Data-Driven Denoising of Accelerometer Signals

Daniel Engelsman and Itzik Klein *Senior Member, IEEE*

***Abstract*—Modern navigation solutions are largely dependent on the performances of the standalone inertial sensors, especially at times when no external sources are available. During these outages, the inertial navigation solution is likely to degrade over time due to instrumental noises sources, particularly when using consumer low-cost inertial sensors. Conventionally, model-based estimation algorithms are employed to reduce noise levels and enhance meaningful information, thus improving the navigation solution directly. However, guaranteeing their optimality often proves to be challenging as sensors performance differ in manufacturing quality, process noise modeling, and calibration precision. In the literature, most inertial denoising models are model-based when recently several data-driven approaches were suggested primarily for gyroscope measurements denoising. Data-driven approaches for accelerometer denoising task are more challenging due to the unknown gravity projection on the accelerometer axes. To fill this gap, we propose several learning-based approaches and compare their performances with prominent denoising algorithms, in terms of pure noise removal, followed by stationary coarse alignment procedure. Based on the benchmarking results, obtained in field experiments, we show that: (i) learning-based models perform better than traditional signal processing filtering; (ii) non-parametric kNN algorithm outperforms all state of the art deep learning models examined in this study; (iii) denoising can be fruitful for pure inertial signal reconstruction, but moreover for navigation-related tasks, as both errors are shown to be reduced up to one order of magnitude. Both our code and dataset are publicly available @ GitHub.**

***Index Terms*—Inertial sensors, MEMS IMU, denoising, deep learning, kNN regression, attitude estimation.**

I. Introduction

INERTIAL Navigation System (INS) is one of the most commonly used navigation systems. It consists of an inertial measurement units (IMU) with three orthogonal gyroscopes and accelerometers to determine the platform position, velocity, and orientation [1], [2]. With recent technological rise of micro-electro-mechanical-systems (MEMS) inertial sensors, their integration became very common in many applications, platforms, and environments (air, ground, sea). Their popularity can be explained due to their small size, high cost-effective, low power consumption, and low-cost prices. However, when external position/velocity update source is not available, the INS navigation solution becomes solely dependent on the inertial sensor performance and its error regime. Integration of noisy measurements propagates into the navigation solution, resulting in a solution drift [3]. This problem worsens when low cost MEMS-IMU are used, due to their characteristic noise and inherent bias [4].

Conventionally, INS filtering does not incorporate pure signal reconstruction. Instead, a fusion mechanism determines the weight given to current state estimates and external measurements, followed by integration into kinematic quantities. But, given extreme nonlinear dynamics, high frequency noises and non-Gaussian distribution of random noise variables, the filtering process is contaminated. To that end, denoisers can be used for pre-filtering, thus covering holes in the model. However, when deterministic noise sources are introduced and a thorough calibration process cannot be taken, performance of conventional denoisers becomes limited. This is where data-driven methods come in handy, as trainable parameters (weights and biases) are capable of extracting the meaningful patterns from the body of observations.

According to our literature review, among all data-driven denoising works, only one work addresses accelerometer denoising, as the rest perform gyroscopes denoising. It is not surprising in light of the instrumental differences, as same physical scenario is projected inherently different over each sensor. For example, while stationary, MEMS gyroscopes (insensitive to Earth's rotation rate) are expected to output a three-dimensional zero vector $\|\boldsymbol{\omega}\| \approx 0$, regardless body orientation (in practice we get sensor noise). Contrarily, MEMS accelerometers sense Earth gravitation depending on a given orientation, such that their axes being regularly subjected to a larger norm $\|\boldsymbol{f}\| \approx$ g. The constant presence of non-zero amplitudes, wider dynamic ranges and larger noise densities [5], are ultimately reflected in the scale and variability of the samples. Additionally, these differences also propagate into the calibration process. While gyroscopes calibration is self-contained as output offset can be simply subtracted from zero, accelerometers require auxiliary attitude information to ensure parallelism to the gravity axis. When data-driven models are required to generalize over these patterns, distributions of specific force measurements exhibit significantly higher levels of variance and bias, making accelerometers denoising much more challenging, thus still unexplored.

To fill this gap, we propose several data-driven models which successfully manage to denoise accelerometer signals, improving dramatically, not only its signal-to-noise ratio, but also subsequent navigation tasks. The main contributions of this paper can be summarized as follows:

1) An exhaustive literature review of MEMS-IMU denoising techniques, with respect to sensors and dynamics.
2) A bi-directional implementation of a recurrent neural network architecture.
3) A naive yet competitive implementation of a k-nearest neighbor algorithm.
4) A navigation-related metric, demonstrating how denoising improves stationary coarse alignment (SCA).

D. Engelsman is with the Hatter Department of Marine Technologies, University of Haifa, Israel.

I. Klein is with the Hatter Department of Marine Technologies, University of Haifa, Israel (e-mail: kitzik@univ.haifa.ac.il).

The rest of the paper is organized as follows: Section II gives an up-to-date literature review of MEMS-IMU denoising methods. Section III describes auxiliary methods used to assess our strategy and section IV presents our proposed solution. Section V elaborates the data preparation process, section VI shows analysis and results and section VII gives conclusions.

II. Literature Review

Study of inertial sensor denoising techniques dates back to the late 90s, and can be inclusively divided into conventional signal processing approaches and to recent learning-based approaches. Followed is an in-depth literature review to conveniently map the problem domain. Some works evaluate their proposed denoisers in terms of reconstruction of the original IMU output, while others examine its contribution as a prefilter upon a wider endgame output (e.g. Euler angles, navigation states). Table I presents conventional signal processing (SP-based) approaches whose analysis and synthesis is based on the signal structure and nature, enabling to detect components of interest.

TABLE I
Summary of SP-based denoisers

Algorithm	Ref.	Gyro	Acc.	Stationary	Dynamic
Moving average	[6]	✓		✓	✓
ARMA	[7]–[11]	✓		✓	
	[12]–[14]	✓	✓	✓	
	[15]	✓	✓		✓
EMD	[16]	✓	✓	✓	✓
	[17], [18]	✓		✓	
	[19]–[21]	✓		✓	✓
Fuzzy logic	[22], [23]	✓	✓	✓	✓
Savitsky-Golay	[24]		✓	✓	
	[25], [26]	✓	✓	✓	✓
	[27]	✓		✓	✓
Wavelets	[28], [29]	✓	✓		✓
	[30]	✓	✓	✓	
	[31]–[33]	✓		✓	
	[34], [35]	✓			✓
	[36], [37]	✓	✓		✓

Moving average (MA) techniques can be used as efficient smoothing filter, based on errors (residuals) from previous forecasts [6]. Other works elaborated this by combining a weighted regression term over the lagged values, namely auto-regressive moving-average (ARMA) [7]–[15]. Empirical mode decomposition (EMD) is a time frequency analysis which decomposes multicomponent signals into a finite number of Intrinsic Mode Functions (namely, its building blocks). This way, local characteristic time scale is emphasized such that non-stationary and nonlinear signals can be robustly handled [16]–[21]. In fuzzy logic, quantified statements are used to determine objects level of membership in values ranging from zero to one, instead of true or false. During inference stage, the formulated logic is used to map elements from a given input space to output space, performing well with high levels of uncertainties in nonlinear dynamics [22], [23]. Savitsky-Golay is a smoothing filter that fits an optimal local curve over a moving window size, using low-degree polynomial regression [24]–[27]. And the wavelet-based algorithms, currently most popular for MEMS-IMU denoising, uses a window function which is scaled and shifted in the signal time axis, projecting it into a set of basis functions named wavelets [28]–[38]. Table II presents a group of learning-based algorithms, which gained much popularity in recent years in a growing number of fields, including inertial sensors and autonomous navigation. Their advantage lies in their ability to identify complex patterns by learning high-level features of the data, thus diminishing the need in domain expertise.

TABLE II
Summary of learning-based denoisers

Algorithm	Ref.	Gyro	Acc.	Stationary	Dynamic
Linear regression	[39]	✓	✓	✓	✓
CNN	[40]	✓		✓	✓
RNN	[41]–[43]	✓		✓	
LSTM	[44], [45]	✓		✓	
	[46]	✓		✓	✓
GRU	[47]	✓		✓	

Gonzalez proposed a multiple linear regression (MLR) model which uses several vector-valued variables to predict the outcome of a dependent variable [39]. Bossard proposed a convolutional neural network (CNN) which computes gyro corrections for the undesirable noise, before integrated into orientation increments [40]. The rest of the references utilized a common extension to feedforward neural networks called recurrent neural networks (RNN), which were designed to handle variable-length sequential signals. Unlike CNN, which excels at finding spatial relations over grid-like topology, their RNN extension has a feedback element which enables forward and backwards connections such that complex dynamic relationships over distant time steps are better captured [41]–[43]. However, despite its ability to process temporal information of any length, it is limited with long-term dependencies due to the exponential decay of the loss function, namely vanishing gradient problem. To that end, an improved versions were introduced, where additional control units (gates) were added to allow better flow of the gradient and to maintain memory over long time periods, namely long short term memory (LSTM) [44]–[46] and gated recurrent unit (GRU) [47].

As already discussed above, it can be seen that the majority of the works (75%) address gyroscopes denoising, especially in the learning-based approaches.

III. Problem Formulation

After describing the big picture, this section presents two functional methods used for benchmarking (*A.*) and evaluating (*B.*) the validity of our proposed approaches.

A. Signal Processing Denoising Approaches

Conventional SP-based denoisers perform noise reduction by either spatial smoothing, local regression, or by imposing spectral constraints to filter out unwanted frequencies.

Following are three reference methods, used for comparison with the data-driven approaches. Their optimality was obtained by brute-force search over the training set.

1) Moving Average (MA): suppresses unstable signal noises by averaging measurements inside a rolling window [6]

$$\hat{\boldsymbol{x}}_{\boldsymbol{MA},i} = \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{x}_{i+t-T} \quad \forall\ i \geq T \tag{1}$$

where $\boldsymbol{x}_i$ is a noisy sample and T is the window size.

2) Savitsky-Golay (SG): allows denoising by deriving observations directly from time domain, thus avoiding spectral decomposition [24]–[27]. By fitting successive sets of adjacent points with a low-degree polynomial, followed by least-squares regression, local noise is smoothed out. Noisy signal $\boldsymbol{x}_i$ is replaced with a set of m convolution coefficients C_t,

$$\hat{\boldsymbol{x}}_{\boldsymbol{SG},i} = \sum_{t=-t_s}^{t_s} C_t \boldsymbol{x}_{i+t} \ , \quad \tfrac{m+1}{2} \leq t \leq t_f - \tfrac{m-1}{2} \tag{2}$$

where t_f is the duration of data point x_t and $t_s = \frac{m-1}{2}$ denotes the window margins.

3) Discrete wavelet transform (DWT): the most common denoising technique, where input signal $\boldsymbol{x}_i$ is represented in both time and frequency domains, by decomposing it into a set of basis functions [28]–[37]. Here, a Daubechies (db4) mother wavelet function (ψ) is used with different scaling ($a = 2$) and shifting (b) parameters, providing a progressively finer outputs, given by coefficients matrix Ψ

$$\Psi_i[b, a^j] = \sum_{t=0}^{N-1} \boldsymbol{x}_i[t] \frac{1}{\sqrt{a^j}} \psi_j\left(\frac{t-b}{a^j}\right) \tag{3}$$

Since small valued coefficients are dominated by noise, hard thresholding T_{Hard} is used to remove them, thus preserving only meaningful information. Then, an inverse transform is applied on the thresholded wavelet coefficients, to reconstruct the denoised matrix back to time domain signal as given by

$$\hat{\boldsymbol{x}}_{\boldsymbol{DWT},i} = \Big(\mathrm{T}_{\text{Hard}}(\Psi_i)\Big)^{-1} \tag{4}$$

B. Inertial Navigation Coarse alignment

In addition to analyzing our proposed approach denoising capabilities on the accelerometer signals, we examine its influence on a stationary coarse alignment (SCA) procedure. To that end, this section gives a brief introduction to the coarse alignment theory. The attitude (roll and pitch angles) can be determined using accelerometer measurements. The orientation of body coordinates frame with respect to navigation frame is given by a transformation matrix, represented by a product of three successive rotations about the z-y-x axes, corresponding to Euler angels: yaw (ψ), pitch (θ) and roll (ϕ).

$$\mathbf{T}_b^n = \begin{bmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi \\ \mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi \\ -\mathrm{s}_\theta & \mathrm{s}_\phi \mathrm{c}_\theta & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix} \tag{5}$$

In stationary conditions, accelerations in navigation frame are equal to zero ($\dot{\mathbf{v}}^n = \mathbf{0}$), thus only the gravity vector is projected upon the accelerometer axes [3]

$$\boldsymbol{f}_{ib}^b = \begin{bmatrix} f_{ib,x}^b \\ f_{ib,y}^b \\ f_{ib,z}^b \end{bmatrix} = -\mathbf{T}_n^b\, \mathbf{g}^n = \begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ -c_\phi c_\theta \end{bmatrix} \mathrm{g} \tag{6}$$

Using analytical coarse alignment, leveling can be computed by the axial components of the specific force vector $\boldsymbol{f}_{ib}^b$ [4]

$$\phi = \arctan_2\Big(-f_{ib,y}^b\,,\, -f_{ib,z}^b\Big) \tag{7}$$

$$\theta = \arctan\left(\frac{-f_{ib,x}^b}{\sqrt{f_{ib,y}^{b\ 2} + f_{ib,z}^{b\ 2}}}\right) \tag{8}$$

The accuracy of the SCA process affects the whole INS performance, especially during pure inertial navigation. To increase the degraded signal-to-noise ratio (SNR), instrumental errors can be reduced in proportion to a square root factor, by either averaging over n identical sensors ($1/\sqrt{n}$), or by averaging over T time steps ($1/\sqrt{T}$). Thereby, effective denoising can reduce averaging times of the SCA procedure.

IV. Proposed Approach

We propose implementing and modifying several learning algorithms for the accelerometer denoising problem, which unlike conventional, model-based, denoising filters, are independent in any analytic error model. Instead, they learn to map noisy readings into their corresponding accurate measurements. Figure 1 presents the proposed approach, where noisy measurements from commercial grade inertial sensors are input to a learning model for the denoising task. The model then outputs denoised accelerometer readings, which are used to calculate an approximation error, with respect to a high-end, accurate inertial sensors readings, as ground-truth (GT). The error is first used to optimize the model during the training phase. Then, during testing phase (VI), the well-trained models are evaluated over unseen noisy samples.

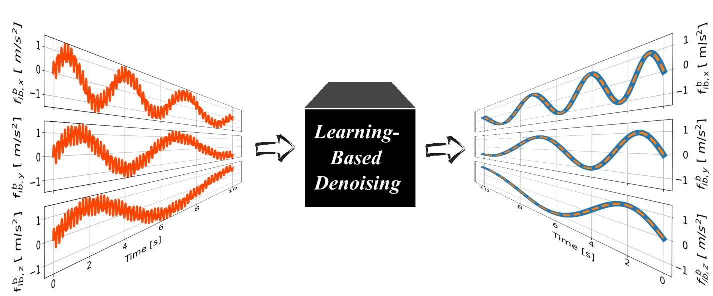

Fig. 1. Denoising mapping function $f : \boldsymbol{x} \rightarrow \hat{\boldsymbol{x}}$

Unlike SP-based methods, here extracted patterns are associated with target functions, enabling signal denoising by generalization. Similarly to human processing, feedforward neural networks (FNN) excel at pattern recognition, as their layered structure perform hierarchical representations, allowing extraction of spatial features [48]. However given sequential relations, their node-specific weights eventually fail to learn, as lengthy inputs and complex ordering impose an exponentially growing number of parameters. To that end, a recurrent feedback mechanism was proposed [49]

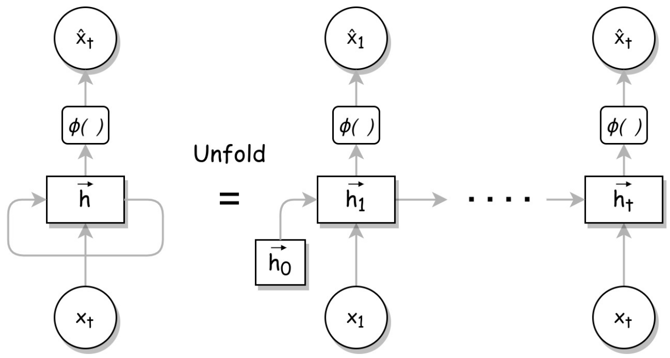

Fig. 2. One-layer recurrent neural network (RNN)

Three dimensional measurements are processed inside an encoding cell called hidden state, using a weight vector of m learnable parameters, such that $\overrightarrow{\boldsymbol{h}} \in \mathbb{R}^{3\times m}$. Unlike FFN, here same weights are shared across the entire sequence, and current inputs are taken from previous outputs. This way, the model avoids spatial memorization and focuses on intertemporal dependencies. Figure 2 illustrates a sequence-to-sequence mapping, consists of single 3D noisy measurements, estimated (denoised) every time step $\hat{x}_t \in \mathbb{R}^3$.
Following, we elaborate on our proposed models: three variations of deep recurrent neural networks (RNN) and one naive machine learning model (kNN).

A. Unidirectional bi-layer LSTM

The architecture of the first model is similarly arranged as shown in Figure 2, but consists of two (bi) stacked layers. The hidden states are an RNN variant called long-short term memory (LSTM) [50], originally developed to handle the decay of the loss function:

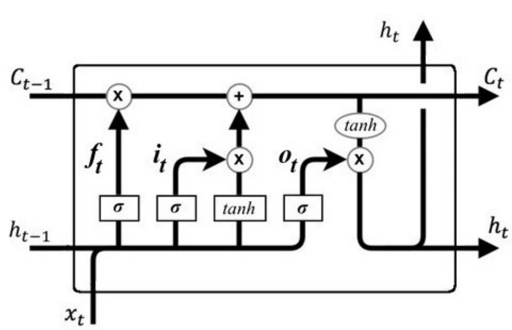

Fig. 3. Hidden state unit of an LSTM

As illustrated in Figure 3, the LSTM hidden state contains three non-linear gates, used to regulate the signal flow. The forget gate $\boldsymbol{f}_t$ decides whether ignoring or adding new (short-term) information to the cell state $\boldsymbol{C}_t$ memory. The input gate $\boldsymbol{i}_t$, is then used to estimate the relevance of current input to past hidden states. This way, the cell state acts as a global memory unit, where distant dependencies (long-term) are maintained across time

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_f[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_f) \tag{9}$$
$$\boldsymbol{i}_t = \sigma(\mathbf{W}_i\,[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_i) \tag{10}$$
$$\tilde{\boldsymbol{C}}_t = \tanh(\mathbf{W}_c[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_c) \tag{11}$$
$$\boldsymbol{C}_t = \boldsymbol{f}_t \otimes \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \otimes \tilde{\boldsymbol{C}}_t \tag{12}$$

Operator $\otimes$ denotes an element-wise multiplication, $\mathbf{W}_j$ and $\mathbf{b}_j$ are the weights and biases respectively, and the sigmoid function scales encoded vectors as follows $\sigma : \mathbb{R}^m \rightarrow (0,1)^m$. Next, the output gate $\boldsymbol{o}_t$ is fused with the updated cell state

$$\boldsymbol{o}_t = \sigma(\mathbf{W}_o\,[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_o) \tag{13}$$

such that the next hidden state is determined as follows:

$$\overrightarrow{\boldsymbol{h}_t} = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{C}_t) \tag{14}$$

Finally, the hidden state signal is taken for state prediction, using a linear activation function ϕ (neuron) which scales it back into a continuous and unbounded range, given by

$$\hat{\boldsymbol{x}}_t = \phi(\overrightarrow{\boldsymbol{h}_t}) = \mathbf{W}_{\hat{y}}^{\intercal}\overrightarrow{\boldsymbol{h}_t} + \mathbf{b}_{\hat{y}} \ \in\ \mathbb{R}^3 \tag{15}$$

B. Bi-directional one-layer RNN

Conventionally, sequential inputs are analyzed along the positive time direction (chronologically), thus limiting the learning with respect to only past states. Figure 4 illustrates a two opposing cells architecture, where future contexts are learned from the opposite time direction (anti-chronological):

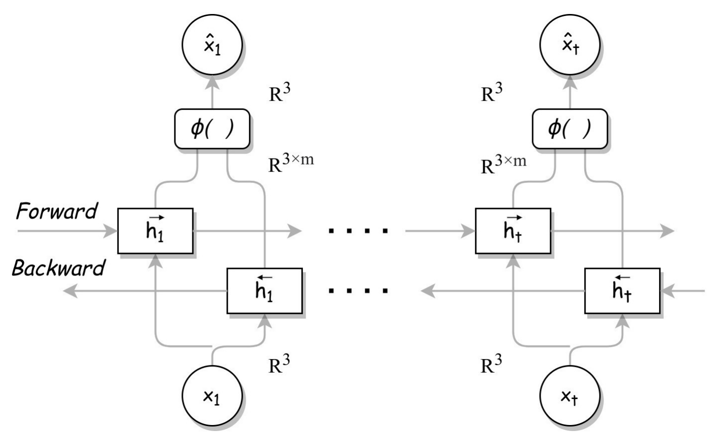

Fig. 4. Bi-directional RNN [51]

Although RNNs lack the cell state concept, their simple parametric relations managed to exhibit competitive results when applied in a shallow (one-layer) arrangement. Figure 5 presents a forward directed hidden unit, at time step t:

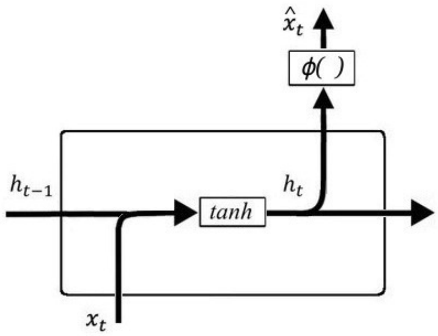

Fig. 5. RNN hidden state and its denoised output

Here, two different hidden layers are allocated to process data both in forward $\overrightarrow{\boldsymbol{h}_t}$ and backward $\overleftarrow{\boldsymbol{h}_t}$ directions

$$\overrightarrow{\boldsymbol{h}_t} = \sigma(\mathbf{W}_{\overrightarrow{h}}[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_{\overrightarrow{h}}) \tag{16}$$

$$\overleftarrow{\boldsymbol{h}_t} = \sigma(\mathbf{W}_{\overleftarrow{h}}[x_t,\ \overleftarrow{h}_{t-1}] + \mathbf{b}_{\overleftarrow{h}}) \tag{17}$$

Then, parameters of the opposing hidden states are stacked $\overline{\boldsymbol{h}}_t$, and fed into the output layer to provide state denoising

$$\overline{\boldsymbol{h}}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t] \ \in\ \mathbb{R}^{3\times 2m} \tag{18}$$

$$\hat{\boldsymbol{x}}_t = \phi(\overline{\boldsymbol{h}}_t) = \mathbf{W}_{\hat{y}}\overline{\boldsymbol{h}}_t + \mathbf{b}_{\hat{y}} \ \in\ \mathbb{R}^3 \tag{19}$$

C. Bi-directional one-layer GRU

The third model is also arranged in a bi-directional form (Figure 4), but utilizes a different gating mechanism called gated recurrent unit (GRU), which merges both input and forget gates into a single update gate [52]. Using only two-gates, it requires shorter training epochs, less computational efforts and is thus more robust to the vanishing gradient problem. Using a sigmoid function, the gates regulate the trade-off between previous hidden states and new input information. Zero output means closed gate, such that historical data cannot pass beyond, and only the new input is emphasized, as presented in Figure 6.

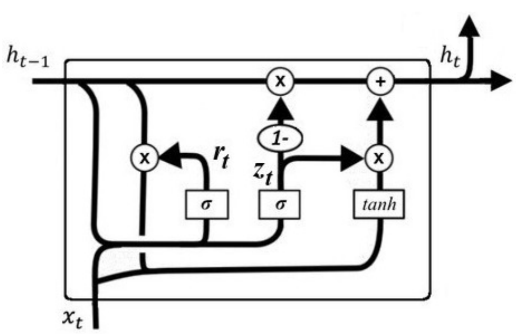

Fig. 6. Hidden state unit of a GRU model

The reset gate $\boldsymbol{r}_t$ controls the dominance of past states with respect to new input, and the update gate z_t specifies how much memory will pass to the next hidden state.

Since both hidden states are symmetrically opposite, only the chronological direction is formulated to simplify notation

$$\boldsymbol{r}_t = \sigma(\mathbf{W}_r[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_r) \tag{20}$$

$$\boldsymbol{z}_t = \sigma(\mathbf{W}_z[x_t,\ \overrightarrow{h}_{t-1}] + \mathbf{b}_z) \tag{21}$$

where the chronological hidden state is defined as

$$\tilde{\boldsymbol{h}}_t = \tanh(\mathbf{W}_{\overrightarrow{h}}[x_t,\ \boldsymbol{r}_t \otimes \overrightarrow{h}_{t-1}] + \mathbf{b}_{\overrightarrow{h}}) \tag{22}$$

$$\overrightarrow{\boldsymbol{h}}_t = (1 - \boldsymbol{z}_t) \otimes \overrightarrow{h}_{t-1} + \boldsymbol{z}_t \otimes \tilde{\boldsymbol{h}}_t \tag{23}$$

Similarly to (19), the opposite cells are horizontally stacked and approximated by the external output layer as

$$\overline{\boldsymbol{h}}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t] \ \in\ \mathbb{R}^{3\times 2m} \tag{24}$$

$$\hat{\boldsymbol{x}}_t = \phi(\overline{\boldsymbol{h}}_t) = \mathbf{W}_{\hat{y}}\overline{\boldsymbol{h}}_t + \mathbf{b}_{\hat{y}} \ \in\ \mathbb{R}^3 \tag{25}$$

Note that the optimality of the above models was obtained heuristically, with respect to hidden state architecture, number of parameters, directionality of signal flow and model depth.

D. k-nearest neighbors algorithm (kNN)

Our last proposed model does not assume priors on the underlying distribution, but is only determined by the number of k nearest neighbors. Let the dataset be a set of n pairs

$$\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{x}_{GT,i})\}_{i=1}^{n} \tag{26}$$

where denoised outputs are determined by the mean GT value of k closest (noisy) samples to the new data point

$$\hat{\boldsymbol{x}}(k) = \frac{1}{k} \sum_{\boldsymbol{x}_i \in \boldsymbol{N}(k)} \boldsymbol{x}_{GT,i} \tag{27}$$

Given r-dimensional data points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, neighborhood $\boldsymbol{N}$ will be consisted of k closest neighbors, as proximity is calculated using an Euclidean distance function

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{j=1}^{r}(x_{1,j} - x_{2,j})^2} \tag{28}$$

To simplify demonstration, Figure 7 illustrates a kNN denoising (k=6), between one-dimensional data points, but applies similarly to higher dimensional vectors as in our experiment

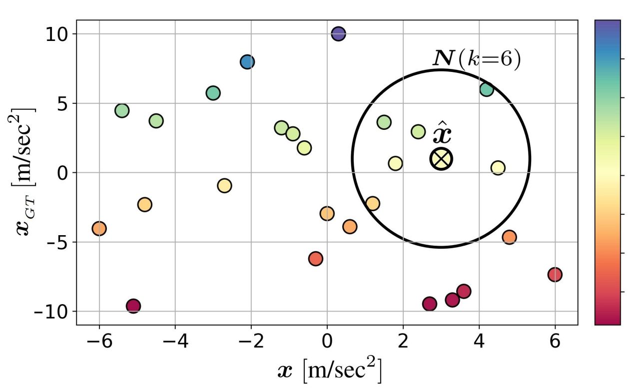

Fig. 7. Non-parametric kNN denoising

Each coordinate in the acceleration plane, refers to a different sensor grade. The horizontal and vertical axes refer to the noisy and GT measurements respectively, where the colorbar indicates only the GT values. Estimating a new data point $\hat{\boldsymbol{x}}$, is obtained by averaging all GT data points inside $\boldsymbol{N}(k)$. In the absence of pre-defined statistical methods to find optimality, an heuristic search brought up the following optimal k

$$k^* = \arg\min_k \|\hat{\boldsymbol{X}}(k) - \boldsymbol{X}_{GT}\|^2 \approx \sqrt{n/5} \tag{29}$$

where $\hat{\boldsymbol{X}}$ is the kNN estimates and $\boldsymbol{X}_{GT}$ is their corresponding GT samples, both taken from the test-set, which is one fifth of the dataset ($n = 90,000$).

E. Loss function

Unlike the kNN optimization, deep learning architectures use a loss function to assess deviations between their denoised predictions and the actual ground truth. The axial error of a single sample containing H time steps is given by

$$\boldsymbol{e}_j = \hat{\boldsymbol{x}}_j - \boldsymbol{x}_{j,GT} \ \in \ \mathbb{R}^H \tag{30}$$

where j denotes an axis index as each sensor measurement consists of three channels - x,y and z. Then, during training, performance is assessed using mean squared error (MSE), such that loss surface is continuously differentiable, and errors can be minimized progressively as a function of the model weights

$$\text{MSE} = \frac{1}{H \times 3} \sum_{j \in \{\text{x,y,z}\}} \sum_{i=1}^{H} (e_{i,j})^2 \tag{31}$$

V. Datasets and Error Measures

To evaluate our methodology, models from both signal processing-based and learning-based approaches are evaluated and compared over simulated and experimental datasets, using unique error measures for the evaluation process.

A. Simulated Dataset

In stationary conditions, the gravity vector is projected into the specific force measurements of the accelerometers. This projection can be represented using Euler angles.

In our simulative setup, Euler angles are drawn within a bounded domain, and using small intervals, a rich combination of data points (orientations) is guaranteed. Using the transformation matrix (5), Euler angles dictate how gravity is projected on the sensor axes (6). Without loss of generality, the yaw angle has no influence on the gravity projection, thus it remains constant ($\psi_{sim.} = 0$), and we address only the roll and pitch angles in the range of

$$-15^\circ \ \leq \phi_{sim.} \leq +15^\circ \tag{32}$$
$$-15^\circ \ \leq \theta_{sim.} \leq +15^\circ \tag{33}$$

Both angles are divided into intervals of 0.1 deg, such that $\frac{15-(-15)}{0.1} = 300$ increments are obtained. Then, the angles are combined together, and the overall number of simulated instances becomes 300^2=90,000 combinations. Notice, that in this study we limit the angles range to $\pm 15^\circ$ to enable model training in fair times on our hardware (Intel i5-9600K CPU @ 3.70 GHz and NVIDIA GTX2080 GPU). Yet, of course, the proposed methods can be applied in any angle range, influencing only the training time. The inference mode (real-time) operation is not influenced by the dataset size and the training time. Next, synthetic noise and error terms are added to each accelerometer channel using the following error terms:

1) Velocity random walk (VRW) - accumulative error due to white noise in the measurement.
2) Bias instability (BI) - stationary stochastic process which act as a low-order Gauss-Markov process.
3) Bias offset (BO) - a constant offset caused by misprojection of the true accelerations.

Figure 8 illustrates the simulation, showing that the constructed dataset contains both simulated GT and measured accelerometer readings. Table III specifies the synthetic additive noise values (Simulation column).

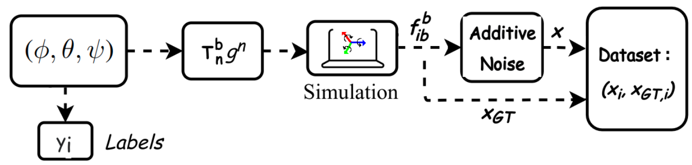

Fig. 8. Simulated dataset generation procedure

B. Field Experiment Dataset

The stationary conditions field experiment dataset is based on a dataset published in [53]. There, a unique device, as shown in Figure 9 was built to align between a Huawei P40 smartphone and an Inertial Lab MRU-P unit [54].

Fig. 9. Unique device aligning between a Huawei P40 smartphone and an Inertial Lab MRU-P unit.

In this setup, the smartphone accelerometer readings are used as unit under test while the MRU's accelerometer readings serve as GT measurements. Both accelerometers provided measurements at 100Hz sampling rate.

Due to practical constraints, smaller amount of raw measurements were taken in much bigger intervals of $\approx 3.0^\circ$ deg

$$-15^\circ \ \leq \phi_{exp.} \leq +15^\circ \tag{34}$$
$$-15^\circ \ \leq \theta_{exp.} \leq +15^\circ \tag{35}$$

The experimental dataset consists of $10^2 = 100$ recordings at different orientations and duration, divided into sub-samples depending on the window size. Unlike the simulated dataset, yaw angles here are non-zeros ($\psi_{exp.} \neq 0$), as they were taken from a wide variety of platform orientations. Table III specifies the sensors noise specs, as stated by manufacturers (Experiment column).

TABLE III
ERROR SOURCES USED IN THE SIMULATION AND EXPERIMENTS

		Simulation		**Experiment**	
Error	Units	Noisy	GT	P40	MRU-P (GT)
VRW	[m/s/$\sqrt{s}$]	0.005	0.00001	0.003	0.00025
BI	[m/s^2]	0.001	0.00001	0.001	0.00005
BO	[m/s^2]	0.05	0.00001	0.067	0.0001

Lastly, to increase variability of the experimental dataset and avoid model overfitting, each sample underwent two consecutive transformations suitable for time-series [55]–[57]: i) Angular augmentation - applying angular transformations over a given orientation, to obtain more different samples and densify the sparse distribution. ii) Noise augmentation - applying stochastic error in form of additive white Gaussian noise. This way, the augmented dataset also contains a total of 90,000 samples, improving generalizability and noise robustness of the learning-based models.

C. Performance Metrics

During the testing phase, several evaluation functions are used to provide relevant measures of the models performances from the perspective of signal reconstruction.

- Root mean squared error (RMSE) - is defined as the square root of the MSE (31), given by

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{36}$$

- Mean absolute error (MAE) - returns an average magnitude of the residuals (30), calculated over n time steps, disregarding their direction, given by

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{37}$$

- Peak signal-to-noise ratio (PSNR) - returns the ratio between maximum signal value (MAX) and noise levels. However here, PSNR expresses ratio between two amplitudes (root-power quantity), thus a measure of order of magnitude is equal to 20 [dB] difference, given by

$$\text{PSNR} = 20\log_{10}\left(\frac{\text{MAX}}{\text{RMSE}}\right) \tag{38}$$

- Relative absolute error (RAE) - returns a relative measure of the discrepancy between the residuals and their corresponding ground truth mean value μ_{GT}, given by

$$\text{RAE} = \frac{\sum_{i=1}^{n}|e_i|}{\sum_{i=1}^{n}|x_{GT,i} - \mu_{GT}|} \tag{39}$$

For example, Table IV presents several error metrics, where outputs of a denoising model are compared to raw accelerometer measurements, in terms of dissimilarity to the accurate GT measurements. Therefore, the optimization task is $\hat{x} \approx x_{GT}$.

TABLE IV
TOY EXAMPLE: EVALUATION

	RMSE [m/s^2]	MAE [m/s^2]	PSNR [dB]	RAE [%]
Model	0.0251	0.0198	51.8317	0.4436
Noisy	0.0626	0.0464	43.8681	1.1096

It can be seen that noise levels are reduced by several aspects and peak signal-to-noise ratio is improved by 8 [dB].

VI. ANALYSIS AND RESULTS

This section presents a comparative performance analysis between all models introduced above: signal processing-based (III-A) and our proposed learning-based (IV). Both simulated and the experimental datasets are assessed, first in terms of pure signal reconstruction, followed by measuring its influence on the stationary coarse alignment procedure.

A. Simulation Assessment

After the simulated dataset is generated, models are trained on the training set before comparing performances. Then, the simulated test-set utilizes to examine the models capability of reducing noise sources, in terms of error minimization. Figure 10 visualizes model inference on a single noisy sample (bubbles), where GT measurements are denoted by solid blue line and models estimates $\hat{x}$ are marked with dashed lines.

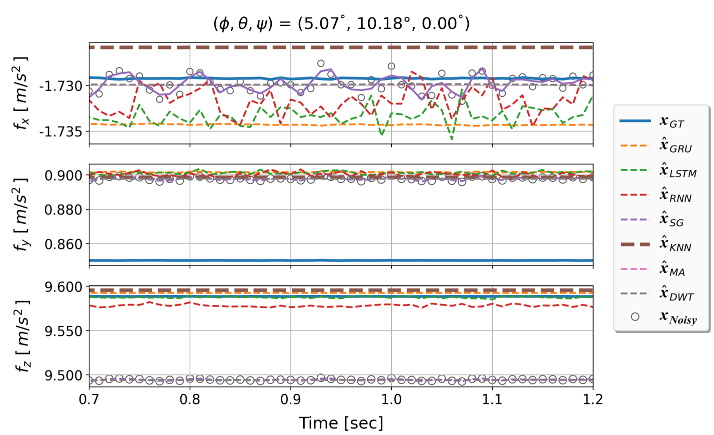

Fig. 10. Reconstruction comparison: Simulation

Since this dataset is simulated, different tradeoffs between deterministic and stochastic noise sources were tested to examine models robustness. In the absence of a calibration process or knowledge about inherent bias, SP-based performances remained bounded by the sensors offset errors. In contrast, the learning-based estimators are optimized during training phase, capable of compensating wide range of error sources.

Table V summarizes the models denoising performances across the simulated training set, providing dissimilarity measures between reconstructed signals and the GT observations.

TABLE V
RECONSTRUCTION COMPARISON: SIMULATION

Model	RMSE [m/s^2]	MAE [m/s^2]	PSNR [dB]	RAE [%]
kNN	0.02481	0.01912	52.10719	0.44484
GRU	0.02635	0.01950	51.39078	0.46668
LSTM	0.02701	0.01999	50.10601	0.47835
RNN	0.02898	0.02145	46.25170	0.51335
DWT	0.06278	0.04623	43.81552	1.11545
SG	0.06298	0.04678	43.62187	1.11605
MA	0.06344	0.04872	43.42730	1.12842
Noisy	0.06614	0.04913	41.73747	1.18383

Models are ranked according to RMSE results, where learning-based denoisers have a clear advantage over SP-based algorithms. Since the kNN model leads the table, let us define a suppression ratio γ to quantify its ability to minimize the reconstruction error, with respect to raw noisy measurements

$$\gamma_{\text{sim.}} = \frac{\text{RAE}_{\text{kNN}}}{\text{RAE}_{\text{Noisy}}} \approx 37.5\% \tag{40}$$

As seen, the relative absolute error (RAE) of the kNN estimates, cuts dissimilarity error in more than half when compared to noisy measurements. Generalization error is minimized the closer the output reaches the GT, but increases proportionally with respect to sudden noise bursts.

Next, the denoised signals in Figure 10 are computed through the analytic SCA equations (7, 8), to examine whether the reconstructed signals improve the roll and pitch accuracy.

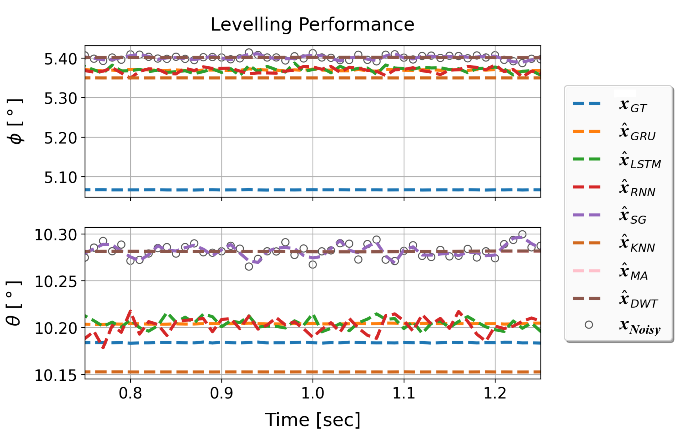

Fig. 11. SCA comparison: Simulation

Figure 11 demonstrates how the computed angles obtained from models outputs, improve the SCA procedure, as computed roll and pitch angles lay closer to the ground-truth. Since the kNN is leading the accuracy, the next definition is used to assess angular error between computed angle and GT angle

$$\varepsilon = \alpha - \alpha_{GT} \ \in \ \mathbb{R} \tag{41}$$

Figure 12 demonstrates the angular errors with respect to GT, as kNN estimated angles (brown) are compared with noisy raw measurements (blue), over the entire simulated test-set.

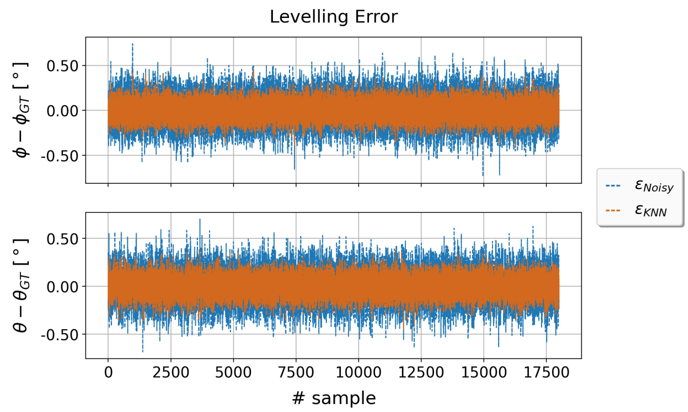

Fig. 12. SCA results of kNN: Simulation

To understand the meaning of these results, Table VI utilizes RMSE as a performance indicator, to summarize the kNN contribution to lowering the overall angular error.

TABLE VI
OVERALL IMPROVEMENT RATE: SIMULATION

	$\varepsilon_{\text{Noisy}}$ [°]	ε_{kNN} [°]	ratio [%]
RMSE(ϕ)	0.18452	0.14895	80.695
RMSE(θ)	0.18147	0.14167	78.068

To conclude, synthetic noises and biases were generated to simulate contaminated measurements and to examine the models capabilities to remove them. Empirical experiments showed that each architecture exhibited different bias-variance tradeoffs. Some estimators performed better using shallow architectures such that variance error (overfitting) was reduced. Others improved when estimated parameters were more noise-sensitive, thus avoiding underfitting due to sensors biases. Over the entire test-set, it was shown that learning-based estimates reduce reconstruction error by up to 37.5%, followed by 20% accuracy improvement of the SCA procedure.

B. Experimental Assessment

Unlike the simulated dataset, error sources in the experimental setup are sensor-specific, and tradeoff between them cannot be modified. Here, noisy samples are given by a consumer-grade smartphone sensor, whose readings are submerged in lower levels of stochastic noise but higher bias levels, as smartphones can be prone to mechanical shocks that impair the orthogonality of the sensor axes.

In contrast, the GT references are taken from a high-end sensor, whose accurate measurements are characterized by significantly lower noise levels [53]. In other words, the generalization task here is to find an approximation function that mimics best the GT sensor.

Similarly to before, Figure 13 compares the denoising capabilities of each model, as the optimization task is to minimize dissimilarity between the model estimates and GT.

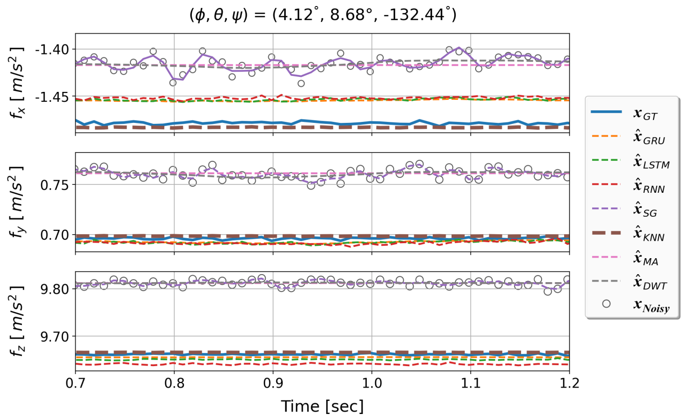

Fig. 13. Reconstruction comparison: Experiment

When compared to the simulated dataset, noise regime (bubbles) exhibits more bias but less stochastic noise, as fluctuations are smaller. Here as well, the learning-based algorithms lay significantly closer to GT measurements, outperforming the SP-based models which lay in the vicinity of the noisy measurements. It is not unusual however, as conventional filters may excel at spatial filtering, but in the absence of a calibration process, constant biases remain unfiltered.
Table VII summarizes the models reconstruction errors when computed across the entire experimental test-set.

TABLE VII
RECONSTRUCTION COMPARISON: EXPERIMENT

Model	RMSE [m/s^2]	MAE [m/s^2]	PSNR [dB]	RAE [%]
kNN	0.00969	0.00658	60.09720	0.171414
GRU	0.01373	0.01016	57.07227	0.242834
LSTM	0.01414	0.01082	56.82121	0.249957
RNN	0.01505	0.01231	56.27788	0.266084
SG	0.10317	0.09124	39.81290	1.727121
DWT	0.10326	0.09126	39.55169	1.823734
MA	0.10588	0.09396	39.52071	1.825294
Noisy	0.10843	0.09427	39.53768	1.828244

Once again, the kNN model demonstrates superiority among all models, as its relative absolute error reduces error by an order of magnitude, as given by the suppression ratio

$$\gamma_{exp.} = \frac{\text{RAE}_{\text{kNN}}}{\text{RAE}_{\text{Noisy}}} \approx 9.38\% \tag{42}$$

The strong differences between $\gamma_{sim.}$ and $\gamma_{exp.}$ can be explained by unknown error sources that are present in real-world devices, but underestimated or mismodeled during the simulation phase, thus beneficial to the experimental results.
Next, the reconstructed signals shown in Figure 13, are used to estimate the roll and pitch angles of the SCA procedure, as shown in Figure 14.

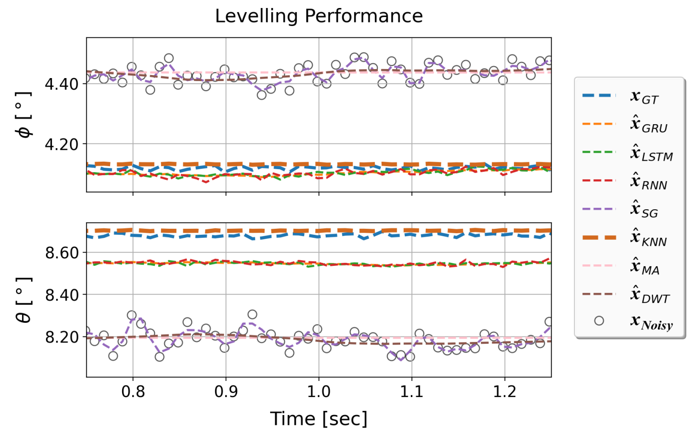

Fig. 14. SCA comparison: Experiment

The learning-based denoisers exhibit better performances than the SP-based, as their outputs manage to minimize the distance error with respect to the GT angles. Focusing again on the best estimator, Figure 15 shows angular errors of computed roll and pitch angles, comparing noisy measurements (blue) with kNN estimates (brown), across the experimental test-set.

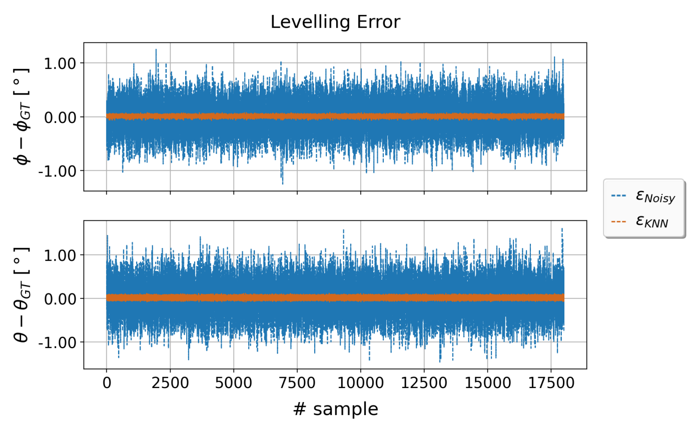

Fig. 15. SCA results of kNN: Experiment

Despite greater noise amplitude of the instruments, the kNN model manages to reduce reconstruction errors drastically. Table VIII shows the overall improvement rate of the computed angles, using RMSE performance indicators to compare the kNN contribution with respect to noisy sensor readings.

TABLE VIII
OVERALL IMPROVEMENT RATE: EXPERIMENT

	$\varepsilon_{\text{Noisy}}$ [°]	ε_{kNN} [°]	ratio [%]
RMSE(ϕ)	0.30042	0.03398	11.311
RMSE(θ)	0.50251	0.06676	13.285

Similarly to the reconstruction results (42), the angular errors are also reduced by one order of magnitude, confirming the contribution of learning-based denoising techniques.

To conclude the experimental section, accelerometer readings from a commercial-grade smartphone sensor were used as noisy measurements, whereas accurate readings from an aligned high-end sensor were used as a GT reference. The denoising capabilities of different models were investigated, followed by examining their contribution to the SCA procedure. Similarly to the simulated scenario, our proposed kNN algorithm happened to outperform all other models, managing to reduce the angular errors by one order of magnitude. We explain its superiority by the simplistic non-parametric approach, which relies on semantic similarity between independent variables, here specific force measurements, with respect to k nearest data points. It is noteworthy that all visualizations were deliberately focused on short time periods, to emphasize local fluctuations and responses.

VII. Conclusions

In this work we addressed the challenging and previously unresearched task of denoising accelerometer signals using data-driven methods, over simulated and experimental datasets. By implementing and adjusting a wide range of denoising algorithms from two different approaches, we assessed their denoising capabilities in terms of pure signal reconstruction, followed by improvement of an SCA procedure. The results showed a clear advantage of the learning-based methods over conventional signal processing algorithms, presumably due to their generalization ability to compensate a wide range of error sources. Improvement rates seemed to vary between both datasets, depending on the trade-off between noise components, as instruments exhibited a more intensive noise regime. While errors in the simulated assessment dropped to half, in the experimental assessment they were reduced by more than one order of magnitude. That is to say that denoising accelerometers by learning-based approaches, can be highly fruitful not only for pure noise reduction, but especially for subsequent INS-related tasks.

References

[1] K. R. Britting, *Inertial Navigation Systems Analysis*. John Wiley & Sons, Inc., 1971.

[2] C. Jekeli, *Inertial navigation systems with geodetic applications*. Walter de Gruyter Berlin, Germany, 2000.

[3] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. American Institute of Aeronautics and Astronautics and the Institution of Electrical Engineers, 2004.

[4] P. D. Groves, *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[5] M. Dadafshar, "Accelerometer and gyroscopes sensors: operation, sensing, and applications," *Maxim Integrated [online]*, 2014.

[6] R. Gonzalez and C. A. Catania, "A statistical approach for optimal order adjustment of a moving average filter," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2018, pp. 1542–1546.

[7] A. Waegli, J. Skaloud, S. Guerrier, M. E. Parés, and I. Colomina, "Noise reduction and estimation in multiple micro-electro-mechanical inertial systems," *Measurement Science and Technology*, vol. 21, no. 6, p. 065201, 2010.

[8] Z. Diao, H. Quan, L. Lan, and Y. Han, "Analysis and compensation of mems gyroscope drift," in *2013 Seventh International Conference on Sensing Technology (ICST)*. IEEE, 2013, pp. 592–596.

[9] S. Yong, C. Jiabin, S. Chunlei, and H. Yongqiang, "Research on the compensation in mems gyroscope random drift based on time-series analysis and kalman filtering," in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 2078–2082.

[10] Y.-H. Tu and C.-C. Peng, "An arma-based digital twin for mems gyroscope drift dynamics modeling and real-time compensation," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 2712–2724, 2020.

[11] M. Ding, Z. Shi, B. Du, H. Wang, L. Han, and Z. Cheng, "The method of mems gyroscope random error compensation based on arma," *Measurement Science and Technology*, vol. 32, no. 12, p. 125109, 2021.

[12] S. Nassar, K.-P. Schwarz, A. Noureldin, *et al.*, "Modeling inertial sensor errors using autoregressive (ar) models," *NAVIGATION, Journal of the Institute of Navigation*, vol. 51, no. 4, pp. 259–268, 2004.

[13] S. Nassar, "Accurate insidgps positioning using ins data de-noising and autoregressive (ar) modeling of inertial sensor errors," *Geomatica*, vol. 59, no. 3, pp. 283–294, 2005.

[14] X. Wang and L. Wang, "Time-serial modeling and kalman filter of mems gyroscope random drift compensation," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. IEEE, 2018, pp. 1–5.

[15] G. Yuan, H. Liang, K. He, and Y. Xie, "Research on signal de-noising technique for mems gyro," in *2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics*. IEEE, 2010, pp. 1288–1291.

[16] Y. Gan, L. Sui, J. Wu, B. Wang, Q. Zhang, and G. Xiao, "An emd threshold de-noising method for inertial sensors," *Measurement*, vol. 49, pp. 34–41, 2014.

[17] C. Liu, Z. Yang, Z. Shi, J. Ma, and J. Cao, "A gyroscope signal denoising method based on empirical mode decomposition and signal reconstruction," *Sensors*, vol. 19, no. 23, p. 5064, 2019.

[18] X. Wang, H. Cao, Y. Jiao, T. Lou, G. Ding, H. Zhao, and X. Duan, "Research on novel denoising method of variational mode decomposition in mems gyroscope," *Measurement Science Review*, vol. 21, no. 1, pp. 19–24, 2021.

[19] C. Shen, J. Li, X. Zhang, Y. Shi, J. Tang, H. Cao, and J. Liu, "A noise reduction method for dual-mass micro-electromechanical gyroscopes based on sample entropy empirical mode decomposition and time-frequency peak filtering," *Sensors*, vol. 16, no. 6, p. 796, 2016.

[20] X. Guo, C. Sun, P. Wang, and L. Huang, "Hybrid methods for mems gyro signal noise reduction with fast convergence rate and small steady-state error," *Sensors and Actuators A: Physical*, vol. 269, pp. 145–159, 2018.

[21] Y. Liu, G. Chen, Z. Wei, J. Yang, and D. Xing, "Denoising method of mems gyroscope based on interval empirical mode decomposition," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[22] T. L. Grigorie, I. R. Edu, and J. I. Corcau, "Fuzzy logic denoising of the miniaturized inertial sensors in redundant configurations," in *Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces*. IEEE, 2011, pp. 521–526.

[23] T. L. Grigorie and R. M. Botez, "Positioning monitoring improvement in a horizontal plane ins by using fuzzy logic data fusion for denoising of inertial sensors in redundant clusters," *International Journal of Fuzzy Systems And Advanced Applications*, vol. 2, pp. 33–40, 2015.

[24] Q. Li, X. Chen, and W. Xu, "Noise reduction of accelerometer signal with singular value decomposition and savitzky-golay filter," *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, vol. 10, no. 15, pp. 4783–4793, 2013.

[25] K. Nirmal, A. Sreejith, J. Mathew, M. Sarpotdar, A. Suresh, A. Prakash, M. Safonova, and J. Murthy, "Noise modeling and analysis of an imu-based attitude sensor: improvement of performance by filtering and sensor fusion," in *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II*, vol. 9912. International Society for Optics and Photonics, 2016, p. 99126W.

[26] M. Karaim, A. Noureldin, and T. B. Karamat, "Low-cost imu data denoising using savitzky-golay filters," in *2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. IEEE, 2019, pp. 1–5.

[27] J. He, C. Sun, and P. Wang, "Noise reduction for mems gyroscope signal: a novel method combining acmp with adaptive multiscale sg filter based on ama," *Sensors*, vol. 19, no. 20, p. 4382, 2019.

[28] C. W. Kang, C. H. Kang, and C. G. Park, "Wavelet denoising technique for improvement of the low cost mems-gps integrated system," *Wiley New York*, 2010.

[29] C.-H. Kang, S.-Y. Kim, and C.-G. Park, "Improvement of a low cost mems inertial-gps integrated system using wavelet denoising tech-

niques," *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 4, pp. 371–378, 2011.

[30] N. El-Sheimy, S. Nassar, and A. Noureldin, "Wavelet de-noising for imu alignment," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 10, pp. 32–39, 2004.

[31] F. Liu, F. Liu, W. Wang, and B. Xu, "Mems gyro's output signal de-noising based on wavelet analysis," in *2007 International Conference on Mechatronics and Automation*. IEEE, 2007, pp. 1288–1293.

[32] Z.-p. Li, Q.-j. Fan, L.-m. Chang, and X.-h. Yang, "Improved wavelet threshold denoising method for mems gyroscope," in *11th IEEE International Conference on Control & Automation (ICCA)*. IEEE, 2014, pp. 530–534.

[33] J. Song, Z. Shi, B. Du, L. Han, H. Wang, and Z. Wang, "Mems gyroscope wavelet de-noising method based on redundancy and sparse representation," *Microelectronic Engineering*, vol. 217, p. 111112, 2019.

[34] Y. S. Shi and Z. F. Gao, "Study on mems gyro signal de-noising based on improved wavelet threshold method," in *Applied Mechanics and Materials*, vol. 433. Trans Tech Publ, 2013, pp. 1558–1562.

[35] J. Yuan, Y. Yuan, F. Liu, Y. Pang, and J. Lin, "An improved noise reduction algorithm based on wavelet transformation for mems gyroscope," *Frontiers of Optoelectronics*, vol. 8, no. 4, pp. 413–418, 2015.

[36] A. S. El-Wakeel, A. Noureldin, H. S. Hassanein, and N. Zorba, "Utilization of wavelet packet sensor de-noising for accurate positioning in intelligent road services," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 1231–1236.

[37] T. A. Ali and A. M. Hasan, "A wavelet-NARX model for SDINS/GPS integration system," *IOP Conference Series: Materials Science and Engineering*, vol. 1094, no. 1, p. 012066, feb 2021. [Online]. Available: https://doi.org/10.1088/1757-899x/1094/1/012066

[38] M. Ben, W. J. Wei, W. J. Tong, and Z. X. Mei, "Mems gyro denoising based on second generation wavelet transform," in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*. IEEE, 2010, pp. 255–258.

[39] R. Gonzalez and C. A. Catania, "Time-delayed multiple linear regression for de-noising mems inertial sensors," *Computers & Electrical Engineering*, vol. 76, pp. 1–12, 2019.

[40] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising imu gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4796–4803, 2020.

[41] C. Jiang, S. Chen, Y. Chen, B. Zhang, Z. Feng, H. Zhou, and Y. Bo, "A mems imu de-noising method using long short term memory recurrent neural networks (lstm-rnn)," *Sensors*, vol. 18, no. 10, p. 3470, 2018.

[42] C. Jiang, S. Chen, Y. Chen, Y. Bo, L. Han, J. Guo, Z. Feng, and H. Zhou, "Performance analysis of a deep simple recurrent unit recurrent neural network (sru-rnn) in mems gyroscope de-noising," *Sensors*, vol. 18, no. 12, p. 4471, 2018.

[43] Z. Ruoyu, G. Shuang, and C. Xiaowen, "Modeling of mems gyro drift based on wavelet threshold denoising and improved elman neural network," in *2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. IEEE, 2019, pp. 1754–1761.

[44] Z. Zhu, Y. Bo, and C. Jiang, "A mems gyroscope noise suppressing method using neural architecture search neural network," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[45] C. Zhu, S. Cai, Y. Yang, W. Xu, H. Shen, and H. Chu, "A combined method for mems gyroscope error compensation using a long short-term memory network and kalman filter in random vibration environments," *Sensors*, vol. 21, no. 4, p. 1181, 2021.

[46] S. Han, Z. Meng, X. Zhang, and Y. Yan, "Hybrid deep recurrent neural networks for noise reduction of mems-imu with static and dynamic conditions," *Micromachines*, vol. 12, no. 2, p. 214, 2021.

[47] C. Jiang, Y. Chen, S. Chen, Y. Bo, W. Li, W. Tian, and J. Guo, "A mixed deep recurrent neural network for mems gyroscope noise suppressing," *Electronics*, vol. 8, no. 2, p. 181, 2019.

[48] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[49] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[51] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[52] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[53] A. Shurin, A. Saraev, M. Yona, Y. Gutnik, S. Faber, A. Etzion, and I. Klein, "The autonomous platforms inertial dataset," *IEEE Access*, vol. 10, pp. 10 191–10 201, 2022.

[54] I. Labs, "Motion Reference Unit," https://www.inertiallabs.com/mru-datasheet, 2021, [Online; accessed 19-July-2021].

[55] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *Proceedings of the 19th ACM international conference on multimodal interaction*, 2017, pp. 216–220.

[56] H. Ohashi, M. Al-Nasser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, and A. Dengel, "Augmenting wearable sensor data with physical constraint for dnn-based human-action recognition," in *ICML 2017 times series workshop*, 2017, pp. 6–11.

[57] O. Steven Eyobu and D. S. Han, "Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network," *Sensors*, vol. 18, no. 9, p. 2892, 2018.

Daniel Engelsman received B.Sc. in Mechanical Engineering from Afeka College of Engineering (2017) and M.Eng. in Aerospace Engineering from the Technion-Israel Institute of Technology (2020). He is currently working as an algorithm researcher in the Autonomous Navigation and Sensor Fusion Laboratory at the University of Haifa, Department of Marine Technologies. His research interest is focused on the thin line between conventional filtering techniques and learning-based approaches for improving performance of inertial navigation systems.

Itzik Klein (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in Aerospace Engineering from the Technion - Israel Institute of Technology, Haifa, Israel, in 2004 and 2007, respectively, and a Ph.D. degree in Geo-information Engineering from the Technion - Israel Institute of Technology, in 2011. He is currently an Assistant Professor, heading the Autonomous Navigation and Sensor Fusion Lab, at the Hatter Department of Marine Technologies, University of Haifa. His research interests include data driven based navigation, novel inertial navigation architectures, autonomous underwater vehicles, sensor fusion, and estimation theory.