2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), 5-8 August 2024, Kristiansand, Norway
DOI:10.1109/ICIEA61579.2024.10665134

A Two-Stage Classification Method for Improved Positioning using Low-Cost Inertial Sensors

Dariusz Maton[1]*, John Economou[1], David Galvão Wall[1], Irfan Khan[1], Robert Cooper[1], David Ward[2] and Simon Trythall[2]
[1]Centre for Defence Engineering, Cranfield University, Defence Academy of the UK, Shrivenham, SN6 8LA, UK
[2]BAE Systems, Rochester, United Kingdom
*Email: *d.maton@cranfield.ac.uk*

***Abstract*— The two-stage subtractive clustering Takagi-Sugeno (2SC-TS) method is proposed which completely replaces the established method of inertial navigation with classification models. The classifiers are designed by the subtractive clustering algorithm, an unsupervised learning method. The accuracy of the trajectories is compared against three competitive data-driven methods on three independent experimental datasets. The results show how 2SC-TS generates trajectories with approximately 20% lower positional error compared with the single-stage version (SC-TS), and halves the error produced by competitive deep learning methods. The proposed method may help improve the positioning of people and robots carrying low-cost inertial sensors.**

Keywords—IMU, Takagi-Sugeno model, inertial navigation.

I. INTRODUCTION

The need for accurate localization of assets under global navigation satellite system (GNSS) denial is in increasing demand. By 2050, it is estimated that 68% of people will live in urban areas and by 2030, one in eight will live in one of the world's 33 megacities [1]. Positioning in these environments can be challenging as multipath and signal attenuation distort or completely null range measurements [2]. A lack of accurate positioning of emergency responders can be particularly serious and, in some cases, fatal [3]. In autonomous systems, the vulnerability of GNSS to spoofing and jamming is problematic for the use of uninhabited systems used in patrol and infrastructure inspection [4], [5].

Other technologies offer attractive alternatives to GNSS. Signals from ultrawideband (UWB) systems operate close to or below the noise floor power and are thus difficult to detect and jam [6], however maximum operating ranges are limited to a few hundred metres [7]. Additionally, like other systems using wireless fidelity (Wi-Fi) [8], radio frequency identification (RFID) [9], photodiodes or visible fiducials [10], they rely on the pre-deployment of hardware into the environment beforehand. This prevents their application in situations where ad hoc tracking is required. Although visual odometry (VO) systems are capable of tracking natural features [11], these features are not always visible in the camera frame leading to sudden collapses in tracking. While thermal imaging cameras have been used for improving situational awareness in extreme conditions, they are vulnerable to saturation and sensitive to the presence of gases and aerosols which can inhibit the observation of thermal gradients [12].

In contrast, inertial sensors are autonomous sensors which do not require any supporting infrastructure for their operation [13]. Such sensors include accelerometers and gyroscopes, measuring the specific force and angular velocity of the sensor body frame, *b*, relative to the inertial reference frame, *i*, denoted $\boldsymbol{f}_{ib}^{b} \in \mathbb{R}^3$ and $\boldsymbol{\omega}_{ib}^{b} \in \mathbb{R}^3$, respectively. Inertial measurement units (IMUs) usually contain triads of mutually orthogonal accelerometers and gyroscopes and, thanks to micro-electromechanical systems (MEMS) manufacturing, low-cost, lightweight, and low power IMUs can be found in most modern electronic devices such as smartphones and augmented reality (AR) headsets. Unlike GNSS however, position cannot be estimated in an absolute sense but relative to a known initial position by a process of dead reckoning.

The conventional method of position estimation, strapdown inertial navigation, involves the double integration of gravity compensated accelerometer measurements. While this method works for high quality optical inertial sensors such as in [14], the noise and stability characteristics of low-cost IMUs are orders of magnitude worse resulting in rapid error accumulation and exponential drift in the estimated position. In response to this problem, researchers have used heuristics to bound the error growth. The zero-velocity update (ZUPT) retroactively corrects for the integral drift in the strapdown method however it assumes frequent periods of zero velocity are available [15]. For IMUs on aerial or marine robots, inside handsets or when travelling relative to a moving platform, zero velocity periods may be rare.

Data-driven methods have been researched to replace elements of the established methods with models generated by machine learning. Robust IMU Double Integration (RIDI) presented a shallow AI method for correcting the low-frequency errors in the linear acceleration estimates before double integration [16]. The method relied on a blend of the strapdown-derived estimate and the regressed velocity making it potentially vulnerable to environmental factors such as changes in sensor temperature. Inertial Odometry Network (IONet) presented a recurrent neural network model for estimating polar displacement vectors directly from windows of IMU data [17]. Deep learning was therefore used to completely avoid double integration in the strapdown method however input data was expressed in the sensor body frame which reduced its generalization ability as shown in [18]. Robust Inertial Navigation in the Wild (RoNIN) proposed a similar method to IONet but regressed planar velocity vectors from IMU measurements projected into the navigation frame beforehand [19]. Thus, only the acceleration to velocity stage was replaced by the deep learning model. While RoNIN and IONet aimed for competitive orientation-agnostic position tracking, [20] showed how similar tracking accuracy could be achieved with an interpretable model based on subtractive clustering and Takagi-Sugeno (SC-TS) fuzzy classification. It was also shown how a single model alone was not capable of generalizing to all sensor orientations, suggesting an ensemble modelling method was needed. This approach was also suggested by [16].

While Kalman filtering is the widely accepted method of fusing data in navigation literature, it cannot be applied to the

This work was supported by the EPSRC iCASE grant reference EP/S513623/1 and BAE Systems.

© 2024 IEEE. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.1109/ICIEA61579.2024.10665134.
Please refer to any applicable publisher terms of use.

problem concerned here and in the mentioned works. This is because the state variables (position, velocity) are not directly observable from a sensor reading. The data-driven methods can therefore be considered open loop estimators of the state.

In this paper, the SC-TS method is enhanced with the addition of another classification stage mapping velocity changes into displacement changes. The proposed two-stage SC-TS method (2SC-TS) thus completely replaces the double integration process with TS classification for improved positioning accuracy. Section II formulates the problem and outlines the original SC-TS method. Section III introduces the enhanced two-stage (2SC-TS) version. Results on three independent datasets are presented in Section III and conclusions in IV.

II. Methods – Improved Algorithm

A. Problem Formulation

The conventional method of estimating position in the navigation frame, n, $\boldsymbol{x}_{nb}^{n} \in \mathbb{R}^3$, involves double integration of acceleration expressed in the navigation frame, $\boldsymbol{a}_{nb}^{n} \in \mathbb{R}^3$ as shown in (1) with $\boldsymbol{a}_{nb}^{n}$ defined in (2):

$$\boldsymbol{x}_{nb}^{n} = \int_{T_0}^{T_n} \left(\int_{T_0}^{T_n} \boldsymbol{a}_{nb}^{n} \, \mathrm{d}t \right) \mathrm{d}t \tag{1}$$

$$\boldsymbol{a}_{nb}^{n} = \mathbf{R}_{b}^{n} \boldsymbol{f}_{nb}^{b} + \boldsymbol{g}^{n} \tag{2}$$

where $t \in \mathbb{R}^+$ is time, $\boldsymbol{f}_{nb}^{b} \in \mathbb{R}^3$ is a vector of accelerometer measurements (under the assumptions of local geographic navigation), $\boldsymbol{g}^{n} \in \mathbb{R}^3$ is the gravity vector in the navigation frame and $\mathbf{R}_{b}^{n} \in G$ is the rotation matrix giving the orientation of the sensor body frame relative to the navigation frame and $G = \mathrm{SO}(3)$ is the three-dimensional Lie group. $\mathbf{R}_{b}^{n}$ estimates are obtained from orientation filters like those described in [21]. In this work, the extended complementary filter introduced in [22] was used to estimate orientations.

As mentioned, the magnitude of errors in low-cost IMU measurements prevents the utility of this method and has inspired research into data-driven approaches such as the SC-TS method.

B. The SC-TS Method

The original SC-TS method in [20] replaced the first integral in (1) with a TS classifier that processed windows of IMU data, of length $n_{win} \in \mathbb{Z}^+$ expressed in the navigation frame to planar velocity vectors, $\boldsymbol{v}_{nb}^{n} \in \mathbb{R}^2$. This mapping is shown in (3):

$$[\boldsymbol{f}_{nb}^{n} \quad \boldsymbol{\omega}_{nb}^{n}]_{n_{win}\times 6} \overset{TS_1}{\longrightarrow} [\boldsymbol{v}_{nb}^{n}]_{n_{win}\times 2} \tag{3}$$

Like in RoNIN, an aligning rotation, $\mathbf{R}_{n_i}^{n_t} \in \mathrm{SO}(3)$, was first applied to the input features to ensure the navigation frame defined by the orientation filter, n_i, agreed with the reference frame of the ground truthing system, n_t. This was required because the heading angle defined by the filter was dependent on the initialization conditions of each sequence (no magnetometer was used in this method). This aligning rotation was assumed to be found by solving the constrained nonlinear optimization problem in (4) based on the first three seconds of motion:

$$\underset{\mathbf{R}_{n_i}^{n_t} \in \mathrm{SO}(3)}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{v}_{n_t\, b_t}^{n_t} - \int \left(\mathbf{R}_{n_i}^{n_t} \mathbf{R}_{b}^{n} \right) \boldsymbol{f}_{nb}^{b} + \boldsymbol{g}^{n} \mathrm{d}t \right)^2 \tag{4}$$

(4) represents the minimization of the mean squared velocity error between the ground truth velocities, $\boldsymbol{v}_{n_t\, b_t}^{n_t} \in \mathbb{R}^3$ and those estimated by taking the first integral of the linear accelerations, $\boldsymbol{v}_{n_i\, b_i}^{n_t} \in \mathbb{R}^3$, as shown in (5).

$$\underset{\mathbf{R}_{n_i}^{n_t} \in \mathrm{SO}(3)}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{v}_{n_t\, b_t}^{n_t} - \boldsymbol{v}_{n_i\, b_i}^{n_t} \right)^2 \tag{5}$$

The TS classifier was created by the SC algorithm which identified clusters in input and output data. Based on the mountain clustering method, SC was chosen over other clustering methods such as fuzzy c-means clustering (FCM) and grid partitioning (GP) as their average trajectory errors over the training and test data were found to be larger as shown in Table I. The error metrics used were the absolute and relative trajectory errors, ATE and RTE respectively, as defined in RoNIN. GP was found to be the most expensive computationally as only multiple input single output (MISO) systems could be generated, thus requiring two models to be trained to estimate planar velocity vectors. The better performance of SC was likely due to the ability of the method to reject outliers in data as shown in [20].

TABLE I. Average Trajectory Error for Different Clusteirng Algorithms.

	SC	FCM	GP
ATE (m):	2.5	2.8	19
RTE (m):	3.1	3.4	21

Optimal parameters for the SC method were found by a grid search. Output data were the low pass filtered first derivative of the positions from the ground truth system. During inference, a first order Euler discretization created position estimates from the estimated velocity vectors. Fig. 1 shows how the data was processed in the SC-TS method.

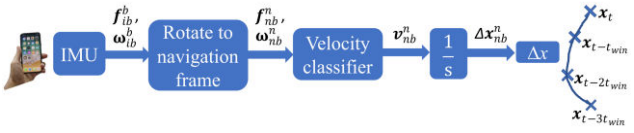


Fig. 1. Diagram of the original SC-TS method.

C. The Two-Stage SC-TS Method (2SC-TS)

The main improvement in 2SC-TS was the replacement of the Euler discretization step, represented by the 1/s block in Fig. 1, with the positional change classifier shown in Fig. 2.

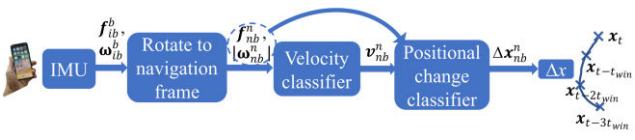


Fig. 2. Diagram of the 2SC-TS method.

The positional change classifier was another TS model created by the SC algorithm. Input data were defined as windows of planar velocity estimates from the first classifier, the specific force expressed in the navigation frame and the L-1 norm of the angular velocities expressed in the navigation frame as shown in (7):

$$[\boldsymbol{f}_{nb}^{n} \quad |\boldsymbol{\omega}_{nb}^{n}|]_{n_{win}\times 4} \overset{TS_1}{\longrightarrow} [\boldsymbol{v}_{nb}^{n}]_{n_{win}\times 2} \tag{6}$$

$$[\boldsymbol{v}_{nb}^{n} \quad \boldsymbol{f}_{nb}^{n} \quad |\boldsymbol{\omega}_{nb}^{n}|]_{n_{win}\times 6} \overset{TS_2}{\longrightarrow} [\Delta \boldsymbol{x}_{nb}^{n}]_{n_{win}\times 2} \tag{7}$$

Comparing (3) and (6), a secondary improvement to the method was the replacement of the angular velocities in the navigation frame, $\boldsymbol{\omega}_{nb}^{n} \in \mathbb{R}^3$, with their L-1 norm $|\boldsymbol{\omega}_{nb}^{n}|$. This reduced the dimensionality of the inputs to the first classifier by a third and the time to generate the model by 35% with no observed detriment to the accuracy of the estimated velocities.

The SC algorithm creating the positional change TS classifier used the same optimal parameters found by the grid search: r_a = 0.3, f_{squash} = 1.4, r_{accept} = 0.45 and r_{reject} = 0.35.

III. Results

To evaluate the performance of the proposed 2SC-TS method, three independent data sets were used: the two subsets from the large-scale folder in the Oxford Inertial Odometry Dataset (OxIOD), a subset from the Robust IMU Double Integration (RIDI) dataset and our own experimental dataset. While the OxIOD and RIDI datasets captured the motion of people carrying inertial sensors inside smartphones, our experimental dataset captured the motion of a carriage travelling around a circuit measuring 1.5x1.5m. Fig. 3 shows the laboratory setup used to capture the IMU and ground truth data. Table II shows details of the duration of the data used for training, their average speeds, $\mu_{\|v\|}$, and respective variances in speed, $\sigma^2_{\|v\|}$. Table II also shows details of the test sequences used to evaluate the performance of the algorithms.

To demonstrate the advantages of the proposed 2SC-TS method, it was compared against IONet, RoNIN and the original SC-TS. While 2SC-TS and SC-TS models were generated on a desktop PC, the RoNIN and IONet were trained using an Nvidia Tesla A100 graphics processing unit. For both, training lasted 100 epochs on 80% of each dataset with 10% allocated for validation and the remainder for testing.

While the code for RoNIN was publicly available, IONet was not so a local implementation was made in MATLAB to best of the authors' knowledge with the details available from [23]. Specifically, a two-layer bidirectional LSTM was used to generate change in heading and displacement estimates from inertial sensor data expressed in the sensor body frame, $\boldsymbol{f}_{ib}^{b}$ and $\boldsymbol{\omega}_{ib}^{b}$. The sequence-to-one regression model therefore estimated the kinematic variables based on sliding windows of IMU data of length 200 samples with a stride of ten. Min-max scaling was applied to the input and output data as this was found to result in lower validation loss. For training, ADAM optimizer was used with initial learning rate of 1e-4.

To ensure a fair comparison with other methods, the same trajectory error metrics used in RoNIN method were also used here: ATE and RTE.

Table III shows the average ATEs and RTEs achieved by each method on the test sequences of the three datasets. Fig. 4(a)-(d) show the trajectories estimated by each algorithm on each of the test sequences.

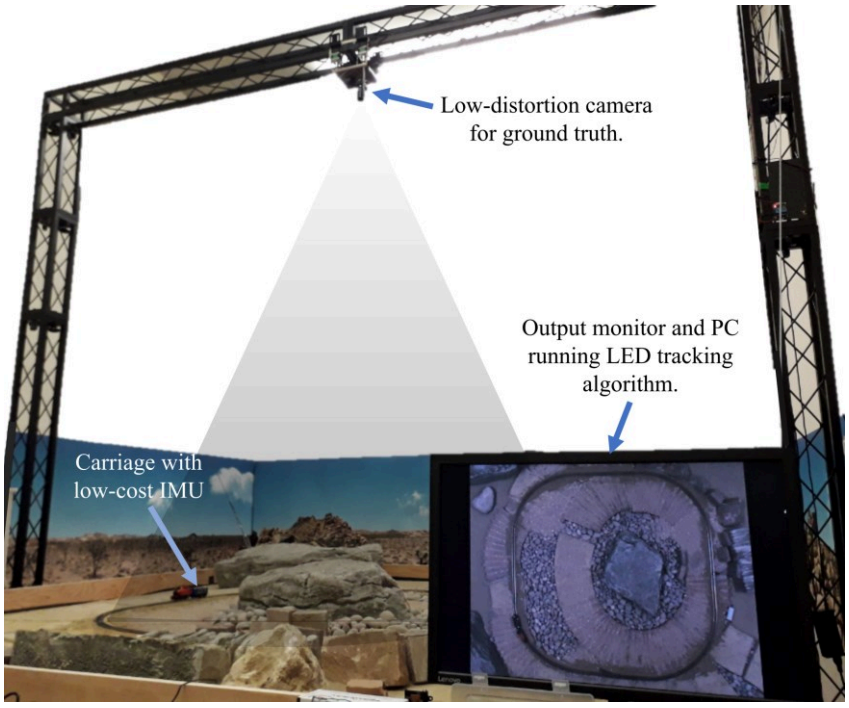


Fig. 3. Laboratory setup used for the acquisition of our own dataset.

TABLE II. PROPERTIES OF DATA USED FOR TRAINING AND TESTING THE MODELS.

Dataset	No. training sequences	Total duration for training (min)	$\mu_{\|v\|}$ (ms^{-1})	$\sigma^2_{\|v\|}$ (ms^{-1})	No. test sequences	Total duration for testing (min)
OxIOD	14	35	0.93	1.0e-3	2.0	4.7
RIDI	12	17	1.1	0.10	1.0	1.9
Own	15	15	0.20	4.0e-3	1.0	1.2

TABLE III. ATES AND RTES OF THE ALGORITHMS ON THE TEST TRAJECTORIES.

	SC-TS		2SC-TS		RoNIN		IONet	
Dataset	***ATE (m)***	***RTE (m)***	***ATE (m)***	***RTE (m)***	***ATE (m)***	***RTE (m)***	***ATE (m)***	***RTE (m)***
OxIOD	2.7	3.7	2.5	3.6	4.4	6.7	44	51
RIDI	5.1	4.9	3.3	3.6	19	12	40	47
Own	0.17	0.30	0.15	0.18	0.59	0.80	0.77	1.4
Average:	**2.7**	**3.0**	**2.0**	**2.5**	**8.1**	**6.5**	**28**	**33**

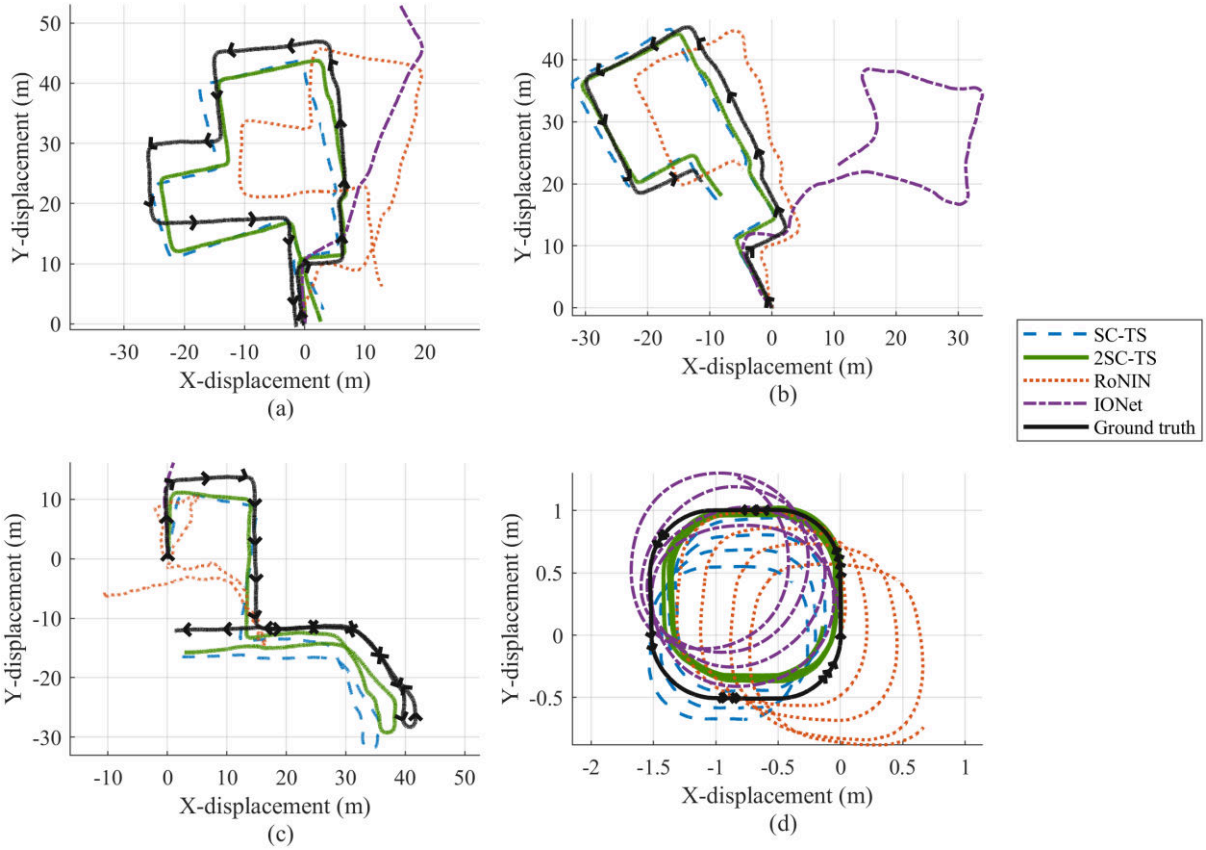


Fig. 4. Estimated trajectories for, (a)-(b) OxIOD largescale test sequences, (c) RIDI test sequence and (d) test sequence from own dataset.

TABLE IV. AVERAGE MODEL SIZES AND INFERENCE TIMES.

	SC-TS	2SC-TS	RoNIN	IONet
Average model size (kb):	4.0	7.0	53000	2000
Average inference time (s)[a]:	0.78	0.85	0.59	6.9

[a.] Inference time on a sequence of 60s duration.

The results from Table III show how 2SC-TS reduced the mean ATE and RTE by about 25% and 15% respectively compared with the best performing baseline algorithm, the original SC-TS method. Compared with RoNIN, 2SC-TS halved the mean trajectory errors on the test sequences.

The trajectories on the first OxIOD test sequence in Fig. 4(a) show how, although 2SC-TS initially underestimated displacement in the y-direction, it retained a more accurate heading estimate compared with RoNIN at the end of the sequence. The proposed method outperformed the original version, SC-TS, as the additional classification stage helped correct the predicted velocities of the first. IONet consistently underestimated the changes in heading angle which caused it to deviate after the first three 90 degree turns. This is likely due to the inputs to IONet being expressed in the sensor body frame rather than the navigation frame as in the other methods, thus making it highly sensitive to small deviations from the conditions of the training data.

Fig. 4(b) shows how the 2SC-TS method remained closer to the ground truth towards the end of the trajectory but less at the start compared with RoNIN. As in Fig. 4(a), IONet rapidly deviated from the ground truth although it does capture some of the 90 degree turns to form a square trajectory.

Fig. 4(c) shows how 2SC-TS captured the general shape of the ground truth trajectory compared with the other methods. The heading estimates from 2SC-TS are more accurate throughout the trajectory compared with SC-TS. However, all performed worst on this data and the 2SC-TS trajectory appears to have underestimated displacements equally in x and y directions. Referring to the properties of the datasets in Table II, this poor performance may be due to the greater variation in the speeds featured in the RIDI data set (approximately ten times larger than the others).

The trajectories on our own test data in Fig. 4(d) show how 2CS-TS conformed the closest to the ground truth throughout the whole sequence and maintained a more accurate heading than RoNIN and IONet. While SC-TS retained a heading equally as accurate as 2SC-TS, after four circuits of the trajectory it drifts into the negative x and y directions.

Table IV shows how the models produced by the SC-TS methods were orders of magnitude smaller in size than the deep learning models. While RoNIN had the fastest inference, the SC-TS methods followed close with IONet being the slowest. While it was expected that 2SC-TS would take twice as long as SC-TS, the reduction in the velocity classifier input dimensions for 2SC-TS meant its inference time was only marginally greater than SC-TS.

IV. Conclusion

Building on previous work in data-driven positioning using low-cost IMUs, this paper presented 2SC-TS, a two-stage classification method. 2SC-TS completely replaces the double integration of IMU data with TS models generated by the SC algorithm. Compared with the original SC-TS method, 2SC-TS improves the positioning accuracy by about 25% in ATE. It also reduces the training and inference times of the velocity classifier by reducing the dimensionality of its inputs by a third. Compared with the RoNIN algorithm, 2SC-TS halves the errors on the test trajectories, generates a model that is orders of magnitude smaller in memory and almost as fast at processing data. The method proposed may help localize people or robots carrying low-cost IMUs more accurately.

Although 2SC-TS presented an improvement compared with the other methods, the ability of the method to generalize to data containing greater variation in speed, as in the RIDI dataset, requires further investigation. Future work will explore the use of different inputs to the classifiers to achieve lower trajectory errors. The generalization of the method may be improved with the creation of an ensemble modelling method to adapt to a wider variety of conditions. Specifically, it is suggested that the ensemble model includes accurate classification of the mode of carriage of the sensor on the body and accurate determination of the direction of motion after a stationary period.

During the development of the method, it was found that the quality of trajectories was also dependent on the accuracy of the aligning rotation described in Section II.B. Future investigations intend to analyse the effect this rotation has on the inputs to the model, thus enhancing conceptual understanding of the method and providing clues to how it may be improved further.

Data Availability Statement

The data supporting the findings of this work are openly available to download from Google Drive via http://deepio.cs.ox.ac.uk/ for the OxIOD data set at https://yanhangpublic.github.io/ridi/index.html for the RIDI data set. Our experimental data is openly available from CORD at 10.17862/cranfield.rd.25943572.

Acknowledgment

The authors are very grateful for the technical assistance of Mr. Robert A. Cooper in the construction of the experimental rig and the many insightful discussions relating to the work.

For the purposes of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Accepted Author Manuscript version arising from this submission.

References

[1] United Nations Department of Economic and Social Affairs Population Division, “World Urbanization Prospects 2018 Highlights,” New York, 2019.

[2] R. Sun et al., “Improving GPS code phase positioning accuracy in urban environments using machine learning,” IEEE Internet Things J., vol. 8, no. 8, pp. 7065–7078, Apr. 2021.

[3] National Fire Protection Association (NFPA), “After effect.” Accessed: Apr. 05, 2024. [Online]. Available: https://www.nfpa.org/news-blogs-and-articles/nfpa-journal/2019/11/01/worcester

[4] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, “An experimental study of GPS spoofing and takeover attacks on UAVs,” in 31th USENIX Security Symposium (USENIX Security 22), Aug. 2022. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/sathaye

[5] L. Yangi et al., “Wall-climbing robot for visual and GPR inspection,” in 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp. 1004–1009, May 2018.

[6] D. Coppens, A. Shahid, S. Lemey, B. Van Herbruggen, C. Marshall, and E. De Poorter, “An overview of UWB standards and organizations (IEEE 802.15.4, FiRa, Apple): Interoperability

Aspects and Future Research Directions," IEEE Access, vol. 10, pp. 70219–70241, Jun. 2022.

[7] A. R. Jimenez Ruiz and F. Seco Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB location systems: Indoor performance analysis," IEEE Trans. Instrum. Meas., vol. 66, no. 8, pp. 2106–2117, Aug. 2017.

[8] K. Derr and M. Manic, "Wireless based object tracking based on neural networks," in 2008 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp. 308–313, Jun. 2008.

[9] H. He, T. Liu, and E. Wang, "Intelligent book positioning system for library based on RFID," in 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp. 723–726, Jun. 2017.

[10] E. Foxlin, T. Calloway, and H. Zhang, "Design and error analysis of a vehicular AR system with auto-harmonization," IEEE Trans. Vis. Comput. Graph., vol. 21, no. 12, pp. 1323–1335, Dec. 2015.

[11] K. Ramesh, S. N. Nagananda, H. Ramasangu, and R. Deshpande, "Real-time localization and navigation in an indoor environment using monocular camera for visually impaired," in 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), IEEE, pp. 122–128, Apr. 2018.

[12] F. Amon, N. Bryner, and A. Hamins, "Thermal imaging research needs for first responders: Workshop proceedings," Gaithersburg, MD, 2005.

[13] A. Noureldin, T. B. Karamat, and J. Georgy, Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp.1-19.

[14] X. Wu, L. Duan, and W. Chen, "A Kalman filter approach based on random drift data of fiber optic gyro," in 2011 6th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp. 1933–1937, Jun. 2011.

[15] J. Wahlstrom and I. Skog, "Fifteen years of progress at zero velocity: A review," IEEE Sens. J., vol. 21, no. 2, pp. 1139–1151, Jan. 2021.

[16] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU Double Integration," in Computer Vision – ECCV 2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 641–656.

[17] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 6468–6476, Jan. 2018.

[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] S. Herath, H. Yan, and Y. Furukawa, "RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, new methods," Proc. IEEE Int. Conf. Robot Autom., pp. 3146–3152, May 2020.

[20] D. Maton, J. Economou, D. Galvão Wall, D. Ward, and S. Trythall, "Subtractive clustering Takagi-Sugeno position tracking for humans by low-cost inertial sensors and velocity classification," Measurement and Control, vol. 56, no.9-10, pp. 1534-1544, Apr. 2023.

[21] D. Maton et al., "Indirect tuning of a complementary orientation filter using velocity data and a genetic algorithm," Sys. Sci. Control Eng., vol. 12, no. 1, Apr. 2024.

[22] S. O. H. Madgwick, S. Wilson, R. Turk, J. Burridge, C. Kapatos, and R. Vaidyanathan, "An extended complementary filter for full-body MARG orientation estimation," IEEE ASME Trans. Mechatron., vol. 25, no. 4, pp. 2054–2064, Aug. 2020.

[23] C. Chen, "Learning Methods for Robust Localization," Doctoral thesis, Dept. Comp. Sci., Univ. Oxford, Oxford, United Kingdom, 2020.



https://dspace.org/

A two-stage classification method for improved positioning using low-cost inertial sensors

Maton, Dariusz; Economou, John T.; Galvão Wall, David; Khan, Irfan; Cooper, Robert ...

2024-08-08
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/ICIEA61579.2024.10665134
Maton D, Economou J, Galvao Wall D, et al., (2024) A two-stage classification method for improved positioning using low-cost inertial sensors. In: 19th IEEE Conference on Industrial Electronics and Applications (ICIEA 2024), 5-8 August 2024, Kristiansand, Norway

Downloaded from a DSpace repository. https://dspace.org/