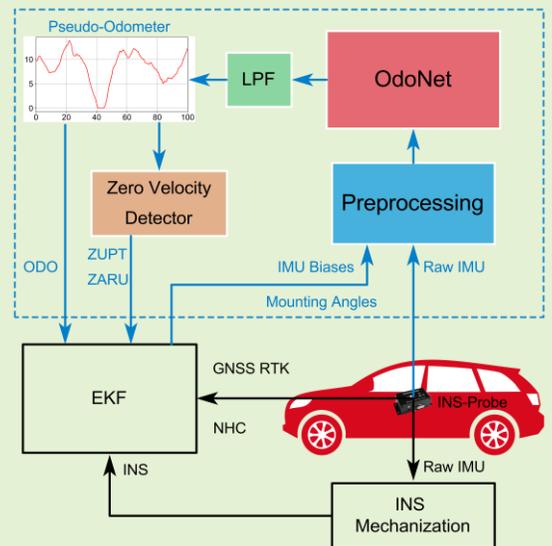


OdoNet: Untethered Speed Aiding for Vehicle Navigation Without Hardware Wheeled Odometer

Hailiang Tang, Xiaoji Niu, Tisheng Zhang, You Li and Jingnan Liu

Abstract—Odometer has been proven to significantly improve the robustness and accuracy of the Global Navigation Satellite System / Inertial Navigation System (GNSS/INS) integrated vehicle navigation in GNSS-denied environments. However, odometer is inaccessible in many applications, especially for aftermarket devices and smartphones. To apply forward speed aiding without hardware wheeled odometer, we propose OdoNet, an untethered one-dimensional Convolution Neural Network (CNN)-based pseudo-odometer model learning from a single Inertial Measurement Unit (IMU). Dedicated experiments have been conducted to verify the generalization capability and the precision of the OdoNet. The results indicate that the IMU individuality, the vehicle loads, and the road conditions have little impact on the robustness and precision of the OdoNet, while the IMU biases and the mounting angles may notably ruin the OdoNet. Hence, a data-cleaning procedure is adopted to effectively mitigate the impacts of the IMU biases and the mounting angles. Compared to the processing mode using only non-holonomic constraint (NHC), by employing the pseudo-odometer, the positioning error is reduced by around 68%, while the percentage is around 74% for the hardware wheeled odometer. In conclusion, the proposed OdoNet can be employed as an untethered pseudo-odometer for vehicle navigation.

Index Terms—Pseudo sensor, deep learning, inertial measurement unit, GNSS/INS integration



I. INTRODUCTION

The Global Navigation Satellite System / Inertial Navigation System (GNSS/INS) integrated navigation system can provide full navigation parameters, including position, velocity, and attitude, and thus has been widely used in land vehicles. However, GNSS can be easily interfered or interrupted in urban environments. Inertial measurement units (IMU) can work independently, and INS can provide continuous navigation service during GNSS outages, which greatly improves the usability of the integrated navigation system. Being restricted by the poor precision and the low stability of the Micro-Electro-Mechanical System (MEMS) IMU, the MEMS integrated navigation system cannot maintain high-accuracy positioning in GNSS-denied environments, especially when the vehicle travels through city canyons, long tunnels, and

underground roads, in where GNSS tends to have long outages. Consequently, extra aiding sources or constraints, such as zero velocity update (ZUPT) [1], non-holonomic constraint (NHC) [2]-[4], wheeled odometer [4], and et. al, have to be employed. ZUPT and NHC are kinds of virtual velocity observations from the vehicle or the IMU itself, which can be achieved independently. However, to use odometer aiding, we must attach an extra encoder to the wheel, or grab speed message from the Controller Area Network (CAN) or other networks inside the vehicle, which greatly increases the cost and complexity of the system integration. As a consequence, the wheeled odometer cannot be applied in many applications, especially for aftermarket devices and smartphones, which leads to poor positioning accuracy during GNSS outages, and thus significantly reduces the reliability of the integrated system in GNSS-denied environments.

Sufficient experiment results have demonstrated that it is possible to estimate the vehicle speed by learning from a single IMU. In this paper, we propose OdoNet, an untethered convolution neural network (CNN)-based pseudo-odometer model, which can directly estimate the vehicle speed. The main contributions of this paper are listed as follows:

- To apply forward speed aiding without hardware wheeled odometer, we propose a CNN-based pseudo-odometer to

This research is funded by the National Key Research and Development Program of China (No. 2020YFB0505803), and the National Natural Science Foundation of China (No. 41974024). (Corresponding author: Tisheng Zhang)

Hailiang Tang, Xiaoji Niu, Tisheng Zhang, and Jingnan Liu, are with the GNSS Research Center, Wuhan University, Wuhan 430079, China (e-mail: zts@whu.edu.cn).

You Li is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China.

estimate the forward speed of the vehicle. The pseudo-odometer combined with an effective zero-velocity detector is employed in the integrated navigation system to improve the positioning accuracy in GNSS-denied environments.

- To fully verify the generalization capability and the precision of the OdoNet, dedicated experiments were carried out. The experiment results indicate that the IMU individuality, the vehicle loads, and the road conditions have little effect on the OdoNet, while the IMU biases and the mounting angles may notably ruin the OdoNet.

- To mitigate the impacts of the IMU biases and the mounting angles, a data-cleaning procedure is adopted to precisely compensate for these factors. The data-cleaning procedure is proven to significantly improve the robustness and the precision of the OdoNet.

The remainder of this paper is organized as follows: In the next section, a brief literature review on IMU-based deep learning network is presented. Then, the details of the OdoNet architecture and the integrated navigation system enhanced by it are described. The experiments and the results are presented to evaluate the precision and robustness of the proposed method. Finally, we conclude the proposed method.

II. RELATED WORKS

The conventional GNSS/INS integrated navigation has been studied for decades, so only deep neural networks (DNN) will be discussed here. According to the application of the deep learning network, we can classify these networks into two categories, including IMU-based odometry and velocity-estimation networks.

A. IMU-based Odometry

Recently, more and more researchers have focused on the end-to-end DNN based IMU, such as inertial odometry [14]-[18] and visual-inertial odometry (VIO) [19]-[22]. The inertial odometry only uses IMU data to estimate the 6-DOF pose, while VIO uses both IMU and camera data.

For inertial-only odometry, Chen proposed IONet [14], a two-layer Bi-LSTM network to learn location transforms in polar coordinates from raw IMU data, but the network was only for 2-dimensional (2D) navigation. As a future work, Chen released OxIOD [15], the dataset for deep inertial odometry. However, the dataset is only available for low-speed PDR applications. Yan [16] proposed three novel neural inertial navigation architectures for Pedestrian Dead Reckoning (PDR) applications on mobile phones, which were proved to outperform previous methods. Esfahani presented AbolDeepIO [17], a triple channel long-short time memory (LSTM)-based DNN for vehicles, which could extract features from accelerometer, gyroscope, and time interval. However, the proposed model did not consider the vehicular characteristics. Afterward, an IMU-based 6-DOF inertial odometry was proposed in [18], which combined CNN with Bidirectional Long-short Time Memory (Bi-LSTM), but the experiments showed large increased error along the vertical axis.

DNN-based VIO, including supervised learning, unsupervised learning, and self-supervised, has also been studied for many years. VINet [19] was an end-to-end deep

learning VIO network, which applied CNN to extract features from the images and LSTM to process IMU, and used a core LSTM to combine visual and inertial data. VIO Learner [20]-[21] was an unsupervised deep network of VIO, which could correct errors online. In [22], a novel self-supervised deep learning VIO network was proposed, which adopted 3D geometric constraints and updated additional bias for IMU using the pose feedback. Many researchers have also focused on the deep learning network combining LiDAR and IMU [23]-[24].

DNN-based odometry substitutes neural networks for precise mathematical models, which completely changes the navigation algorithms. To train a general DNN model, we must carry out heavy data collection work, and even so, we still cannot guarantee that the model can work every time in every scene. In practical application, there is still a lot of work to do for DNN-based odometry.

B. Velocity Estimation

Using IMU data to estimate velocity has also become a research hotspot. Some researchers only construct a classification problem to detect some special states, like zero velocity and zero angular velocity states. Others employ IMU-based DNN to directly estimate velocity.

ZUPT is an effective approach to constrain INS drifts in PDR [1], [5]-[7] and VDR [9] (Vehicle Dead Reckoning) applications. Xinguo et al. proposed AZUPT [5], a CNN-based zero-velocity detector, which could adaptively pick ZUPT points in different motion types. In [6]-[7], LSTM-based recurrent neural network (RNN) [8] was used as a zero-velocity detector to improve the foot-mounted inertial navigation. Zero-velocity states can also be detected using regressed velocity from a DNN, and thus directly estimating velocity may be more effective and reasonable.

Brossard [9] applied an LSTM network to detect a variety of situations of interest on wheels, including zero velocity, zero angular velocity, zero lateral velocity, and zero vertical velocity. These constraints were treated as pseudo-measurements to refine the estimates of the IEKF (Invariant Extended Kalman Filter). However, if we model the lever-arms and the mounting angles of the IMU to the vehicle precisely, then the NHC (zero lateral velocity and zero vertical velocity) is typically satisfied in the non-steering wheel of the vehicle, except for some special cases. In other words, DNN can be applied to directly estimate the vehicle speed, not just NHC.

Moreover, the IMU-based DNN, which directly estimates the velocity, is also a research hotspot. A 2D CNN-based deep-learning model was used to predict 3D speed based on a window of IMU samples for mobile phone applications [10], and the regressed speed was used as a pseudo measurement in Extended Kalman Filter (EKF). RIDI [11] applied Support Vector Machine and Support Vector Regression to form a regression model and predicted velocity in smart-phone for correcting low-frequency biases in the accelerations, which were integrated twice to estimate position. A hybrid network [12]-[13] was proposed for PDR applications, which used a CNN to extract features from a signal stream, and a Bi-LSTM to estimate velocity from the extracted features.

As one can see, most of the velocity estimating models are designed for PDR applications, which are not suitable for

vehicle navigation. The vehicle has many characteristics that are completely different from mobile phones or foot-mounted devices. In addition, previous works show few pieces of evidence to prove the robustness and generalization capability of their network, which are the most important characteristics for data-driven algorithms. For vehicle integrated navigation system, what we need is an aiding source that can maintain high-accuracy positioning during GNSS signal blockage or degradation, taking our OdoNet for example. Furthermore, estimating only the forward speed of the vehicle rather than three-dimensional speed can reduce the complexity of the DNN model, and thus improve the robustness and practicality.

III. PROPOSED METHOD

This section will describe the details of the proposed method, including the deep learning architecture of the OdoNet, the indispensable data-cleaning procedure, and the framework of the integrated navigation system enhanced by the OdoNet.

A. OdoNet Architecture

According to our literature review, various basic models have been applied for IMU-based DNN, including LSTM [6]-[7], [9], [16]-[17], Bi-LSTM [14], [16], CNN [5], [10], [15]-[16], and even hybrid model [12]-[13], [18]. RNN based networks, including LSTM and Bi-LSTM, are typically used for time-series data processing, e.g. time-series of IMU data. Although RNN-based networks might have better precision, their convergence speed is extremely low during model training, which greatly increases the verification cycle of the model. According to our experiments in vehicle speed estimation, CNN outperforms RNN in both precision and efficiency running in a GPU, which will be proved in IV.B. Hence, we pick the one-dimensional (1D) CNN model, and the network architecture of the OdoNet is shown in Fig. 1. As for the size of the window, many previous works [12]-[13], [17] have evaluated its effects on their networks. If the window size is too small, it is hard to extract enough features to estimate speed, while it might cause a long-time delay if the window size is too large. For our OdoNet, experiment results demonstrate that one second is the most appropriate window size. Whilst, the output rate of the 6-axes IMU in our self-developed GNSS/INS integrated system is 50 Hz, so the input data is treated as a 1D

vector with a length of 50 and a depth of 6.

As shown in Fig. 1, the OdoNet consists of four 1D convolution layers, with max-pooling layers for sub-sampling. After flattening depth, three fully connected layers are employed and output the regressed speed. The activation functions between two layers are ReLU [32] units, except the pooling layers. Moreover, two dropout layers with probability of 0.5 are used before the first two fully connected layers, to avoid over-fitting during training. The kernel size, the depth of the convolution layer, and the units of the fully connected layer can be found in Fig. 1. Hence, the input and output of the OdoNet can be described as follows:

$$(\mathbf{w}_{iv}^v, \mathbf{f}^v)_{50 \times 6} \xrightarrow{f_\theta} s \cdot \tilde{v}_{odo} \quad (1)$$

where \mathbf{w}_{iv}^v and \mathbf{f}^v are the IMU measurements processed in the data-cleaning procedure, i.e., the biases and mounting angles compensated measurements in the vehicle-frame (v-frame), see III.B; f_θ is the OdoNet itself; s is the scale factor of the output, and \tilde{v}_{odo} is the regressed speed. For typical land vehicle applications, the maximum speed of the vehicle can reach 30 m/s in an urban environment. Hence, we use a fixed scale factor of $s = 1/30$ to normalize the speed, which can accelerate the convergence of the model during training. To solve the optimal parameter θ^* inside the OdoNet, we can minimize the loss function l on the training dataset as follows:

$$\theta^* = \operatorname{argmin} l(f_\theta(\mathbf{w}_{iv}^v, \mathbf{f}^v), s \cdot v_{odo}) \quad (2)$$

The loss function is defined as the mean squared error (MSE) between the truth $s \cdot v_{odo}$ and the regressed value $s \cdot \tilde{v}_{odo}$ as follows:

$$l = \frac{1}{N} \sum (s \cdot v_{odo} - s \cdot \tilde{v}_{odo})^2 \quad (3)$$

where N is the size of the IMU samples.

Since ReLU activation function [32] is applied in the network, which means that the output of the network is always greater than or equal to zero, this characteristic provides an opportunity for zero-velocity detection. In other words, the regressed speed is zero or close to zero during stationary states. Consequently, we can apply a fixed threshold detector to determine zero-velocity states. Mathematically, we can

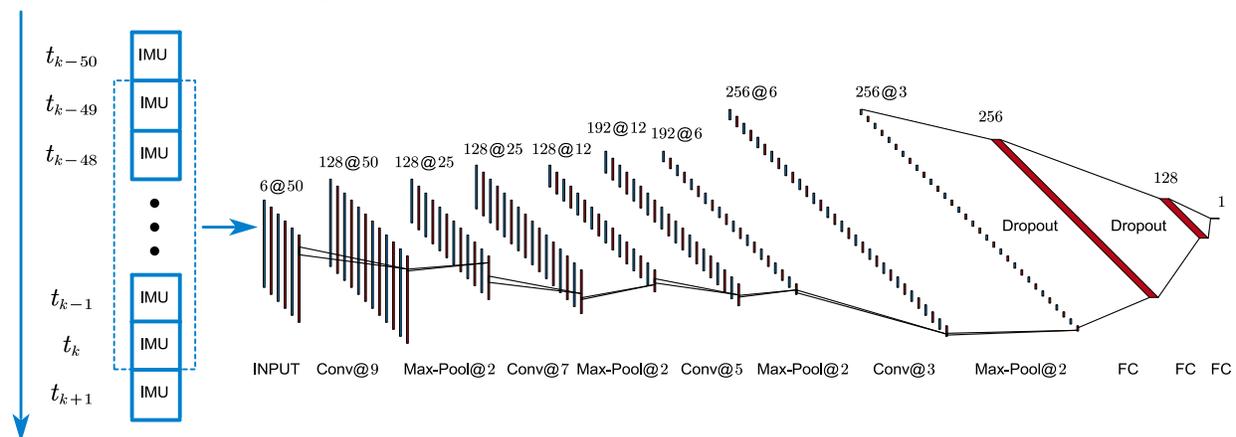


Fig. 1. Proposed OdoNet network architecture. 6-axes IMUs with the output rate of 50 Hz are used, and thus the size of the input layer is 6@50. The INPUT, Conv, Max-Pool and FC represent input layer, convolution layer, max pooling layer, and fully connected layer, respectively. The name of each layer is shown below the graph, with the kernel size for convolution and max pooling layers. The numbers above the network are the depth@length for convolution and max pooling layers, or the units for the fully connected layers. The schematic of the DNN is generated by [29].

formalize the detection problem as a binary hypothesis testing problem, where the detector can choose between the following two hypotheses [1]:

$$\begin{aligned} H_0: & \text{IMU is moving} \\ H_1: & \text{IMU is stationary} \end{aligned} \quad (4)$$

According to the experience of engineering, the threshold of the zero-velocity detector is set to 0.1 m/s. Hence, the judgment for these two hypotheses can be written as follows:

$$\begin{aligned} H_0: & \tilde{v}_{odo} \geq 0.1 \text{ m/s} \\ H_1: & \tilde{v}_{odo} < 0.1 \text{ m/s} \end{aligned} \quad (5)$$

It is generally known that the robustness and precision of a DNN are affected by various factors. The possible factors include the IMU biases, the mounting angles, the IMU individuality, the vehicle loads, and the road conditions. In section IV. D, experiment results will be presented to demonstrate the robustness and the generalization capability of the OdoNet. It should be noted that the OdoNet can only estimate positive speed, i.e. the OdoNet will fail if the vehicle run in reverse direction.

B. Data Cleaning

According to our experiments, we find that the IMU biases and the mounting angles have strong impacts on the precision of the OdoNet. To mitigate or eliminate their effects, we compensate them in this data-cleaning procedure, which has been proven to significantly improve the robustness and the precision of the OdoNet. Specifically, we first compensate the IMU biases, and then convert the biases compensated IMU measurement into v-frame.

Low-cost MEMS IMU has extremely low stability, with poor temperature characteristics, which leads to huge and rapidly changing biases [25]. As a matter of experience, taking ICM20602 for example, the biases of the gyroscope typically range from $-2^\circ/\text{s}$ to $2^\circ/\text{s}$, and the biases of the accelerometer typically range from -0.2 m/s^2 to 0.2 m/s^2 . Moreover, each IMU has different characteristics of biases, which might seriously affect the robustness and precision of the OdoNet. Improving the complexity of the model or using a larger dataset might reduce the impact of the IMU biases, but it involves heavy dataset collecting work and more computational resources. The GNSS/INS integrated system can effectively and accurately estimate the IMU biases, as in (6). With the estimated gyroscope biases $\hat{\mathbf{b}}_g$ and accelerometer biases $\hat{\mathbf{b}}_a$, we can compensate gyroscope measurement $\tilde{\mathbf{w}}_{ib}^b$ and accelerometer measurement $\tilde{\mathbf{f}}^b$ in advance. $(\cdot)^b$ represents the vector in the IMU body frame (b-frame).

As we all know, the calculated speed by INS in v-frame is the function of the lever-arms \mathbf{l}_{wheel}^b and the IMU mounting angles \mathbf{C}_b^v , as can be seen in (10). Hence, the regressed speed estimated by the OdoNet should be in v-frame exactly, rather than in b-frame. In addition, for most vehicle navigation applications, the navigation devices (including the IMU) are usually mounted on the given spot of the vehicle, which means that the lever-arms are almost unchanged. However, the mounting angles of the IMU with respect to the vehicle might be randomly set and unknown, which will change the IMU signal

significantly and ruin the OdoNet. Consequently, the mounting angles must also be compensated in preprocessing. The estimating algorithm of the mounting angles is well described in [26]. The mounting angles are estimated in post-processing currently, but we can also estimate them online. However, the roll mounting angle cannot be estimated due to a lack of observability [26], so the OdoNet must be able to adapt this factor. Estimated pitch mounting angle $\Delta\theta$ and heading mounting angle $\Delta\psi$ can be converted to direction cosine matrix \mathbf{C}_b^v [26]. Then, the raw IMU measurements can be converted from b-frame to v-frame, as \mathbf{w}_{iv}^v and \mathbf{f}^v , see (1).

To sum up, in order to eliminate the impact of the unique IMU biases and the mounting angles, we compensate them in advance, as mentioned in (1). In section IV. D, we will show evidence about the importance of the data-cleaning procedure, and illustrate how these factors affect the robustness and precision of the OdoNet.

C. Integrated Navigation System Framework

The overview of the proposed integrated navigation system for vehicle application is depicted in abstract section. In a conventional GNSS/INS integrated navigation system, raw IMU data is processed by the INS mechanization algorithm [27]-[28], and an EKF is applied to fuse GNSS and INS, combining with non-holonomic constraint or other constraints or aiding sources. However, without odometer aiding, it might still lead to large positioning drifts in GNSS-denied urban environments, especially in the forward direction of the vehicle. Based on the proposed OdoNet, the pseudo-odometer with a zero-velocity detector can be employed in the integrated navigation system. As can be seen in the overview, raw IMU samples are compensated by the estimated IMU biases and the mounting angles in data-cleaning procedure. With the compensated IMU samples, the OdoNet predicts the raw vehicle speed, which is then filtered by an FIR (Finite Impulse Response) low-pass filter. The parameters of the FIR filter are tuned by evaluating the error of the filtered the speed, as can be seen in section IV.C. By considering the precision and time delay of the filtered speed, the cut-off frequency of the FIR filter is set to 0.1 Hz and its order is set to 64. The filtered vehicle speed is treated as a pseudo-odometer measurement and is also used to determine the zero-velocity state of the vehicle. Hence, the pseudo-odometer speed update, the zero-velocity update (ZUPT), and the zero angular rate update (ZARU) can be employed in EKF.

The error state vector of the used EKF is defined as follows:

$$\mathbf{x} = [\delta\mathbf{r}^n \ \delta\mathbf{v}^n \ \psi \ \delta\mathbf{b}_g \ \delta\mathbf{b}_a]^T, \quad (6)$$

where $\delta\mathbf{r}^n$ and $\delta\mathbf{v}^n$ can be written as follows:

$$\delta\mathbf{r}^n = [\delta r_N \ \delta r_E \ \delta r_D]^T, \quad (7)$$

$$\delta\mathbf{v}^n = [\delta v_N^n \ \delta v_E^n \ \delta v_D^n]^T, \quad (8)$$

$\delta\mathbf{r}^n$ represents the position error in navigation frame (n-frame) for the NED (North-East-Down) coordinate system; $\delta\mathbf{v}^n$ represents the velocity error; ψ denotes the attitude error using the psi-angle model [27]; $\delta\mathbf{b}_g$ and $\delta\mathbf{b}_a$ represent the gyroscope and accelerometer biases error, respectively. The state equation, the GNSS positioning observation equation, the

implementation of the Kalman filter, and more detailed information can be referred to [27]-[28].

NHC is a kind of virtual velocity observation in the v-frame, which means that there is no lateral and vertical speed for the land vehicle. Odometer can provide forward speed aiding, and NHC and odometer can be combined to achieve three-dimensional velocity aiding in v-frame. The measured velocity in v-frame can be written as follows:

$$\tilde{\mathbf{v}}_{wheel}^v = [\tilde{v}_{odo}^v \ 0 \ 0]^T + \mathbf{e}_v. \quad (9)$$

The calculated velocity in v-frame by INS can be expressed as follows:

$$\hat{\mathbf{v}}_{wheel}^v = \mathbf{C}_b^v \hat{\mathbf{C}}_n^b \hat{\mathbf{v}}_{IMU}^n + \mathbf{C}_b^v (\hat{\boldsymbol{\omega}}_{nb}^b \times) \mathbf{l}_{wheel}^b, \quad (10)$$

where superscript ($\hat{\cdot}$) denotes the parameters that are calculated by INS; \tilde{v}_{odo}^v denotes the speed of the odometer; \mathbf{e}_v denotes the observation noise; \mathbf{C}_b^v represents the IMU mounting angles with respect to the vehicle frame; \mathbf{l}_{wheel}^b is the lever-arms vector between b-frame and v-frame in b-frame. Typically, the measurement center of velocity observation in the v-frame is defined at the point in the rear wheel, where the wheeled odometer is installed, and it is tangent to the ground, as depicted in Fig. 2.

When the stationary state is determined by the zero-velocity detector, ZUPT and ZARU can be applied to EKF. Zero velocity means that the IMU or the vehicle is stationary, and it can be written as:

$$\tilde{\mathbf{v}}^n = \mathbf{e}_v. \quad (11)$$

By using ZUPT during the stationary state, the drift of velocity and position can be constrained effectively, except the heading angle, and thus ZARU can be used simultaneously. Zero angular rates can be expressed as:

$$\tilde{\boldsymbol{\omega}}_{ib}^b = \mathbf{e}_w, \quad (12)$$

where i denotes the inertial frame (i-frame), and \mathbf{e}_w represents the observation noise.

According to our dedicated experiments, the standard deviation values are set to at 0.1 m/s for both NHC and ZUPT and 0.1 °/s for ZARU. As for the odometer, we should consider its precision. Specifically, the standard deviation value of the real wheeled odometer is set according to its resolution, i.e. 0.1 m/s for the used wheeled odometer (DFS60E). The value for the pseudo-odometer is derived from its evaluation metrics, i.e. 0.3 m/s for the pseudo-odometer, see IV.C and IV.E. However, the basic hypothesis of NHC is not satisfied occasionally in some extreme cases, like slippage. The OdoNet might fail to estimate and present wrong speed with large error, and the zero-velocity detector might also give wrong judgment. Consequently, a fault detection algorithm can be applied to eliminate gross error in these observations. We use innovation [28] in EKF to detect and reject faults. The normalized innovation of the dimension j at time k can be defined as:

$$y_{k,j}^- = \frac{\delta z_{k,j}^-}{\sqrt{\mathbf{C}_{k,j,j}^-}}, \quad (13)$$

where $\delta z_{k,j}^-$ is the measurement innovation of this dimension; $\mathbf{C}_{k,j,j}^-$ is the covariance of this innovation. For the

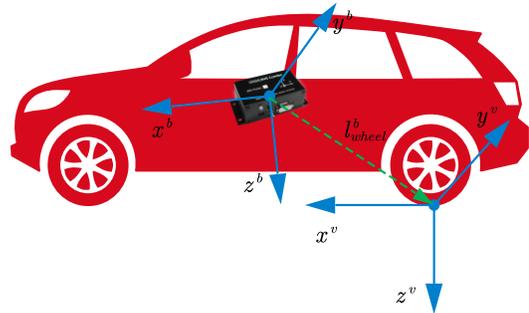


Fig. 2. Definitions of the vehicle frame (v-frame) and the IMU body frame (b-frame), and illustration of the IMU mounting angles and the lever-arms with respect to the v-frame.

one-dimensional observation, if the degree of confidence is set to 95%. The observation with $y_{k,j}^- > 3.84$ (3.84 is the chi-square test parameter for the degree of confidence of 95%) will be judged as outlier or gross error, and will not be employed in the EKF.

IV. EXPERIMENTS AND RESULTS

This section examines the proposed OdoNet and the integrated navigation system enhanced by it. A comprehensive vehicle dataset was collected to train and evaluate the OdoNet. After preprocessing the dataset, the model was trained. Then, the metrics of the regressed speed and zero-velocity detection on the testing dataset were evaluated. Next, dedicated experiments were carried out to evaluate the possible factors that might affect the robustness of OdoNet, so as to verify the generalization capability of the OdoNet. Finally, we compared the performance of the integrated navigation system on three typical testing sequences using three different aiding sources, to evaluate the contribution of the proposed pseudo-odometer. Experiments mentioned below were all done offline.

A. Dataset Collection

As shown in Fig. 3, the whole sensors setup contains 9 INS-Probe (our self-developed GNSS/INS integrated navigation system) modules, a high-precision wheeled odometer, and a navigation-grade POS (Position Orientation System) POS-A15. INS-Probe modules are responsible for collecting multi-sensors data, which includes GNSS RTK positioning from low-cost GNSS module (ZED-F9P), raw IMU data from MEMS IMU chip (ICM20602), and wheeled odometer (DFS60E). These sensors are precisely synchronized through hardware triggering. The output rate of the IMU is 50 Hz, and the wheeled odometer is synchronously sampled with the IMU. The wheeled odometer is an optic encoder, DFS60E from SICK with 2048 lines, which will be used as a benchmark for the proposed OdoNet. POS-A15 can provide high-precision ground truth for evaluation of the positioning accuracy. The noise parameters used in this paper for ICM20602 and POS-A15 are the same as that in [33].

The whole dataset consists of 14 test sequences, covering various areas in Wuhan city. As depicted in Fig. 4, our dataset contains most of the typical scenes, including freeway, university campus, residential districts, industrial park, and other different testing scenes in urban environments. 9 INS-Probe modules were used, and thus the data of 9 individual IMUs were collected, which greatly enriches the diversity of



Fig. 3. Photograph of the test vehicle and system setup inside and outside the vehicle.

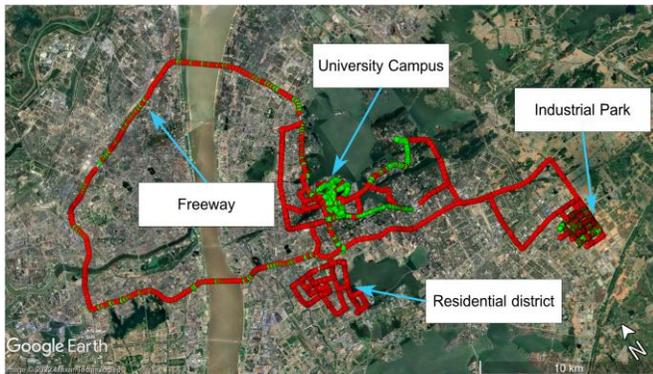


Fig. 4. The trajectory of the collected dataset. The green dots denote the testing dataset, while the red dots denote the training dataset. The map is generated by Google Earth.

the dataset. We split the dataset into two parts, one for training and one for testing. It should be noted that the training dataset and the testing dataset cannot contain the data in the same test sequence or the data collected by the same module, so as to ensure the effectiveness of the model. Specifically, the test sequences and the modules used in model training cannot be used in model testing again. The training dataset contains 10 different sequences per INS-Probe module from 7 modules, that is 70 sequences in total. The testing dataset is collected from the rest two INS-Probe modules, each with 4 sequences, that is 8 sequences in total. The rest of the dataset (48 sequences) will not be used for training either testing.

The labels for model training were generated from the real wheeled odometer. Statistical analysis was carried out on the prepared dataset, as showed in Fig. 5. The distributions of the dataset show that the speeds of most samples are lower than 15 m/s. There is a total of 468786 samples for training, and 56744 samples for testing. The collected dataset contained various road conditions, where the vehicle traveled at different speeds in different scenes. Specifically, the freeway is a high-speed scene, and the university campus and industrial park are low-speed scenes and median-speed scenes respectively.

B. Model Training

Our model was implemented based on the public available Keras framework with TensorFlow. The training was done on an NVIDIA TITAN Xp GPU. The training dataset was shuffled randomly, and 20% of them were used as validation dataset during the training. The hyperparameters in model training were derived by applying grid search to evaluate validation results. As a result, the Adam optimizer [30] was used with a

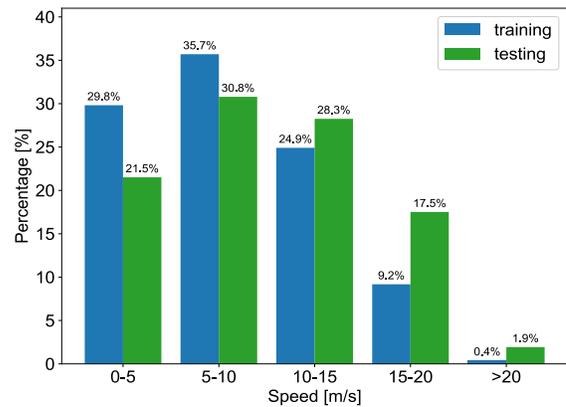


Fig. 5. The distributions of the speed on training and testing dataset.

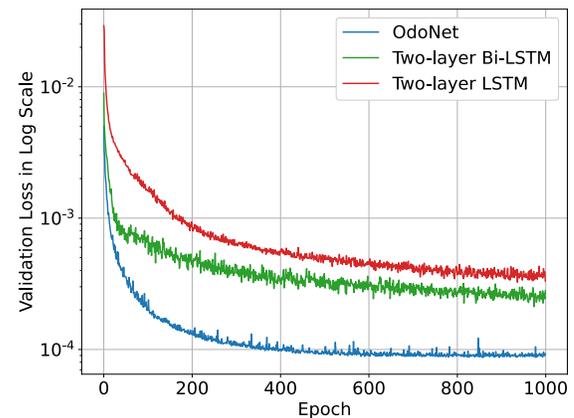


Fig. 6. The validation loss of the OdoNet and other two networks during the training. The loss is in log scale. The Bi-LSTM based network and LSTM based network have the same structures with OdoNet, except the four-layer 1D CNN network. Each layer of the Bi-LSTM has 64 nodes, and each layer of the LSTM has 128 nodes.

learning rate of 0.00005, and a big batch size of 1024 samples was used for better generalization capability and higher precision.

We trained OdoNet for 1000 epochs, costing 7 seconds for each epoch. We also trained a two-layer LSTM network and a two-layer Bi-LSTM network. The LSTM-based network and Bi-LSTM-based network have the same structures as the OdoNet, except the four-layer 1D CNN network. Each layer of the LSTM has 128 nodes, and each layer of the Bi-LSTM has 64 nodes. As depicted in Fig. 6, the validation loss of the OdoNet converged rapidly with satisfying precision. In contrast, the two-layer LSTM network and the two-layer Bi-LSTM have lower convergence speed and precision, even if we have used a larger learning rate of 0.001. In addition, the LSTM and Bi-LSTM consumed much more computational resources, which led to longer training and predicting time. More specifically, the two-layer LSTM network took about 4 times as much time as the OdoNet for each epoch during training, and the value for the two-layer Bi-LSTM was about 8 times. All the above experiment results demonstrate that our CNN-based OdoNet has better performance with higher efficiency, compared to the RNN-based networks.

C. Regressed Speed Evaluation

1) Raw Regressed Speed on Testing Dataset

With well-trained OdoNet, we evaluated the regressed speed

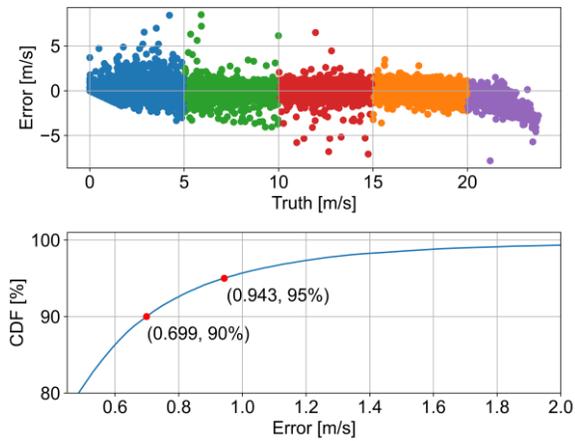


Fig. 7. The distributions of the error of regressed speed and the cumulative distribution function on the testing dataset.

on the testing dataset. The errors of the regressed speed are shown in Fig. 7, where the first subplot depicts the distribution of the regressed errors, and the second subplot depicts the cumulative distribution function of the speed errors. As we can see, 90% of the regressed speed error are less than 0.699 m/s, and 95% of them are less than 0.943 m/s.

More specifically, the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of the regressed speed are listed in Table I. The MAE and RMSE on the overall testing dataset are 0.315 m/s and 0.490 m/s respectively in a range of 0-25 m/s. However, we cannot ignore the fact that the precision of the OdoNet degrades seriously when the speed is higher than 20 m/s, which can be seen in both Fig. 7 and Table I. According to our analysis, lacking enough high-speed samples is the main reason. As depicted in Fig. 5, the samples of speeds higher than 20 m/s are less than 2%. Adding more high-speed samples should be able to solve this issue. The scenes where the vehicle can travel at a high speed of more than 20 m/s are generally wide roads away from tall buildings, e.g. highway, open-sky environments specifically. In general, GNSS can attain high-accuracy RTK positioning in such open-sky areas, and thus these low-precision regressed speeds should have a limited impact on the final results of the integrated navigation system or be eliminated by the gross error detection algorithm (13).

2) Regressed Speed Sequences on Testing Sequences

For each 50 consecutive IMU samples, the OdoNet can predict a corresponding vehicle speed, and thus we can attain speed with an update rate of 50 Hz. The speed sequence is then filtered by the FIR low-pass filter. The testing dataset consists of two INS-Probe modules with 4 different sequences, including three sequences in different speed scenes and one sequence for robustness testing. These three scenes are mentioned in IV. A, and the details of these scenes can refer to IV. E. The RMSE of the raw regressed speed and filtered speed are shown in Table II, which shows great consistency between these two individual IMU. However, the RMSE of the regressed speed in the high-speed scene is much larger than the other two scenes, which corresponds to the previous results in Table I.

3) Zero Velocity Detection on Testing Dataset

We also evaluated the performance of the zero-velocity detector. We used (5) to generate the ground truth of the zero-velocity states from the real wheeled odometer. The

TABLE I
THE ERROR OF REGRESSED SPEED ON TESTING DATASET

Velocity [m/s]	0-5	5-10	10-15	15-20	20-25	Overall
MAE [m/s]	0.300	0.253	0.294	0.397	1.070	0.315
RMSE [m/s]	0.534	0.391	0.419	0.520	1.337	0.490

TABLE II
RMSE OF THE SPEED SEQUENCES IN THREE SCENES

Scene	M1		M2	
	Raw [m/s]	Filtered [m/s]	Raw [m/s]	Filtered [m/s]
Low Speed	0.457	0.339	0.449	0.329
Median Speed	0.381	0.240	0.374	0.249
High Speed	0.618	0.486	0.594	0.449

M1 and M2 represents two different INS-Probe modules in the testing dataset. Raw denotes the raw regressed speed, and filtered denotes the filtered speed.

TABLE III
RESULTS OF ZERO VELOCITY DETECTION USING REGRESSED SPEED

Metrics	Value	Metrics	Value
True Positive	3636	False Positive	178
True Negative	52895	False Negative	35
Precision	95.33%	Recall	99.05%

evaluation metrics for binary classification on the testing dataset are shown in Table III. The precision and recall are 95.33% and 99.05% respectively, which illustrates that the false-alarm rate is slightly larger than the missed-detection rate. The results are reasonable because the regressed speeds are mixed with gross errors. ZUPT and ZARU are extremely important in some scenes because they can improve the precision and stability of the integrated system effectively. Moreover, a false alarm of zero velocity might be eliminated by the fault-detection model in (13). In other words, the results of the zero-velocity detector are acceptable.

D. Robustness Evaluation

In previous section, we evaluate the regressed speed metrics, and the results illustrate that the raw regressed speed on the testing dataset is less than 0.5 m/s in a wide dynamic range of 0-25 m/s. However, the most common challenges for deep-learning network are the robustness and the generalization capability, which determine their usability in more general scenes. In this section, we evaluate some common factors that might affect the robustness of OdoNet, including the compensation of the IMU biases and the mounting angles, the vehicle loads, and the road conditions. Previous experiments have demonstrated that the IMU individuality has little impact on the robustness and precision of the OdoNet, so it will not be evaluated repeatedly. The M1 and M2 mentioned below represent the two INS-Probe modules in the testing dataset.

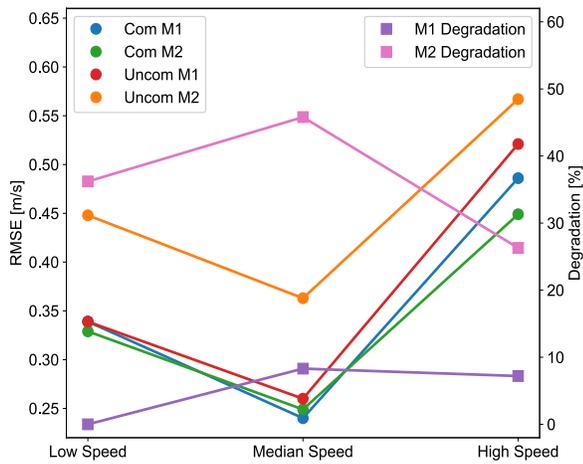


Fig. 8. The RMSE of the filtered speed using compensated model and uncompensated model. The circle represents the speed RMSE, while the square represents the degradation rate. Com denotes that all the IMU sequences are well compensated in training and testing, while Uncom denotes no compensation. The degradation rate is defined as (Uncom-Com)/Com. The uncompensation is only for the IMU biases, and the mounting angles are compensated for all sequences in this test.

1) Impact of Compensation

In the data-cleaning procedure, the IMU biases and the mounting angles are precisely compensated. In this section, experiments will be conducted to demonstrate the impacts of the IMU biases and the mounting angles.

The biases of a single IMU are unique and different from each other, so we trained and tested the OdoNet with raw IMU samples (without compensating for the biases) to evaluate their impacts. We compared the RMSE of the filtered regressed speed of two different models in three testing scenes, as depicted in Fig. 8. The impacts on M1 and M2 are completely different. The uncompensated model has little impact on the precision of the M1, and the RMSE increase by less than 10%. However, the precision of M2 on uncompensated models degrades as much as 45%. A possible explanation might be the insufficient IMU individuality in the training dataset, which is uncontrollable due to the uncertainty of the IMU biases. In contrast, it is convenient to attain IMU biases from the integrated navigation system online. Hence, training with biases calibrated IMU data makes the OdoNet more robust and more precise.

The mounting angles of the IMU in our dataset are small, and each axis is less than 2° typically (according to the estimated results), mainly because the IMU is mounted aligning to the vehicle. We only considered small mounting angles here, and the large mounting angles could be compensated in advance. Consequently, we manually added extra mounting angles to the IMU samples, which means that we rotated the IMU samples with different mounting angles, and evaluated the RMSE of the regressed speed. The data in the median-speed scene is used in this test. As we can see in Fig. 9, the pitch and heading mounting angles have a completely different impact on the precision of the regressed speed. The impact of the pitch mounting angle is much larger than the heading mounting angle. If 0.4 m/s is the acceptable precision, the tolerable pitch mounting angle is about 2° , while the tolerable heading mounting angle is about 12° . It can be explained that the acceleration is directly related to the vehicle speed, and the

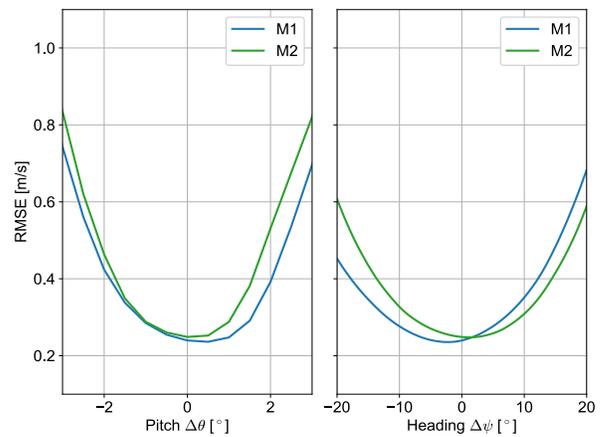


Fig. 9. The RMSE of the filtered speed with the change of mounting angles in the median-speed scene. We evaluated pitch and heading mounting angles separately. The pitch mounting angle $\Delta\theta$ and the heading mounting angle $\Delta\psi$ can be converted to direction cosine matrix \mathbf{C}_b^v [26], as mentioned in III.B.

TABLE IV
RMSE OF THE SPEED SEQUENCES WITH DIFFERENT VEHICLE LOADS

Loads	M1		M2	
	Raw [m/s]	Filtered [m/s]	Raw [m/s]	Filtered [m/s]
Two People	0.389	0.259	0.441	0.296
Three People	0.423	0.304	0.443	0.307
Four People	0.410	0.260	0.472	0.311

TABLE V
RMSE OF THE SPEED SEQUENCES IN NEW SCENES

Sequences	Raw [m/s]	Filtered [m/s]
Seq. 1	0.472	0.336
Seq. 2	0.443	0.333
Seq. 3	0.377	0.277

Seq. n denotes different test sequences.

pitch mounting angle might lead to the change of the gravity component on different axes of the accelerometer. The error of the estimated pitch mounting angle is typically within 0.005° [26] for low cost MEMS IMU, while the error of the estimated heading mounting angle is much larger. As long as the accuracy of the estimated mounting angles is within 2° , the OdoNet is capable of adapting these factors easily and maintains reasonable accuracy of the regressed speed.

2) Impact of Vehicle Loads

Different loads might affect the dynamic characteristics of the vehicle in some cases, and thus the vehicle loads might affect the precision of OdoNet to some extent. A special experiment was carried out to evaluate the impact of the vehicle loads. We drove the vehicle on the same road on a round trip with different loads (i.e. with a different number of passengers), and the length of the road was about 6 km. Three tests were conducted, and the number of passengers was two, three, and

four, respectively. The RMSE of the regressed speed in each test is displayed in Table IV. No notable difference is found between these three tests, not only from the raw-speed error but also from the filtered-speed error, which illustrates that the impact of the vehicle loads is negligible.

3) Impact of Road Conditions

It is obvious that different road conditions might affect the motion of the vehicle, and thus might affect the precision of the OdoNet. Even though our dataset has covered various roads in Wuhan city, we should evaluate the OdoNet with data collected in totally new scenes. Three new IMU sequences were used in this test. These testing sequences were collected in a totally strange scene by a new IMU, specifically, another industrial district in Wuhan city. As shown in Table V, the RMSE of the regressed vehicle speed on these three sequences are very close, mainly because of the same testing scene. Moreover, the results show nearly the same accuracy compared to previous results in Table II, which demonstrates the superior generalization capability of the OdoNet to different road conditions. Moreover, the experiment results illustrate that the OdoNet is robust to different IMU individuality once again.

E. Navigation Results

In this part, we will present the results of the GNSS/INS integrated navigation using three different aiding modes, including the NHC-only, the pseudo-odometer aiding, and the wheeled odometer aiding. By comparing the navigation results of these three processing modes, we can judge the precision of our proposed method. When applying odometer aiding, NHC is always applied simultaneously. The difference between the NHC-only mode and the odometer mode is mainly in the forward direction of the vehicle instead of the lateral and vertical direction, and thus the horizontal error is adopted to evaluate the positioning accuracy in the following sections.

In addition, ZUPT and ZARU are used only when the odometer is used, because the zero-velocity state is determined by the odometer speed. The standard deviation of pseudo-odometer observation is set to 0.3 m/s, according to the RMSE of the filtered speed in Table II, while 0.1 m/s for real wheeled odometer according to its actual precision, as mentioned in III.C. The GNSS positioning used in the experiments were derived from ZED-F9P working in Real Time Kinematic (RTK) mode. Only fixed solutions were employed in the integrated navigation system to ensure the quality of the GNSS positioning.

1) Low-speed Scene

Due to narrow roads and complex traffic conditions, the vehicle traveled very slowly around the university campus. In this low-speed scene, there was intensive coverage of tall trees along the roads, which led to long GNSS outages occasionally. The longest three GNSS outages lasted 36, 56, and 131 seconds, respectively. Fig. 10 shows the filtered regressed speeds of M1 and their error. As we can see, the highest speed is only about 10 m/s, and most of the filtered speeds are precise, except for some gross error exceeding 1 m/s.

Fig. 11 shows the cumulative distribution function (CDF) of the two testing modules with three processing modes. Due to higher precision, the wheeled odometer improves the precision of navigation significantly. The pseudo-odometer also shows great improvement compared to the NHC-only mode. The

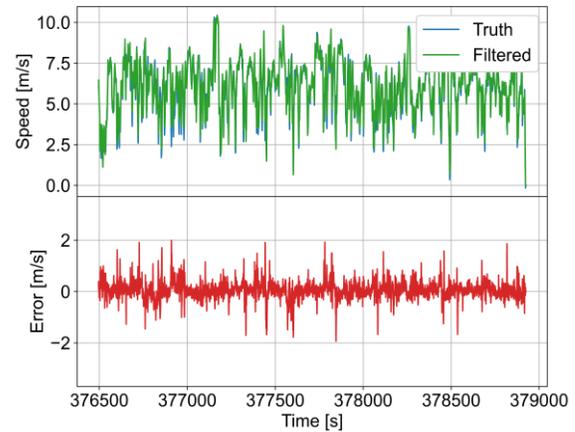


Fig. 10. The filtered regressed speed of the testing M1 and its error in the low-speed testing scene.

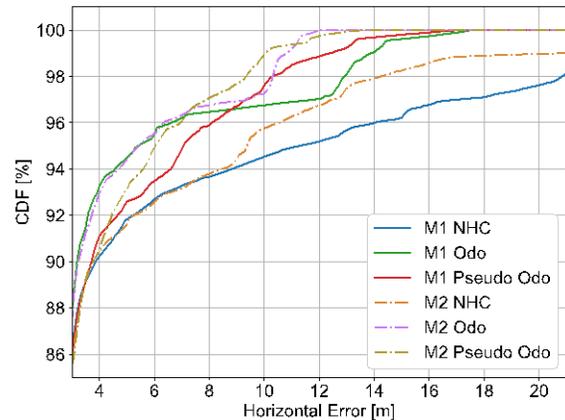


Fig. 11. The cumulative distribution function of horizontal error in the low-speed testing scene. The M1 and M2 represents two different INS-Probe modules in testing dataset. The Odo denotes the wheeled odometer aiding. The Pseudo Odo denotes the pseudo-odometer aiding from OdoNet.

precision of the pseudo-odometer is extremely close to the real wheeled odometer in this low-speed scene. However, as we can see, the pseudo-odometer only takes effects during long GNSS outages, which means that the integrated navigation system might have a large drift without pseudo-odometer aiding. The most important thing is to reduce large drift in vehicle navigation, and the pseudo-odometer and the wheeled odometer can play the same role on this issue.

2) High-speed Scene

The high-speed scene is the Second Ring Road of Wuhan city, without any traffic light. There are four long tunnels during the travels, with some other GNSS-challenged environments. The longest three GNSS outages lasted 73, 80, and 156 seconds, respectively. The filtered regressed speed and its error are shown in Fig. 12. The vehicle traveled at a very high speed, except when traffic jams occurred. As mentioned in IV. C, the OdoNet shows poor precision when the speed exceeds 20 m/s; the regressed speed contains a large error at 446000s in Fig. 12, for example. Besides, the RMSE of both the raw and the filtered speed in the high-speed scene are larger than that of the other two scenes, as can be seen in Table II.

Fig. 13 shows the CDF of the horizontal error in the high-speed scene. The pseudo-odometer still outperforms the NHC-only mode, but cannot achieve the precision as the same as the real wheeled odometer. However, the pseudo-odometer

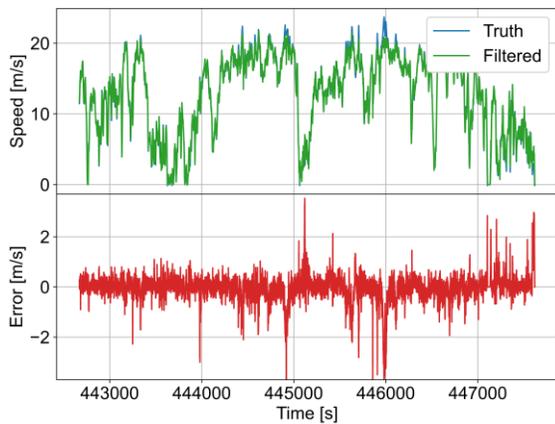


Fig. 12. The filtered regressed speed of the testing M1 and its error in the high-speed testing scene.

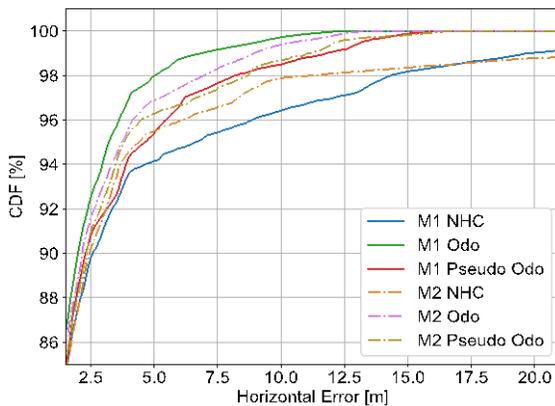


Fig. 13. The cumulative distribution function of horizontal error in the high-speed testing scene.

shows outstanding performance in reducing large drifts. As depicted in Fig. 13, 100% of the error is less than 17 m after applying the pseudo-odometer aiding or the wheeled odometer aiding. However, if only the NHC is used, the maximum error is far larger than 20 m when the vehicle travels through the long tunnels, where GNSS tends to have long outages. All in all, the pseudo-odometer and the wheeled odometer almost make the same contribution to reducing large positioning drifts.

3) Median-speed Scene

This scene is located in an industrial district of Wuhan city, with no tall buildings and few other vehicles. We drove the vehicle back and forth on these roads, and we often had to stop at the crossroads waiting for the traffic lights. As depicted in Fig. 14, the vehicle traveled steadily, without violent motions, either running around a fixed speed or staying stationary. The error of the filtered regressed speed is extremely small. The RMSE of the filtered speed is only 0.240 m/s and 0.249 m/s for the two testing modules respectively, which outperforms that in the other two scenes.

This scene is an open-sky environment, and thus GNSS can always achieve centimeter-level positioning. In this scene, we evaluated the performance of the integrated system by mimic GNSS outages [31]. We interrupted the GNSS positioning update in EKF for 60 seconds with a period of 180 seconds. During each outage, we picked up the maximum horizontal positioning error and then calculated the RMS of all the maximum errors in each outage. Three tests were conducted with a different start time of the outages. Table VI shows the

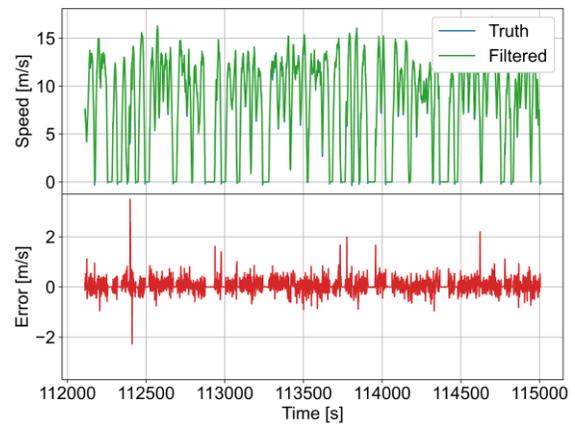


Fig. 14. The filtered regressed speed of the testing M1 and its error in the median-speed testing scene.

TABLE VI
THE RMS OF HORIZONTAL ERROR DURING MIMIC GNSS OUTAGES

Module	Aiding	Seq. 1 [m]	Seq. 2 [m]	Seq. 3 [m]	RMS [m]
M1	NHC	15.624	8.974	10.669	12.090
	Pseudo Odo	5.151	3.024	3.713	4.061
	Odo	4.018	2.273	2.780	3.111
M2	NHC	11.351	8.572	16.364	12.518
	Pseudo Odo	3.567	3.894	4.005	3.827
	Odo	3.367	2.752	3.359	3.172

Seq. n denotes three different outage sequences, which means that outages have different start time. Seq. 1 and Seq. 2 each has 13 outages, and Seq. 3 has 12 outages. The length of each GNSS outage is 60 seconds. The last column represents the RMS of the results of the three sequences.

TABLE VII
THE RMS OF HORIZONTAL ERROR FOR TWO MODULES

Module	M1 [m]	M2 [m]	RMS of M1 and M2 [m]	Percentage [%]
NHC	12.090	12.518	12.306	-
Pseudo Odo	4.061	3.827	3.946	68%
Odo	3.111	3.172	3.142	74%

The percentages represent the percent reduction of the RMSE compared to the NHC-only mode.

experiment results of these two testing modules in three processing modes. As we can see, compared to the NHC-only mode, the pseudo-odometer significantly improves the precision of the positioning, and the RMSE are reduced by 66.4% and 69.4% for the two testing modules respectively. In fact, ZUPT and ZARU also make contributions to the improvement, which benefits from the reliable and effective zero-velocity detector using the pseudo-odometer speed.

The experiment results in the median-speed scene are statistically significant, and thus we can obtain quantitative conclusions. More specifically, the RMSE of the two testing modules are listed in Table VII. Compared to the NHC-only mode, by using the pseudo-odometer, the horizontal error is reduced by about 68%, while the percentage for the wheeled

odometer is about 74%. The pseudo-odometer and the wheeled odometer almost yield the same accuracy. Consequently, the proposed pseudo-odometer can act as an alternative to the real wheeled odometer.

V. CONCLUSIONS

In this paper, we propose OdoNet, an untethered CNN-based deep pseudo-odometer model for vehicle GNSS/INS integrated navigation system, which is applicable for common navigation devices and even smartphones. Dedicated experiment results demonstrate that the OdoNet is robust enough to adapt to various changing conditions, including the IMU individuality, the vehicle loads, and the road conditions, while the IMU biases and the mounting angles can be compensated in data-cleaning procedure to mitigate their impacts. Navigation results indicate that by using the pseudo-odometer, the horizontal positioning error (60-second GNSS outages) is reduced by around 68% (compared to the NHC-only mode), which is very close to that of the hardware wheeled odometer. In conclusion, the proposed pseudo-odometer can significantly improve the robustness and accuracy of the integrated navigation system, and it is a reliable alternative to the hardware wheeled odometer.

VI. ACKNOWLEDGEMENT

This research is funded by the National Key Research and Development Program of China (No. 2020YFB0505803), and the National Natural Science Foundation of China (No. 41974024). The authors would also like to thank Dr. Xin Feng in our group for proofreading the paper.

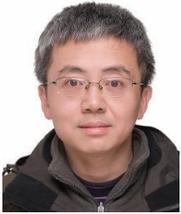
REFERENCES

- [1] I. Skog, P. Handel, J. O. Nilsson, and J. Rantakokko, "Zero-Velocity Detection—An Algorithm Evaluation," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 11, pp. 2657 – 2666, Nov. 2010.
- [2] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Trans. Robot. Automat.*, vol. 17, no. 5, pp. 731 – 747, Oct. 2001.
- [3] X. Niu, Y. Li, Q. Zhang, Y. Cheng, and C. Shi, "Observability Analysis of Non-Holonomic Constraints for Land-Vehicle Navigation Systems," *JGPS*, vol. 11, no. 1, pp. 80 – 88, Jun. 2012.
- [4] X. Niu, S. Nassar, and N. EL-SHEIMY, "An Accurate Land-Vehicle MEMS IMU/GPS Navigation System Using 3D Auxiliary Velocity Updates," *Navigation*, vol. 54, no. 3, pp. 177 – 188, 2007.
- [5] X. Yu et al., "AZUPT: Adaptive Zero Velocity Update Based on Neural Networks for Pedestrian Tracking," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1 – 6.
- [6] B. Wagstaff and J. Kelly, "LSTM-Based Zero-Velocity Detection for Robust Inertial Navigation," *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1 – 8, Sep. 2018.
- [7] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Robust Data-Driven Zero-Velocity Detection for Foot-Mounted Inertial Navigation," *IEEE Sensors J.*, vol. 20, no. 2, pp. 957 – 967, Jan. 2020.
- [8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735 – 1780, 1997.
- [9] Brossard, Martin, Axel Barrau, and Silvere Bonnabel, "RINS-W: Robust inertial navigation system on wheels," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [10] S. Cortés, A. Solin, and J. Kannala, "DEEP LEARNING BASED SPEED ESTIMATION FOR CONSTRAINING STRAPDOWN INERTIAL NAVIGATION ON SMARTPHONES," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2018, pp. 1 – 6.
- [11] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU Double Integration," in *Computer Vision – ECCV 2018*, Cham, 2018, vol. 11217, pp. 641 – 656.
- [12] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, "A Bidirectional LSTM for Estimating Dynamic Human Velocities from a Single IMU," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Pisa, Italy, Sep. 2019, pp. 1 – 8.
- [13] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, "RNN-Aided Human Velocity Estimation from a Single IMU," *Sensors*, vol. 20, no. 13, p. 3656, Jun. 2020.
- [14] Chen, Changhao, et al. "Ionet: Learning to cure the curse of drift in inertial odometry," *Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1*, 2018.
- [15] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "OxIOD: The Dataset for Deep Inertial Odometry," *arXiv:1809.07491 [cs]*, Sep. 2018.
- [16] Herath, Sachini, Hang Yan, and Yasutaka Furukawa, "RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [17] M. Abolfazli Esfahani, H. Wang, K. Wu, and S. Yuan, "AbolDeepIO: A Novel Deep Inertial Odometry Network for Autonomous Vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 21, no. 5, pp. 1941 – 1950, May 2020.
- [18] J. P. Silva do Monte Lima, H. Uchiyama, and R. Taniguchi, "End-to-End Learning Framework for IMU-Based 6-DOF Odometry," *Sensors*, vol. 19, no. 17, p. 3777, Aug. 2019.
- [19] Clark, Ronald, et al. "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," *Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1*, 2017.
- [20] Shamwell, E. Jared, Sarah Leung, and William D. Nothwang, "Vision-aided absolute trajectory estimation using an unsupervised deep network with online error correction," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [21] E. J. Shamwell, K. Lindgren, S. Leung, and W. D. Nothwang, "Unsupervised Deep Visual-Inertial Odometry with Online Error Correction for RGB-D Imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020.
- [22] Han, Liming, et al. "Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [23] Velas, Martin, et al. "CNN for IMU assisted odometry estimation using velodyne LiDAR," *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018.
- [24] C. Li, S. Wang, Y. Zhuang, and F. Yan, "Deep Sensor Fusion between 2D Laser Scanner and IMU for Mobile Robot Localization," *IEEE Sensors J.*, pp. 1 – 1, 2020.
- [25] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "OriNet: Robust 3-D Orientation Estimation With a Single Particular IMU," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 399–406, Apr. 2020.
- [26] Q. Chen, Q. Zhang, and X. Niu, "Estimate the Pitch and Heading Mounting Angles of the IMU for Land Vehicular GNSS/INS Integrated System," *IEEE Trans. Intell. Transport. Syst.*, pp. 1 – 13, 2020.
- [27] E.-H. Shin, "Estimation techniques for low-cost inertial navigation," Ph.D. dissertation, Dept. Geomatics Eng., Univ. Calgary, Calgary, AB, Canada, 2005.
- [28] D. Groves, *Principles of GNSS, Inertial and Multisensor Navigation Systems*. London, U.K.: Artech House, 2008.
- [29] A. LeNail, "NN-SVG: Publication-Ready Neural Network Architecture Schematics," *JOSS*, vol. 4, no. 33, p. 747, Jan. 2019.
- [30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.
- [31] Xiaoji Niu, C. Goodall, S. Nassar, and N. El-Sheimy, "An Efficient Method for Evaluating the Performance of MEMS IMUs," in *2006 IEEE/ION Position, Location, and Navigation Symposium*, Coronado, CA, 2006, pp. 766 – 771.

- [32] Ramachandran P, Zoph B, Le Q V, "Searching for activation functions". *arXiv preprint arXiv:1710.05941 [cs]*, 2017.
- [33] L. Chang, X. Niu, and T. Liu, "GNSS/IMU/ODO/LiDAR-SLAM Integrated Navigation System Using IMU/ODO Pre-Integration," *Sensors*, vol. 20, no. 17, p. 4702, Aug. 2020.



Hailiang Tang received the B.E. and M.E. degrees from Wuhan University, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in communication and information system with the GNSS Research Center, Wuhan University. His current research interests include GNSS/INS integration, deep learning, visual SLAM, and autonomous robotics system.

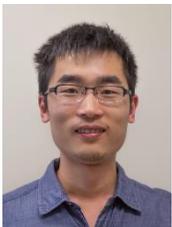


Xiaoji Niu received his bachelor's and Ph.D. degrees from the Department of Precision Instruments, Tsinghua University, in 1997 and 2002, respectively. He is currently a Professor with the GNSS Research Center, Wuhan University, China. He did post-doctoral research with the University of Calgary and worked as a Senior Scientist in SiRF Technology Inc. He has published more than 90 academic papers and own 28 patents. He leads a multi-sensor navigation

group focusing on GNSS/INS integration, low-cost navigation sensor fusion, and its new applications.



Tisheng Zhang is an associate professor in GNSS Research Center at Wuhan University, China. He holds a B.Sc. and a Ph.D. in Communication and Information System from Wuhan University, Wuhan, China, in 2008 and 2013, respectively. From 2018 to 2019, he was a PostDoctor of the HongKong Polytechnic University. His research interests focus on the fields of GNSS receiver and multi-sensor deep integration.



You Li is a Professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, China. He received Ph.D. degrees from Wuhan University and University of Calgary in 2015 and 2016, respectively, and a BEng degree from China University of Geoscience (Beijing) in 2009. His research focuses on positioning and motion-tracking techniques and their uses in IoT devices, smartphones, robots, and cars. He has

hosted/participated in five national research projects, and co-published over 70 academic papers, and has over 20 patents pending. He serves as an Associate Editor for the IEEE Sensors Journal, a committee member at the IAG unmanned navigation system and ISPRS mobile mapping working groups.



Jingnan Liu, member of Chinese Academy of Engineering, professor, Ph. D supervisor. He is an expert in geodesy and surveying engineering with the specialty of GNSS technology and applications. He has been engaged in the research of geodetic theories and applications, including national coordinate system establishment, GNSS technology and software development, as well as large project implementation. Over the past few decades, he has been engaged in the research of

geodetic theories and applications. And so far he has published more than 150 academic papers and supervised more than 100 postgraduates.