

Multitask Deep Neural Network for IMU Calibration, Denoising and Dynamic Noise Adaption for Vehicle Navigation

Frieder Schmid ^{1*} and Jan Fischer ^{1*}

¹ ANavS GmbH, Munich, Germany

* Correspondence: frieder.schmid@anavs.de, jan.fischer@anavs.de

Abstract: In intelligent vehicle navigation efficient sensor data processing and accurate system stabilization is critical to maintain robust performance, especially when GNSS signals are unavailable or unreliable. Classical calibration methods for Inertial Measurement Units (IMUs), such as discrete and system-level calibration, fail to capture time-varying, nonlinear, and non-Gaussian noise characteristics. Likewise, Kalman filters typically assume static measurement noise levels for Non-Holonomic Constraints (NHC), resulting in suboptimal performance in dynamic environments. Furthermore, zero-velocity detection plays a vital role in preventing error accumulation by enabling reliable zero-velocity updates during motion stops, but classical thresholding approaches often lack robustness and precision. To address these limitations, we propose a novel multitask deep neural network (MTDNN) architecture that jointly learns IMU calibration, adaptive noise level estimation for NHC, and zero-velocity detection solely from raw IMU data. This shared-encoder design is utilized to minimize computational overhead, enabling real-time deployment on resource-constrained platforms such as Raspberry Pi. The model is trained using post-processed GNSS-RTK ground truth trajectories obtained from both a proprietary dataset and the publicly available 4Seasons dataset. Experimental results confirm the proposed system's superior accuracy, efficiency, and real-time capability in GNSS-denied conditions.

Keywords: Multitask; Neural Network; IMU; Non-Holonomic-Constraint; Standstill;Zero-Velocity

Received:

Revised:

Accepted:

Published:

Citation: Schmid, F.; Fischer, J. Multitask Deep Neural Network for IMU Calibration, Denoising and Dynamic Noise Adaption for Vehicle Navigation. *Journal Not Specified* **2025**, 1, 0. <https://doi.org/>

Copyright: © 2025 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reliable and robust vehicle navigation in GNSS-degraded environments, such as tunnels, urban canyons, or indoor parking garages, is essential for autonomous driving and ADAS, as noted by Reid et al. [1]. While techniques like Precise Point Positioning (PPP) offer high accuracy, their performance during GNSS outages depends heavily on the quality of complementary sensors like the Inertial Measurement Unit (IMU) [2].

Automotive IMUs are typically low-cost MEMS sensors affected by noise, scale factor errors, and long-term bias drift. To mitigate these effects, GNSS/INS systems incorporate auxiliary modules such as IMU calibration, zero-velocity detection, and adaptive noise level estimation for velocity constraints. These modules can be implemented through classical signal processing methods or learning-based techniques. However, traditional approaches often require extensive manual tuning and prior knowledge of the system dynamics, which limits adaptability and robustness.

In this work, we combine the three often independent deep learning architectures into a unified Multitask Deep Neural Network (MTDNN) architecture: one for IMU denoising and calibration (based on Brossard et al. [3]), one for adaptive noise level estimation within

Kalman filtering (based on Brossard et al. [4]), and a third for zero-velocity detection (based on Brossard et al. [5]). While originally developed for distinct purposes, these models are integrated into a multitask framework tailored for vehicular use.

Our architecture features a shared-encoder that processes raw IMU data and three task-specific decoder branches. This design enables shared parameter learning, reduces computational load, making the system suitable for deployment on resource-constrained platforms. The model is trained using only IMU input and supervised using ground truth trajectories derived from GNSS-RTK and IMU fusion. We evaluate our method on the 4Seasons, and an in-house dataset. The result is a compact, multipurpose, real-time capable Multitask Deep Neural Network, designed to enhance navigation performance and robustness in challenging scenarios.

This work was conducted within the EU-funded DREAM project (Driving-aids powered by E-GNSS, AI & Machine Learning, <https://dream-project-eu.com/>), which aims to enhance localization and perception for public transport through robust, AI-based navigation modules. Our contribution supports this objective by enabling high-accuracy positioning even under degraded GNSS conditions. Both the code and the in-house dataset will be made publicly available at <https://github.com/anavsgmbh/MTDNN>.

The remainder of this paper presents related work (Section 2), details the proposed multitask network and training setup (Section 3), evaluates its performance (Section 4), and concludes with a discussion and outlook (Sections 5–6).

2. Related Work

2.1. IMU Calibration and Noise Compensation

IMUs are widely used in vehicle navigation, but their accuracy is limited by non-Gaussian, temporally correlated noise. Classical calibration methods can be divided into discrete calibration—often involving high-precision hardware like turntables—and system-level approaches, typically based on Kalman filters (Huang et al. [6]; Trinh et al. [7]). These methods either fail to capture time-dependent characteristics or struggle with nonlinear and non-Gaussian random errors.

To overcome these limitations, several deep learning approaches have been proposed. Brossard et al. [3] introduced a CNN-based architecture that removes both systematic and random IMU errors. This line of research was extended by Liu et al. [8] with LGC-Net and by Chao et al. [9] with TinyGC-Net, both targeting deployment on resource-constrained platforms through reduced model size. Yuan and Wang [10] proposed a simple self-supervised, non-iterative calibration technique for multi-sensor setups. Xu et al. [11] presented a dynamic receptive field mechanism to capture long-term temporal dependencies in IMU signals.

2.2. Zero-Velocity Detection

Zero-Velocity Updates are a well-established technique for mitigating drift in inertial navigation systems. Wahlström and Skog [12] provide a comprehensive overview of classical threshold-based methods and their limitations. In contrast, more recent research has shifted towards machine learning-based detection. For example, Wagstaff and Kelly [13] and Zhang et al. [14] developed LSTM-based networks for identifying zero-velocity intervals in foot-mounted IMUs.

For wheeled vehicles, Li et al. [15] and Kilic et al. [16] combined learned standstill detection with Kalman and factor graph-based filtering, enhancing robustness in GNSS-challenged environments. RINS-W, introduced by Brossard et al. [5], employs a LSTM network to detect both zero-velocity and zero-angular-rate constraints, which are incorporated as pseudo-measurements in a Kalman filter.

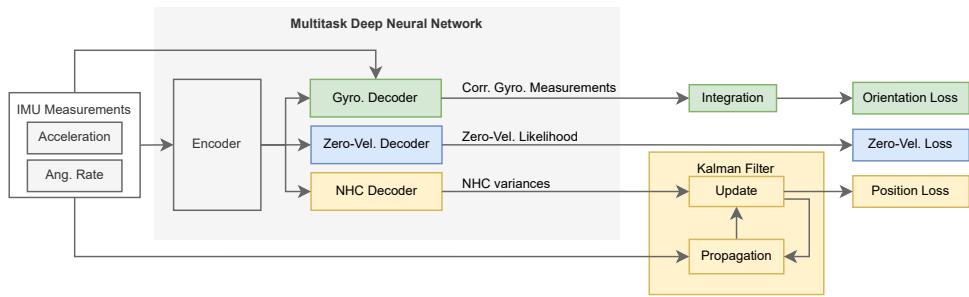


Figure 1. Sequence of the training process of an epoch of the MTDNN.

2.3. Noise Level Adaptation for Non-Holonomic Constraints

Non-holonomic constraints (NHCs) are commonly used in GNSS/INS integration to constrain vehicle motion, for wheeled vehicles this means enforcing zero lateral and vertical velocity. Early integration approaches were presented by Dissanayake et al. [17] and Jiang et al. [18], and later embedded into Kalman filter formulations (Niu et al. [19]). In order to adapt to constraint violations the NHC are often introduced with a noise level, which has to be manually tuned. To overcome limitations, learning-based alternatives have been proposed. Xu et al. [20] predicted IMU-based position increments and applied NHC filtering during GNSS outages. SdoNet, introduced by Wang et al. [21], estimates vehicle velocity directly from IMU input and dynamically adjusts measurement noise via a learned adapter module. Brossard et al. [4] combined a NHC noise adaption network with Kalman filtering to enable inertial dead reckoning in GNSS-denied environments. Xiao et al. [22] enhanced confidence estimation for NHCs using a residual attention mechanism.

3. Methodology

3.1. System Overview

In GNSS-degraded environments, robust navigation relies heavily on multiple separate auxiliary modules such as IMU denoising, zero-velocity detection, and adaptive handling of non-holonomic constraints (NHC). We propose to replace this fragmented architecture with a unified multitask deep learning model that simultaneously addresses all three components. The key idea is to exploit the fact that all three tasks share a common input domain: the raw IMU measurements. These measurements contain rich temporal structure and motion-related patterns that can be learned and generalized across multiple tasks. For instance, detecting zero-velocity and identifying gyro bias both rely on recognizing low-magnitude, stable IMU signals, suggesting that shared features can benefit both tasks when learned jointly. The overall sequence of the training process for one epoch of the overall system is illustrated in Figure 1.

Our approach uses a single encoder architecture to extract meaningful features from the 6D IMU input (3D acceleration + 3D angular rate), which has to be run only once per epoch. These shared features are then passed to three lightweight decoder branches for:

- correcting gyroscope measurements (green pipeline in Figure 1),
- estimating noise levels for NHC (blue pipeline in Figure 1),
- and predicting the likelihood of vehicle standstill (yellow pipeline in Figure 1).

By consolidating the tasks into a single network, we reduce the number of trainable parameters and eliminate the need for redundant processing. This makes the system well-suited for deployment on resource-constrained platforms like a Raspberry Pi, without sacrificing performance.

Implementation-specific architectural choices, training strategies, and downstream integration with the Kalman filter are described in the following subsections.

3.2. Network Architecture

The network is defined by the four key components:

Encoder: Designed as a four-layer Temporal Convolutional Network (TCN) based on work done by Brossard et al. ([3]). It uses 1D convolutions with increasing dilation and channel dimension to capture temporal dependencies efficiently. All layers apply kernel size of 3, GELU activation, batch normalization, dropout, and residual connections to stabilize learning and model multi-scale motion.

Each decoder consists of one fully connected layer, making them lightweight and computationally efficient.

Gyroscope Decoder: Outputs corrected angular velocities $\hat{\omega}_n$ by adding the correctional term δ_n estimated from the encoder outputs to the raw gyroscope measurements ω_n^{IMU} .

$$\hat{\omega}_n = \hat{\mathbf{C}}_\omega \omega_n^{\text{IMU}} + \delta_n, \quad (1)$$

with $\hat{\mathbf{C}}_\omega = \hat{\mathbf{S}}_\omega \hat{\mathbf{M}}_\omega \in \mathbb{R}^{3 \times 3}$ accounting for gyroscope axis-misalignment and scale factors (optimized only during training) and δ_n including the static and time-varying gyroscope bias as well as noise (estimated online).

NHC Noise Level Decoder: Estimates dynamic variances for the NHC using one fully connected network with softsign activation. Its output $[z_n^{\text{lat}}, z_n^{\text{up}}] \in \mathbb{R}^2$ scales the baseline covariances $[\sigma_{\text{lat}}^2, \sigma_{\text{up}}^2] \in \mathbb{R}^2$ via a logarithmic factor, enabling a wide numerical range for the variance $\mathbf{N}_{\text{NHC}} \in \mathbb{R}^{2 \times 2}$ used in the Kalman filter.

$$\mathbf{N}_{\text{NHC}} = \text{diag}\left(\sigma_{\text{lat}}^2 10^{\beta \tanh(z_n^{\text{lat}})}, \sigma_{\text{up}}^2 10^{\beta \tanh(z_n^{\text{up}})}\right), \text{ with } \beta \in \mathbb{R}_{>0} \quad (2)$$

Zero-velocity Decoder: Produces a scalar activation value between 0 and 1 indicating zero-velocity likelihood at each timestep using a sigmoid activation function. This signal can trigger zero-velocity updates or influence the filter during motion stops.

3.3. Training Objectives

The training objective combines three loss functions, each corresponding to one of the multitask network's outputs. These losses are computed independently and optimized using separate learning schedules and optimizers due to differences in convergence rates and computational complexity. The orientation and zero-velocity losses are computed every epoch, while the position loss is computed at scheduled intervals (every 10th epoch) to manage training time. All three losses backpropagate through their task-specific decoder and the shared encoder. Hence, the shared parameters are updated by every task and the optimization is interleaved.

The gyroscope correction branch is supervised using a rotation increment loss that compares integrated angular velocities to orientation increments derived from ground truth orientation data. The loss formulation, used from Brossard et al. [3], computes the error between predicted and ground truth incremental rotation matrices. The rotational difference ΔR is mapped to the Lie algebra $\mathfrak{SO}(3)$ via the logarithm map, and an Huber loss ($\rho(\cdot)$) is applied to the resulting vector:

$$L_{\text{rot}} = \rho(\log(\Delta R))$$

The NHC decoder is supervised by evaluating the impact of its predicted lateral and vertical variances $\mathbf{N}_{\text{NHC}} \in \mathbb{R}^{2 \times 2}$ in a Kalman filter. The Kalman filter, based on the formulation from Brossard et al. [4] includes position, velocity, orientation, and IMU biases in its state vector, and uses IMU-based prediction with NHC updates. We extend the filter

to also include odometry as a third velocity measurement. The resulting filtered trajectory x_{KF} is compared to the ground truth trajectory x_{GT} using the Huber loss:

$$L_{pos} = \rho(x_{KF} - x_{GT})$$

The zero-velocity detection decoder is trained as a binary classification task. The output activation $s \in [0, 1]$, representing the likelihood of a standstill (zero velocity), is compared against binary ground truth labels $y \in \{0, 1\}$. These are derived by thresholding the body-frame velocity magnitude. To penalize incorrect predictions and focus learning on harder samples, a weighted focal loss is used with focusing parameter $\gamma = 2.0$. The loss is based on the binary cross-entropy and modulated by a factor that down-weights well-classified examples. Additionally, class-dependent weighting α is introduced to penalize false positives more strongly:

$$L_{zero-velocity} = \alpha(1 - p_t)^\gamma \cdot \text{BCE}(s, y), \quad \text{with } p_t = \exp(-\text{BCE}(s, y)).$$

Here, α is a class-dependent weight set to 1.0 for zero-velocity samples ($y = 1$) and 10.0 for motion samples ($y = 0$) to stronger penalize false positives.

3.4. Ground Truth Generation

Ground truth supervision relies on GNSS-RTK/INS fused data, including positions, orientations, and body-frame velocities. Two datasets are used, namely the 4Seasons [28] and an ANavS in-house dataset. Both datasets include recordings under varying environmental conditions—such as different temperatures and days or months between recordings. The ANavS dataset, in particular, which was collected using a Epson M-G365 IMU, combined with the vehicle’s internal odometer and a Septentrio Mosaic X5 GNSS receiver, includes a broad range of driving dynamics and motion patterns. Ground truth trajectories were generated by a proprietary GNSS-RTK multi-sensor fusion module from ANavS GmbH.

For the gyroscope correction and NHC noise level estimation tasks, ground truth signals such as orientation increments and positions are directly obtained from the fused trajectories. Zero-velocity labels are derived by thresholding the body-frame velocity magnitude below 0.1 km/h. Both datasets are segmented into 1-minute sequences, yielding approximately 100 samples per dataset. The amount of training data is considered adequate for this study. The main reason for this assumption is that the proposed network is designed to be compact, which reduces the amount of data required to fit the hypothesis class without overfitting. Furthermore, the proposed design builds upon prior architectures that were trained on comparable or even smaller data volumes. For the evaluation done in Chapter 4, 20% of the each dataset is used as validation data. The validation data is ensured to also contain a range of driving dynamics and motions patterns. Each sequence is resampled to a fixed rate of 100–125 Hz and further pre-processing is applied, like removal of segments without reliable trajectory information.

4. Results

For evaluation of the proposed framework, each decoder is analyzed individually and in combination. Table 1 summarizes the quantitative performance across the full 4Seasons and ANavS datasets, whereas Figure 2 illustrates a representative result on a single sequence from the 4Seasons dataset to provide qualitative insight into the network behavior. The gyroscope denoising/calibration is evaluated against raw IMU open-loop attitude, following Brossard et al. [3]. A Kalman filter with additional sensors like GNSS may estimate gyroscope biases, but since the network estimates the biases from the IMU data

alone, such a comparison could be considered rigged. The gyroscope correction decoder demonstrates that the network successfully learns to reduce cumulative orientation drift without explicit knowledge of the absolute attitude. As shown in Figure 2a, the corrected signals exhibit consistent alignment with the ground truth and remain notably stable during standstill phase, an indication that the network captures both dynamic and quasi-static IMU characteristics. In this particular situation, the pitch angle drift during standstill phase of the ground truth clearly shows the limitation of the used ground truth. Quantitatively, 4Seasons shows substantial improvement, with yaw RMS error reduced by 90.4% and its 95th percentile by 87.0%. Similarly roll and pitch improve by 27.5–49.1%. On the ANavS dataset, yaw RMS decreases by 38.7%, confirming robustness under varied motion patterns. These results indicate that the network not only corrects for dynamic motion bias but also attenuates long-term drift, especially in heading.

The zero-velocity decoder detects standstill phases with high fidelity, showing high agreement with the ground truth in Table 1. In the exemplaric case (Fig. 2b), its activation pattern is visibly narrower, reflecting increased sensitivity and reduced detection delay. On 4Seasons, it achieves 94.7% precision and 95.9% recall while the results on the ANavS dataset yield even higher precision (98.3%) but slightly reduced recall (87.3%), producing a still-robust F1-score of 92.4%. This precision focused profile favors filter consistency since false positives during motion can significantly degrade the filter, whereas missed detections delay updates but do not introduce direct errors. A comparison to a threshold-based IMU standstill detector is not included because adequate performance of such a threshold based detector typically requires extensive, platform-specific tuning (mounting, vibration, low-speed creep), which conflicts with the objective of minimizing manual tuning. Comparative evaluations against classical gyroscope denoising/calibration and KF-based bias estimation are left to future work and constitute a limitation of the present study.

The NHC variance decoder is compared against a tuned static diagonal NHC covariance within the same KF. The results (Fig. 2b) show that the network successfully estimates context-aware constraint variances without requiring any manual tuning or heuristic thresholding. Compared to the use of static and strongly constraining NHC variances, the learned model reduces RMS position error on the 4Seasons dataset by 29.9% and the 95th percentile error by 25.7%. On the ANavS dataset, the proposed model also achieves clear improvements: RMS position error is reduced by 19.4%, and the 95th percentile error drops by 10.8%. These results confirm that the adaptive variance decoder provides robust results across motion profiles, without requiring manual tuning.

Finally, the benefit of jointly learning all three tasks is evident in Figure 2c. Compared to the raw IMU setup with static NHC (red) and the gyroscope-corrected variant still relying on static NHC (green), the full multitask system (blue) achieves the best performance by integrating all corrections. Furthermore the proposed multitask framework achieves even greater parameter efficiency compared to Brossard et al. ([3] and [4]), by incorporating residual connections into the shared encoder. While Brossard et al. use over 40,000 parameters for gyroscope and NHC noise level network, our updated model requires only 12,293 in total—including all three decoders. The added tasks for zero-velocity detection and NHC noise level estimation introduce minimal overhead (decoders total: 393 parameters). In contrast, RINS-W [5] still uses around 370,000 parameters for zero-velocity detection alone. Furthermore, we evaluated the model’s runtime performance on a Raspberry Pi 5 and confirmed that it achieves inference rates significantly exceeding the 125 Hz IMU sampling frequency, demonstrating its suitability for real-time embedded deployment.

Table 1. Evaluation on 4Seasons and ANavS datasets. For each decoder task, the performance of the baseline and the proposed MTDNN method is compared.

Metric	Axis	4Seasons dataset		ANavS dataset		Unit
		Base	Proposed	Base	Proposed	
Task 1: NHC Noise Adapter – 3D Position Error (after Kalman filtering)						
RMS		Static	MTDNN	Static	MTDNN	[m]
95 Perc.		36.79	25.79	19.10	15.39	[m]
		81.05	60.24	45.40	40.49	[m]
Task 2: Gyroscope Denoiser – Orientation Error						
RMS	Roll	Raw	MTDNN	Raw	MTDNN	[°]
	Pitch	1.82	1.32	1.27	1.13	[°]
	Yaw	1.52	0.89	1.43	0.97	[°]
95 Perc.	Roll	7.09	0.68	3.69	2.26	[°]
	Pitch	4.57	2.98	2.70	2.42	[°]
	Yaw	3.46	1.76	2.87	1.99	[°]
		11.95	1.55	8.67	5.88	[°]
Task 3: Zero-Velocity Detection – Classification Metrics						
Precision			0.9467		0.9825	[-]
Recall			0.9595		0.8726	[-]
F1-Score			0.9531		0.9243	[-]

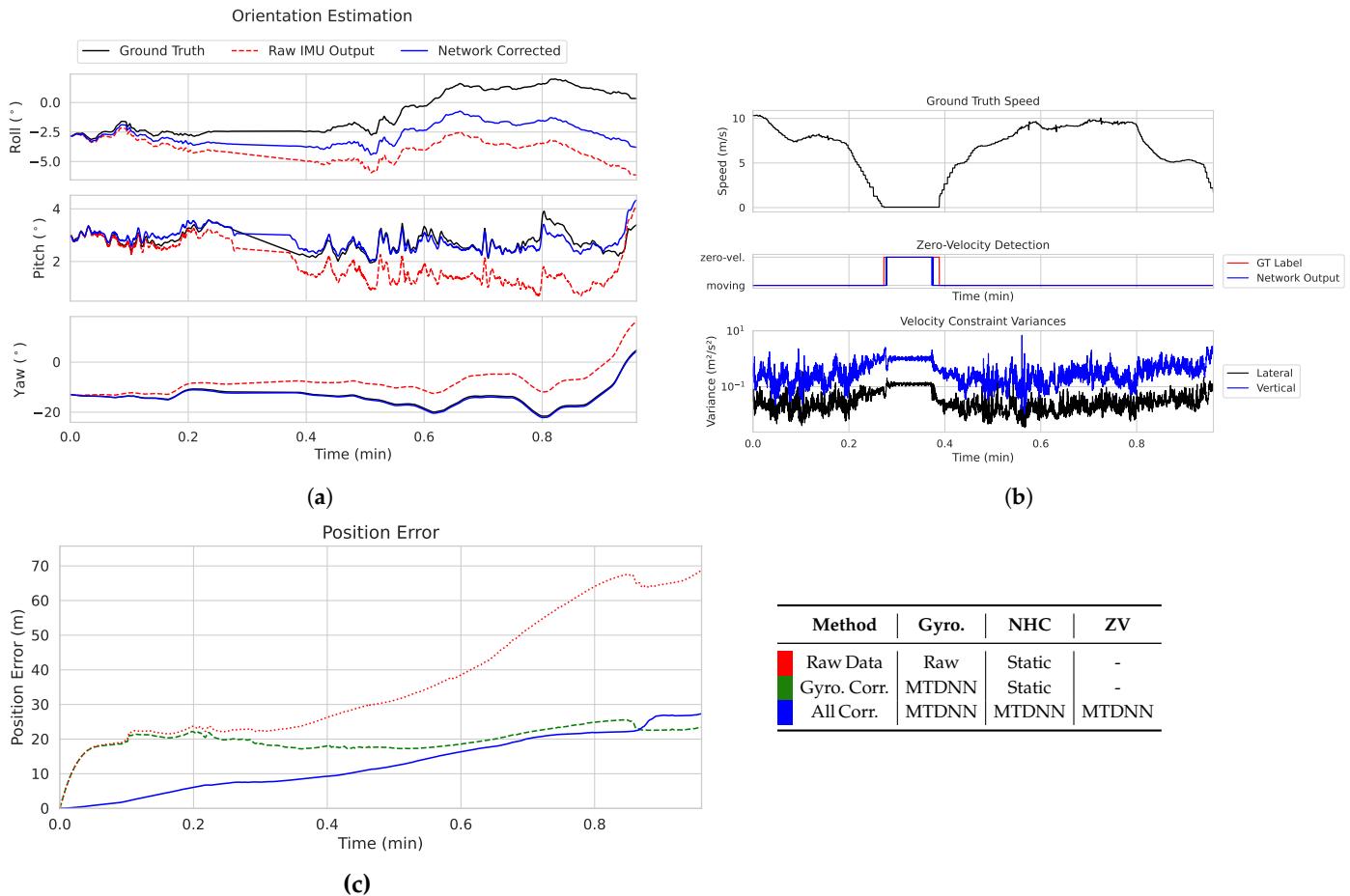


Figure 2. Results from the proposed MTDNN evaluated on a exemplary sequence of the 4Seasons dataset: (a) Orientation estimation using the IMU measurements before and after correction by the gyroscope decoder. (b) Results from the zero-velocity detector and the variances estimated by the NHC decoder in relation to the ground truth speed. (c) Resulting scalar position errors for raw IMU, gyro-corrected IMU and fully corrected output.

5. Discussion

The proposed MTDNN effectively unifies IMU calibration, adaptive NHC noise level estimation and zero-velocity detection into a compact, real-time-capable module. By exploiting shared temporal structures in raw IMU data, the shared encoder learns transferable features that enhance all three tasks. This multitask setup improves lowers computational cost while still achieving real-time inference on a Raspberry Pi. Quantitative results across both datasets confirm that each decoder contributes meaningfully to the overall navigation performance. The orientation drift is reduced through gyroscope correction, velocity errors are minimized via the NHC noise level estimation and the zero-velocity detector reliably enables zero-velocity-updates.

Limitations arise from the reliance on RTK-derived ground truth, which introduces potential label noise, especially in zero-velocity classification. Generalization to highly dynamic or off-road environments remains to be validated, as well as a detailed comparison of the resulting network to its three separate predecessors.

The model is well-suited for embedded navigation in GNSS-degraded settings, offering a lightweight alternative to multi-module pipelines. Future work will extend the architecture to multimodal inputs (e.g., cameras, encoders) and explore self-supervised learning to reduce dependence on high-quality ground truth.

6. Conclusions

We presented a multitask deep neural network that consolidates IMU calibration, zero-velocity detection and adaptive NHC noise level estimation into a single module. This design improves simplifies integration and enables real-time performance on devices with restricted computational power. Evaluation on two datasets shows consistent improvements in navigation performance. By replacing fragmented components with a unified network, we reduce tuning effort and deployment complexity. The proposed approach provides a scalable foundation for robust, real-time vehicle navigation.

Author Contributions: Conceptualization, F.S. and J.F.; methodology, F.S. and J.F.; software, F.S.; validation, F.S. and J.F.; formal analysis, F.S.; investigation, F.S.; resources, F.S.; data curation, F.S.; writing—original draft preparation, F.S.; writing—review and editing, F.S. and J.F.; visualization, F.S.; supervision, J.F.; project administration, F.S.; All authors have read and agreed to the published version of the manuscript.

Funding: The work is performed in the frame of the DREAM project, funded by the EUSPA as part of the Fundamental Elements Programme (contract number: EUSPA/GRANT/03/2022). <https://dream-project-eu.com/>

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The 4Seasons datasets used in this study is publicly available at <https://cvg.cit.tum.de/data/datasets/4seasons-dataset>. The in-house dataset will be available under <https://github.com/anavsgmbh/MTDNN>.

Acknowledgments: The authors would like to thank ANavS GmbH for providing infrastructure and internal dataset access used in this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reid, T., Bevly, D., & Houts, S. (2019). Localization Requirements for Autonomous Vehicles. *arXiv preprint arXiv:1906.01061*. <https://arxiv.org/abs/1906.01061>

2. Henkel P., Parimi S., Fischer J., Mittmann U. & Bensch R. (2024). Precise Positioning for Autonomous Driving in both Indoor and Outdoor Environments. *Proceedings of the 2024 International Technical Meeting of ION*, pp. 1074-1084. <https://doi.org/10.33012/2024.19534> 291
292

3. Brossard, M., Bonnabel, S., & Barrau, A. (2020). Denoising IMU Gyroscopes With Deep Learning for Open-Loop Attitude Estimation. *IEEE Robotics and Automation Letters*. <https://doi.org/10.1109/LRA.2020.3003256> 293
294

4. Brossard, M., Barrau, A., & Bonnabel, S. (2020). AI-IMU Dead-Reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4), 585–595. <https://doi.org/10.1109/TIV.2020.2980758> 295
296

5. Brossard, M., Barrau, A., & Bonnabel, S. (2020). RINS-W: Robust Inertial Navigation System on Wheels. *IEEE Robotics and Automation Letters*, 5(2), 362–369. <https://doi.org/10.1109/LRA.2019.2957011> 297
298

6. Huang, F., Wang, Z., Xing, L., & Gao, C. (2022). A MEMS IMU gyroscope calibration method based on deep learning. *IEEE Transactions on Instrumentation and Measurement*, 71, Article ID 3160538. <https://doi.org/10.1109/TIM.2022.3160538> 299
300

7. Trinh, X.-D., Le, M.-C., & Tran, N.-H. (2020). IMU Calibration Methods and Orientation Estimation Using Extended Kalman Filters. In *Recent Advances in Electrical Engineering and Related Sciences: Theory and Application* (pp. 541–551). Springer. 301
302

8. Liu, Y., Cui, J., & Liang, W. (2023). LGC-Net: Lightweight Gyroscope Errors Compensation Network for Effective Attitude Estimation. In *2023 42nd Chinese Control Conference (CCC)* (pp. 3326–3331). IEEE. <https://doi.org/10.23919/ccc58697.2023.10241087> 303
304

9. Chao, C., Zhao, J., Long, H., & Zhang, R. (2024). TinyGC-Net: an extremely tiny network for calibrating MEMS gyroscopes. *Measurement Science and Technology*, 35(11), 115109. <https://doi.org/10.1088/1361-6501/ad67f8> 305
306

10. Yuan, K., & Wang, Z. J. (2023). A Simple Self-Supervised IMU Denoising Method for Inertial Aided Navigation. *IEEE Robotics and Automation Letters*, 8(2), 944–950. <https://doi.org/10.1109/lra.2023.3234778> 308
309

11. Xu, B., Tang, Y., & Chen, Y. (2024). Deep Learning-Based IMU Errors Compensation with Dynamic Receptive Field Mechanism. In *Proceedings of ICARM 2024* (pp. 113–123). https://doi.org/10.1007/978-981-96-2216-0_10 310
311

12. Wahlström J. & Skog I. (2020). Fifteen Years of Progress at Zero Velocity: A Review. <https://arxiv.org/abs/2008.09208> 312

13. Wagstaff B. & Kelly J. (2018). LSTM-Based Zero-Velocity Detection for Robust Inertial Navigation. <https://arxiv.org/abs/1807.05275> 313
314

14. Zhang L., Chen B., Li H., & Liu Y. (2021). Deep neural network-based adaptive zero-velocity detection for pedestrian navigation system. *Electronics Letters*, 58(1), 27–31. <https://doi.org/10.1049/ell2.12339> 315
316

15. Li Q., Li K., & Liang W. (2023). A zero-velocity update method based on neural network and Kalman filter for vehicle-mounted inertial navigation system. *Measurement Science and Technology*, 34(4), 045110. <https://doi.org/10.1088/1361-6501/acabde> 317
318

16. Kilic C., Das S., Gutierrez E., Watson R. & Gross J. (2021). ZUPT Aided GNSS Factor Graph with Inertial Navigation Integration for Wheeled Robots. <https://arxiv.org/abs/2112.07176> 319
320

17. Dissanayake, G., Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (2001). The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation*. <https://doi.org/10.1109/70.964672> 321
322

18. Jiang, Z.-P., Lefeber, E., & Nijmeijer, H. (1998). Stabilization and tracking of a nonholonomic mobile robot with saturating actuators. 324
325

19. Niu, X., Nassar, S., & El-Sheimy, N. (2007). An accurate land-vehicle MEMS IMU/GPS navigation system using 3D auxiliary velocity updates. <https://doi.org/10.1002/j.2161-4296.2007.tb00403.x> 326
327

20. Xu, Y., Wang, K., Jiang, C., Li, Z., Yang, C., Liu, D., & Zhang, H. (2023). Motion-Constrained GNSS/INS Integrated Navigation Method Based on BP Neural Network. *Remote Sensing*, 15(1), 154. <https://doi.org/10.3390/rs15010154> 328
329

21. Wang, X., Zhuang, Y., Cao, X., Li, Q., Wang, Z., Cao, Y., & Chen, R. (2023). SdoNet: Speed Odometry Network and Noise Adapter for Vehicle Integrated Navigation. *IEEE Internet of Things Journal*, 10(21), 19328–19343. <https://doi.org/10.1109/jiot.2023.3294947> 330
331

22. Xiao, Y., Luo, H., Zhao, F., Wu, F., Gao, X., Wang, Q., & Cui, L. (2021). Residual Attention Network-Based Confidence Estimation Algorithm for Non-Holonomic Constraint in GNSS/INS Integrated Navigation System. *IEEE Transactions on Vehicular Technology*, 70(11), 11404–11418. <https://doi.org/10.1109/tvt.2021.3113500> 332
333

23. Wu, F., Luo, H., Jia, H., Zhao, F., Xiao, Y., & Gao, X. (2021). Predicting the Noise Covariance With a Multitask Learning Model for Kalman Filter-Based GNSS/INS Integrated Navigation. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–13. <https://doi.org/10.1109/tim.2020.3024357> 335
336

24. Su, T.-L., Zhang, L.-Y., Jin, X.-B., Kong, J.-L., Bai, Y.-T., & Ma, H.-J. (2022). A Dead Reckoning Method Based on Neural Network Optimized Kalman Filter. In *2022 IEEE International Conference on Unmanned Systems (ICUS)* (pp. 368–373). IEEE. <https://doi.org/10.1109/icus55513.2022.9986796> 338
339

25. Brossard, M., & Bonnabel, S. (2019). Learning Wheel Odometry and IMU Errors for Localization. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 291–297). <https://doi.org/10.1109/icra.2019.8794237> 341
342

26. Melendez-Pastor, C., Ruiz-Gonzalez, R., & Gomez-Gil, J. (2017). A data fusion system of GNSS data and on-vehicle sensors data for improving car positioning precision in urban environments. *Expert Systems with Applications*, 80, 28–38. <https://doi.org/10.1016/j.eswa.2017.03.018> 343

27. Wang, J.-H., & Gao, Y. (2010). Land Vehicle Dynamics-Aided Inertial Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 46(4), 1638–1653. <https://doi.org/10.1109/taes.2010.5595584> 344

28. Wenzel, P.; Wang, R.; Yang, N.; Cheng, Q.; Khan, Q.; von Stumberg, L.; Zeller, N.; Cremers, D. 4Seasons: A Cross-Season Dataset for Multi-Weather SLAM in Autonomous Driving. *In Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020. 345

346

347

348

349

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

350

351

352