKF stands out in the most difficult parts.

For the multi-contexts(A) trajectory, the LEAR-EKF reduces the maximum 2D RSE by 14.5% and 11.2% compared to the M-EKF($\alpha^{(3)}$) and M-EKF($\alpha^{(1)}$) respectively. It occurred when passing under the suspended walkways between 550-700s (urban canyon). One can also notice that the LEAR-EKF

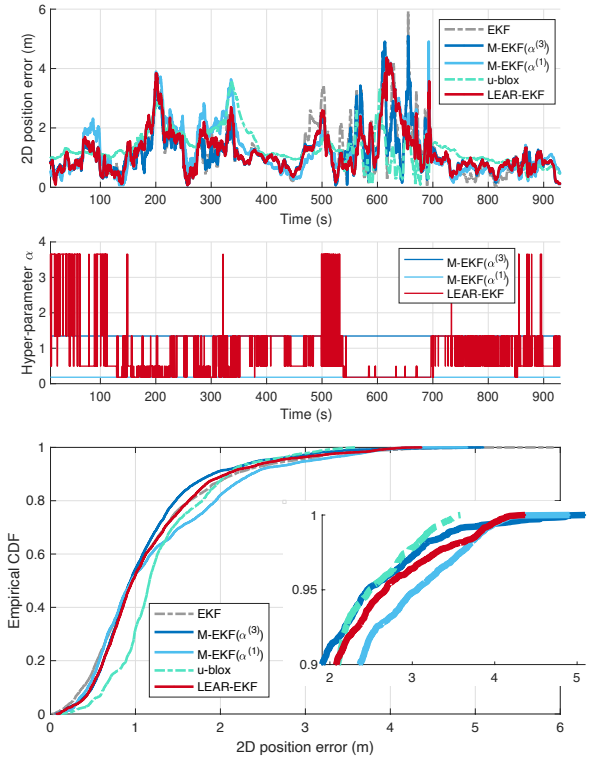

Fig. 7. Performances with the multi-contexts (A) trajectory: **Up**: 2D RSE over time; **Middle**: applied α over time ; **Down**: empirical CDF with zoom. Sections (roughly): before 150s: open-sky, 150-400s: trees, 400-550s: urban, 550-700s: urban canyon, after 700s: canyon.

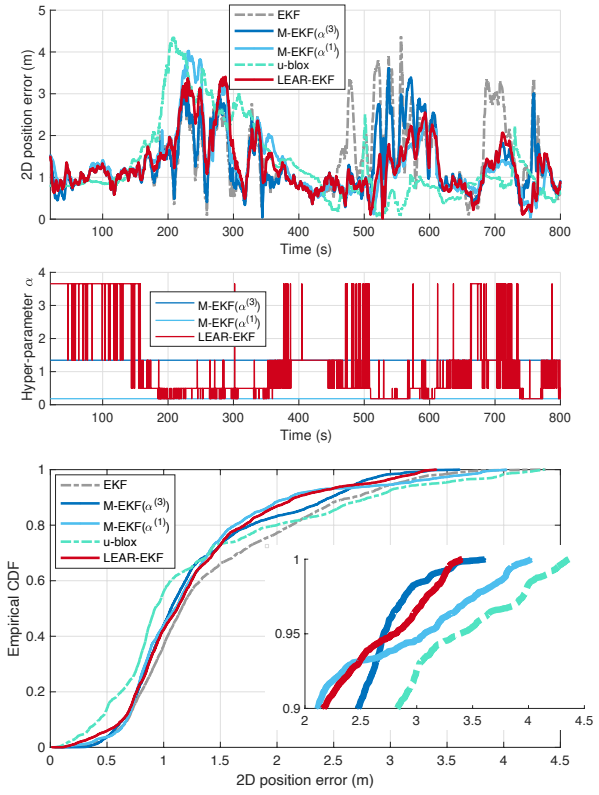

Fig. 8. Performances with the multi-contexts (B) trajectory using raw measurements provided by Ettus receiver: **Up**: 2D RSE over time; **Middle**: applied α over time ; **Down**: empirical CDF with zoom. Sections (roughly): before 150s: open-sky, 150-350s: trees, 350-500s: urban, 500-600s: urban canyon, after 600s: canyon.

TABLE III
POSITIONING PERFORMANCES

	Multi-contexts(A) raw measurements by **u-blox**			**Multi-contexts(B)** raw measurements by **MSR**			**Aircraft inspection** raw measurements by **u-blox**		
	2D RMSE (m)	2D RSE (m)		2D RMSE (m)	2D RSE (m)		2D RMSE (m)	2D RSE (m)	
		95% CDF	Maximum		95% CDF	Maximum		95% CDF	Maximum
EKF	1.40	2.84	5.92	1.65	3.06	4.37	2.89	6.06	10.75
M-EKF($\alpha^{(3)}$)	**1.30**	2.53	5.10	1.48	**2.70**	3.61	2.46	3.97	10.29
M-EKF($\alpha^{(1)}$)	1.51	3.06	4.91	1.48	3.07	4.02	1.53	2.56	3.33
u-blox M8T	1.43	**2.49**	**3.58**	1.61	3.28	4.56	1.85	3.11	4.16
LEAR-EKF	1.37	2.67	4.36	**1.43**	2.80	**3.41**	**1.51**	**2.56**	**3.28**

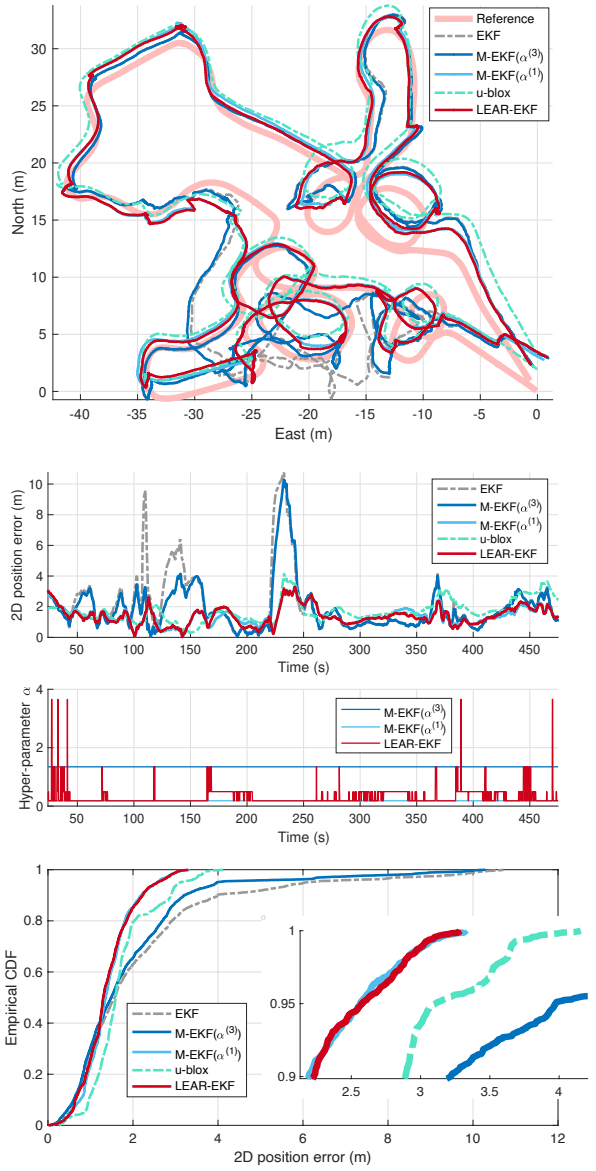

Fig. 9. Performances with the aircraft inspection trajectory: **Up:** position estimates ; **Second:** 2D RSE over time ; **Third:** applied α over time ; **Down:** empirical CDF with zoom. The starting and finishing points are both 5-6 meters in front of the nose of the airplane. The section of 230-250s corresponds to the inspection point totally under the right wing, between the main landing gear and the engine nacelle.

alleviates the loss of efficiency of the M-EKF($\alpha^{(1)}$) between 50-100s (open-sky) and 250-350s(trees). We can notice that, during challenging sections, where the M-EKF($\alpha^{(3)}$) shows strong errors, weaker values of α_k are preferred which appropriately penalize the outliers, and larger ones are inferred in nicer environments. The u-blox M8T solution also gives comparable performances.

For the multi-contexts(B) trajectory with raw measurements provided by the MSR, similar results are observed. The LEAR-EKF reduces the maximum 2D error by 5.5%, 15.2% and 25.2% compared to the M-EKF($\alpha^{(3)}$), M-EKF($\alpha^{(1)}$) and u-blox M8T solution. This happens between 500-600s (urban canyon), where small α_k are inferred. Meanwhile, between 200-250s (trees), the M-EKF($\alpha^{(1)}$) performs badly, while the LEAR-EKF chooses intermediate values for α_k and alleviates this breakdown. This shows the generalization capability of the learned hyper-parameter adapter with respect to the GNSS receiver and satellite configuration.

The aircraft inspection trajectory is a more challenging scenario with entirely different environment and satellite configuration. The LEAR-EKF mitigates the breakdowns of the M-EKF($\alpha^{(3)}$) by enforcing small α_k values most of the time, thus having the same performances as the M-EKF($\alpha^{(1)}$). The maximum 2D error is reduced by 68% and 21% when compared to the M-EKF($\alpha^{(3)}$) and the u-blox M8T solution respectively. Thus, the LEAR-EKF is able to generalize to another 3D environment.

Finally, the proposed LEAR-EKF proved to be a more robust positioning solution in various situations, in different environments, with different satellites configurations, and using different hardware.

V. CONCLUSION

This paper proposes to enhance robust statistics-based estimators for GNSS navigation with data-driven methods to better handle challenging and changing environments. The introduced LEAR-EKF infers the hyper-parameter of the Huber loss function from the available satellite data, whatever their number, yielding a more conservative behavior in the presence of many outliers. In real-world scenarios, including multiple contexts (open-sky, trees, urban and canyon) trajectories with different GNSS receivers, and an airplane ground inspection by a robot, the LEAR-EKF performs better or equally well than the M-EKF with different fixed hyper-parameters, and as well as the u-blox M8T commercial GNSS solution. In particular,

it manages to avoid their largest breakdowns. This validates a methodology which is simple and lightweight, the resulting neural network has about twenty-five thousand parameters. In future works, we will explore the extension of the LEAR-EKF to a broader set of problems.

ACKNOWLEDGMENTS

The authors would like to thank the Navigation team of ISAE-SUPAERO for fruitful discussion and work synergies. Sincere appreciation to Aéroscopia museum for the realization of the measurement campaign. The authors also wish to thank Meryem Benammar for insightful discussions.

REFERENCES

[1] J. Breßler, P. Reisdorf, M. Obst, and G. Wanielik, "GNSS Positioning in Non-line-of-Cight Context — A Survey," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1147–1154.

[2] S. Zair, S. Le Hégarat-Mascle, and E. Seignez, "Outlier Detection in GNSS Pseudo-Range/Doppler Measurements for Robust Localization," *Sensors*, vol. 16, no. 4, 2016.

[3] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, 2018.

[4] H. Li, D. Medina, J. Vilà-Valls, and P. Closas, "Robust Variational-Based Kalman Filter for Outlier Rejection With Correlated Measurements," *IEEE Transactions on Signal Processing*, vol. 69, pp. 357–369, 2021.

[5] P. J. Huber *et al.*, "Robust regression: asymptotics, conjectures and Monte Carlo," *Annals of statistics*, vol. 1, no. 5, pp. 799–821, 1973.

[6] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.

[7] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 196.

[8] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak, "Robust statistics in data analysis—A review: Basic concepts," *Chemometrics and intelligent laboratory systems*, vol. 85, no. 2, pp. 203–219, 2007.

[9] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge University Press, 2018.

[10] D. Borio, "Robust signal processing for GNSS," in *2017 European Navigation Conference (ENC)*. IEEE, 2017, pp. 150–158.

[11] D. Medina, H. Li, J. Vilà-Valls, and P. Closas, "Robust Statistics for GNSS Positioning under Harsh Conditions: A Useful Tool?" *Sensors*, vol. 19, no. 24, p. 5402, Dec 2019.

[12] D. Medina, H. Li, J. Vilà-Valls, and P. Closas, "Robust Filtering Techniques for RTK Positioning in Harsh Propagation Environments," *Sensors*, vol. 21, no. 4, p. 1250, 2021.

[13] Y. Ding, G. Pages, P. Asseman, and É. Chaumette, "Robust Tightly Coupled GNSS/INS Experimental Assessment for Autonomous Aircraft Inspection," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3452–3463.

[14] L.-T. Hsu, "GNSS multipath detection using a machine learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[15] R. Sun, G. Wang, W. Zhang, L.-T. Hsu, and W. Y. Ochieng, "A gradient boosting decision tree based GPS signal reception classification algorithm," *Applied Soft Computing*, vol. 86, p. 105942, 2020.

[16] E. Munin, A. Blais, and N. Couellan, "Convolutional Neural Network for Multipath Detection in GNSS Receivers," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, 2020, pp. 1–10.

[17] G. Zhang, P. Xu, H. Xu, and L.-T. Hsu, "Prediction on the Urban GNSS Measurement Uncertainty Based on Deep Learning Networks With Long Short-Term Memory," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20 563–20 577, 2021.

[18] P. Borhani-Darian and P. Closas, "Deep Neural Network Approach to GNSS Signal Acquisition," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1214–1223.

[19] H. Li, P. Borhani Darian, P. Wu, and P. Closas, "Deep Learning of GNSS Signal Correlation," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 09 2020, pp. 2836 – 2847.

[20] L. Du, J. Ji, Z. Pei, and W. Chen, "A Novel Error Correction Approach to Improve Standard Point Positioning of Integrated BDS/GPS," *Sensors*, vol. 20, no. 21, 2020.

[21] A. V. Kanhere, S. Gupta, A. Shetty, and G. X. Gao, "Improving GNSS Positioning using Neural Network-based Corrections," in *32nd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2021*, 2021.

[22] C. Hegarty and E. Kaplan, *Understanding GPS Principles and Applications, Second Edition*. Artech, 2005.

[23] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. Artech House, 2013.

[24] B. Bell and F. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[25] P. J. Huber, *Robust statistics*. John Wiley & Sons, 2004, vol. 523.

[26] J. Fox and S. Weisberg, "Robust regression," *An R and S-Plus companion to applied regression*, vol. 91, 2002.

[27] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

[28] O. G. Crespillo, D. Medina, J. Skaloud, and M. Meurer, "Tightly coupled GNSS/INS integration based on robust M-estimators," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018, pp. 1554–1561.

[29] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU Dead-Reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.

[30] R. Sun, J. Wang, Q. Cheng, Y. Mao, and W. Ochieng, "A new IMU-aided multiple GNSS fault detection and exclusion algorithm for integrated navigation in urban environments," *GPS Solutions*, vol. 25, p. 147, 10 2021.

[31] M. Benammar, E. D. C. Gomes, and P. Piantanida, "CSI-aided Robust Neural-based Decoders," in *2021 11th International Symposium on Topics in Coding (ISTC)*, 2021, pp. 1–5.

[32] B. Priot, A. Dion, G. Beaugendre, and R. Kasaraneni, "Accurate Events Synchronization in a System-on-Chip Navigation Receiver," *International Conference on Localization and GNSS*, 06 2019.

[33] B. Zamanlooy and M. Mirhassani, "Efficient vlsi implementation of neural networks with hyperbolic tangent activation function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 39–48, 2014.

[34] H. Hartinger and F. Brunner, "Variances of GPS Phase Observations: The SIGMA-ε Model," *GPS Solutions*, vol. 2, pp. 35–43, 1999.