Sensors & Transducers, Vol. 261, Issue 2, June 2023, pp. 1-9

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2023
http://www.sensorsportal.com

Vehicle Localization by Optimally Weighted Use of Cross-Correlated MEMS Sensor Data

Takayoshi YOKOTA
Department of Information Design, Tokyo Information Design Professional University,
2-7-1 Komatsugawa, Edogawa-ku, Tokyo, 132–0034, Japan
Tel.: +81-3-6808-3201
E-mail: yokota@tid.ac.jp

Received: 10 April 2023 Accepted: 15 June 2023 Published: 26 June 2023

Abstract: The Global Navigation Satellite System (GNSS) is widely utilized to accurately determine the position of a vehicle, but at present, the accuracy may be degraded depending on the radio wave reception conditions from the satellite. There are also growing concerns about cyber-attacks on GNSS. We have been developing a method for estimating the position of a traveling vehicle using MEMS sensor data obtained from air pressure, acceleration, gyro, and geomagnetic sensors. In the current work, we set up an experimental environment to collect data at 50 Hz and implemented a running test on an actual road. Optimizing the weighting of each item of sensor information enabled us to obtain characteristics of the road that could not be captured in the past. In addition, the application of an extreme data elimination filter confirmed that the position estimation accuracy was improved on the order of sum-meter.

Keywords: Localization, Atmospheric pressure, Acceleration, Geomagnetic, MEMS sensor, Optimum weight.

1. Introduction

The Global Navigation Satellite System (GNSS) is widely utilized for position estimation of vehicles and pedestrians in everyday life, but its accuracy may be degraded depending on the reception status of the radio waves from the satellite. This paper is focused on position estimation of vehicles. Urban areas with high-rise buildings have long been a problem in this regard, and in recent years, radio interference caused by solar flares and intentional jamming, although rare, have occurred in some areas. This has created the need for a new technology that can compensate for the inherent disadvantages of vehicle position estimation technology, which is highly dependent on GNSS and requires huge maintenance costs. In this study, we propose a localization algorithm for a traveling vehicle based on the characteristics of the road obtained from MEMS sensors including acceleration, angular velocity, and magnetization for x, y, and z axes relying only minimally on GNSS. This paper is an extension of an earlier conference paper [1] with new results concerning the effect of the optimum weighting of cross-correlation functions compared with the results obtained by equal weight.

2. Proposed Algorithm

2.1 Literature Review

Terrain-based localization is an interesting research field that has emerged in recent years [1– 6], [8–12]. Most studies in this vein have focused on using the pitch rate of a running vehicle as features of the road surface. Similar to the author's approach [1–6], the

position of a vehicle is estimated by acquiring the properties of the road surface in real time by matching vibration data [8], pitch data [9, 10], or IMU data [11] in time domain. These are position estimation algorithms that attempt to identify a current position by matching reference sensor data prepared in advance. The accuracy of the position estimation obtained by these studies is on the order of several meters, and it remains unclear if sub-meter accuracy is possible. Landmark-based localization is a new technique that deals with suitable objects in a road scene that can be used as features for identifying the location of a vehicle or user. Image processing-based localization including 3D LiDAR depth maps can be effective [12], but the cost of preparing data is not small. Whereas most of the prior studies have used the pitch rate of running vehicles as a feature of the road surface, in the current study, nine types of sensor data comprising acceleration, angular velocity, and geomagnetism for three directions (x, y, and z) are investigated to determine which are useful. Atmospheric pressure data is not included because it does not contain much information in flat road cases.

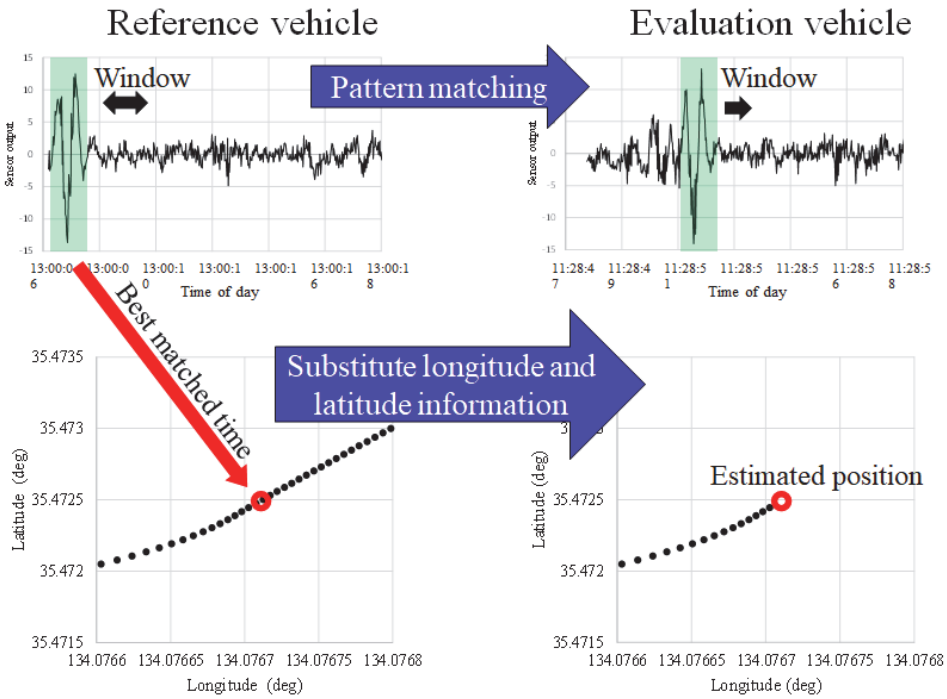


Fig. 1. Outline of proposed algorithm. The reference vehicle stores sensor data together with RTK-GNSS position data for later reference.

2.2. Vehicle Localization Algorithm by MEMS Sensor

We have been working on a vehicle localization algorithm based on MEMS sensors for the past several years [1–6]. In this algorithm, accurate location information obtained by high-precision GNSS including RTK-GNSS (u-blox ZED-F9P) [14] as well as acceleration, angular velocity, and geomagnetism information (Invensense 9250) [15] are collected in advance on an actual road for use as reference data. A test vehicle for evaluating the performance of the localization algorithm is then run to acquire sensor data, which are subsequently compared with the above reference data to obtain an accurate location. The outline of the proposed algorithm is shown in Fig. 1. Pattern matching is performed on the basis of the normalized cross-correlation function defined as

$$R(i,j) = \frac{\sum_{k=0}^{N-1}\left(f_{i+j+k}-\bar{f}_{i+j}\right)\left(g_{i+k}-\bar{g}_i\right)}{\sqrt{\sum_{k=0}^{N-1}\left(f_{i+j+k}-\bar{f}_{i+j}\right)^2}\sqrt{\sum_{k=0}^{N-1}\left(g_{i+k}-\bar{g}_i\right)^2}}, \quad (1)$$

$$\bar{f}_{i+j} = \frac{1}{N}\sum_{k=0}^{N-1} f_{i+j+k}, \quad (2)$$

$$\bar{g}_i = \frac{1}{N}\sum_{k=0}^{N-1} g_{i+k}, \quad (3)$$

where i in Eq. (1) is the index of evaluation for the vehicle's travel time and j is the index of the time lag between two vehicles (reference vehicle and evaluation vehicle). If we can determine this time lag accurately, the location of the evaluation vehicle can be substituted by the location (latitude and longitude)

of the reference vehicle at the matched time. Pattern matching can then be achieved by finding the maximum value of similarity of the sensor data with a mutual correlation function calculated over a specific time region (e.g., one second).

3. Experiments

3.1 Experimental Setup

We evaluated the proposed algorithm on a flat road section (Fig. 2) with very little change in the road attributes, where the localization error was as large as 100 m or more in tests with an earlier version of our method [2], [3]. In the prior tests, even when an accelerometer and gyro sensor were utilized in addition to the barometric sensor, there was no improvement of the position accuracy at the processing interval of 0.1 s. This was due to the influence of aliasing noise caused by the sampling interval being too large. Therefore, the objective in the current study is to reduce the effect of aliasing noise by performing normalized cross-correlation at 0.02 s (50 Hz). The experimental setup is summarized in Table 1. The MEMS sensors (Invensense Mpu9250) and Bosch BME280s were installed in the rear seat of the vehicle, as shown in Fig. 3, and the resultant data were processed with the resolution of 50 Hz for Mpu9250 data. The atmospheric sensor data was utilized simply to identify the flatness of the test course.



Fig. 2. Flat and straight 1.36-km test course alongside a Grade 2 river.

Fig. 4 shows the altitude along the test course, where we can see that the altitude change is very small, especially in the range of 30 to 50 s of the travel time. Since this road is alongside a river, the overall altitude change is 11.0 m/km (15 m / 1.36 km) and is quite smooth. The altitude was measured by averaging 16 atmospheric pressure sensors (Bosch BME280) [13] to reduce random noise and then estimated from the atmospheric pressure.

Table 1 Experimental setup.

Date	Dec.5(Sun),2021 ,Mar.6(Sun),2022
Vehicle	Daihatsu Mira geno (Small car)
MEMS sensors	1. Invensense Mpu-9250, 50Hz data acquisition. Acceleration, angular velocity,geomagnetism 2. Bosche BME280 atmospheric pressure sensor, 20Hz data acquisition
RTK-GNSS	u-blox ZED-F9P,5Hz data acquisition
Test course	1.36km, Shikano-cho, Tottori-shi,Japan

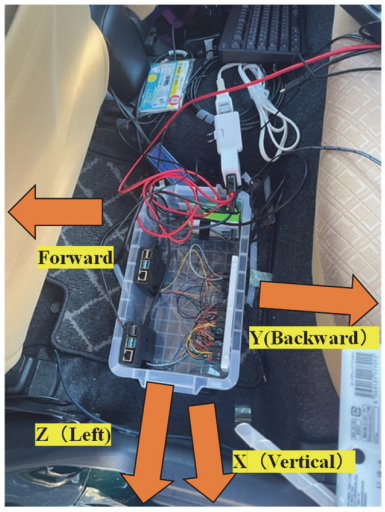


Fig. 3. Sensor axes of Mpu-9250 (indicated by arrows).

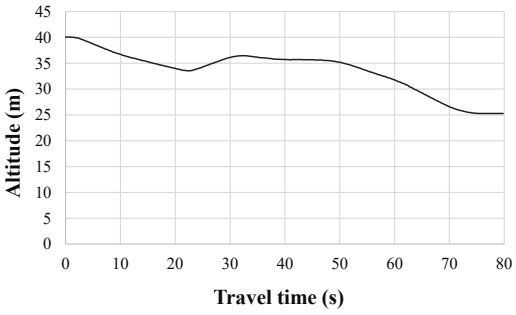


Fig. 4. Altitude profile along test course.

3.2. Pattern Matching by Normalized Cross-correlation Function

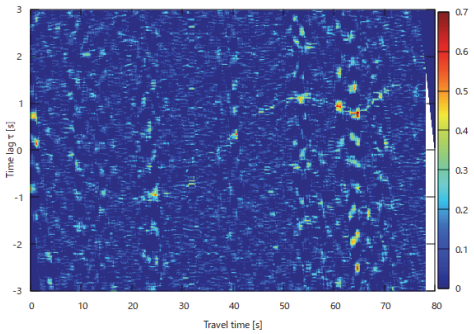
Nine normalized cross-correlation functions were obtained for each sensor data (acceleration, angular

velocity, and magnetization for x, y, and z axes). Each item of sensor data of the evaluation vehicle on Mar. 6, 2022 was correlated with the reference vehicle data of Dec. 5, 2021.

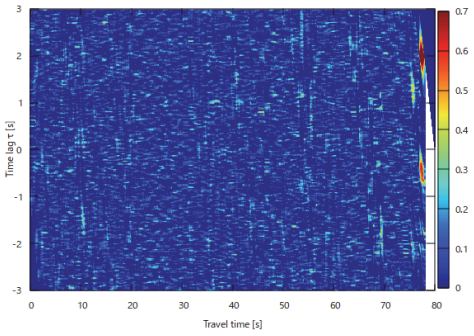
The cross-correlation functions are displayed using the jet colormap in MATLAB.

3.2.1 Normalized Cross-correlation of Acceleration Sensor Data

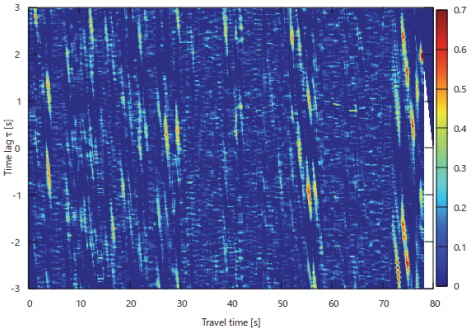
Fig. 5 shows the results of the normalized cross-correlation of the acceleration data. As we can see, there is a slight time lag profile for ax, that is, acceleration in the vertical direction.



(a) ax



(b) ay



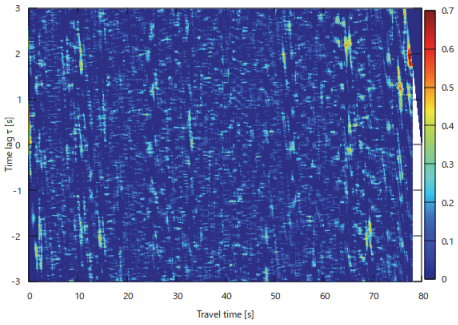
(c) az

Fig. 5. Normalized cross-correlation of acceleration sensor data. Acceleration data toward x, y, and z axes are denoted as ax, ay, and az.

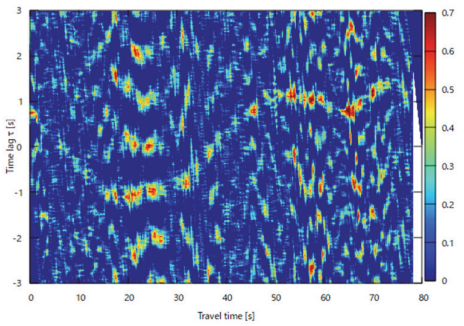
3.2.2. Normalized Cross-correlation of Acceleration Sensor Data for Gyroscope

Fig. 6 shows the results of the normalized cross-correlation of the gyroscope data, where omz clearly shows the time lag profile. Omy shows the time lag profile as well, though its resolution is lower. Gyroscope data around the z axis represents the pitch rate and around the y axis represents the roll rate.

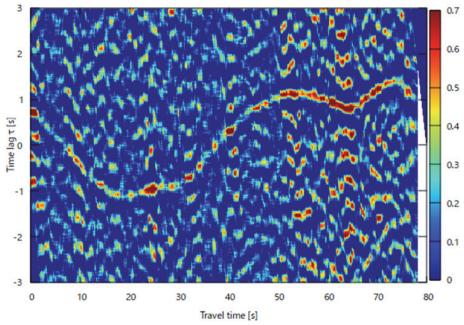
Yaw rate, that is, the rotation rate around the x axis, does not effectively capture the road features and the time lag profile cannot be seen. This is because the test road is mostly straight and so there is very little change in the yaw rate.



(a) omx



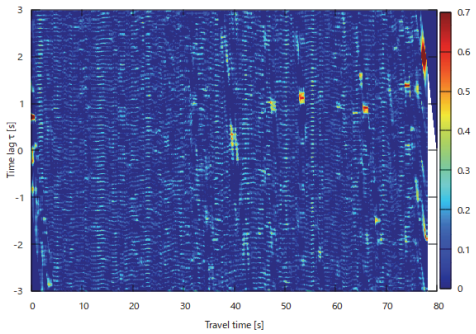
(b) omy



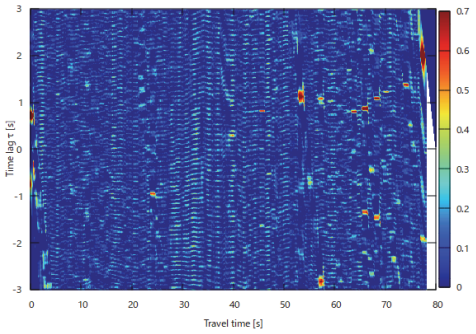
(c) omz

Fig. 6. Normalized cross-correlation of gyroscope data. Gyroscope data around x, y, and z axes are denoted as omx, omy, and omz.

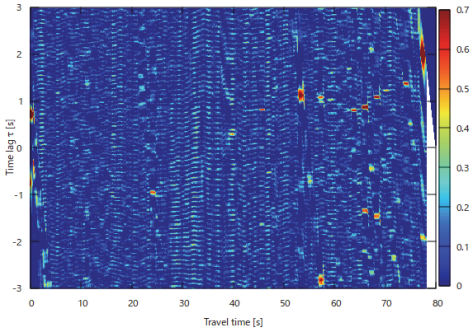
Fig. 7 shows the results of the normalized cross-correlation of the geomagnetism data. Some portions of the time lag profile are visible, especially for the latter part from 40 to 80 s of the evaluation vehicle travel time. The resolution is rather high.



(a) mx



(b) my



(c) mz

Fig. 7. Normalized cross-correlation of geomagnetism data. Geomagnetism data along x, y, and z axes are denoted as mx, my, and mz.

3.3. Optimum Weight of Normalized Cross-correlation Functions

In the previous sections, we demonstrated that nine normalized cross-correlation functions can be obtained, some of which clearly show the time lag profile and others of which do not. In this section, we describe how to optimally weight these functions and average them to form an optimum normalized cross-correlation function. The optimum weights are calculated by a simulated annealing method so that the measure function F (defined in Eq. (7)) is maximized. F measures the signal-to-noise ratio (SNR) of the optimum normalized cross-correlation function of Eq. (4).

$$R_{opt}(i,j) = \sum_{n=0}^{M-1} w_n R_n(i,j) \tag{4}$$

$$w_n \geq 0, n = 0,1,\ldots,8 \tag{5}$$

$$\sum_{n=0}^{8} w_n = 1 \tag{6}$$

$$F = \frac{1}{M}\sum_{i=0}^{M-1} \frac{max_j R_{opt}(i,j)}{\left|min_j R_{opt}(i,j)\right|} \tag{7}$$

Fig. 8 shows the optimum weights we derived. As we can see, the pitch rate, vertical magnetism, roll rate, and backward magnetism all play significant roles (74.7%) in vehicle localization in the proposed algorithm. Fig. 9 shows the optimally weighted cross-correlation functions, where the time lag profile is clearly recognizable.

Fig. 10 shows the equally weighted cross-correlation functions. As we can see, the time lag profile is much more clearly observed in Fig. 9 than in Fig. 10.

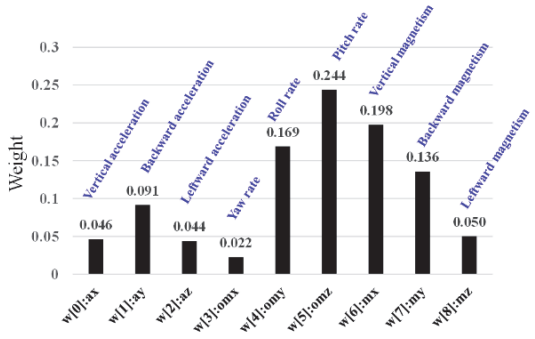


Fig. 8. Optimum weight of cross-correlation functions.

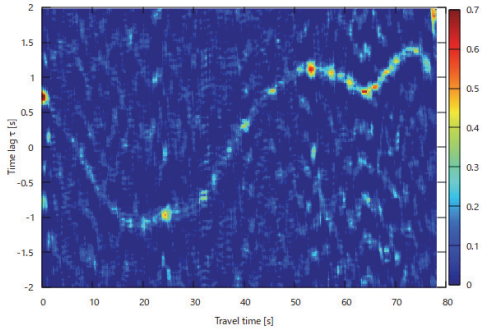


Fig. 9. Optimally weighted cross-correlation functions.

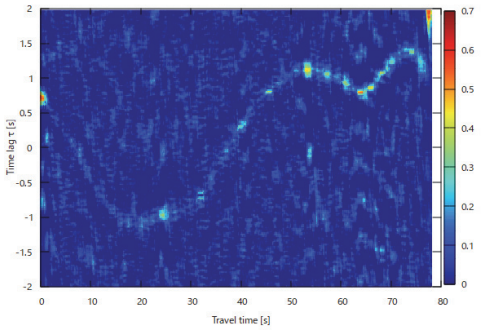


Fig. 10. Equally weighted cross-correlation functions.

3.4. Convergence of Optimum Weight

We applied a simple simulated annealing algorithm to obtain a set of optimum weights. Fig. 11 shows the value profile of the measure defined by Eq. (7) during the simulated annealing. As we can see, it successfully converges to a maximum value of 1.383 starting at its initial value of 1.305, where all weights are set equal to 1/9 = 0.1111.

Fig. 12 shows the values of each weight, all of which have converged almost by iteration 200. The CPU consumption time using an INTEL core-i5 CPU is several minutes. Since these weights only need to be calculated once, high-speed calculation (e.g., within one second) is not required.

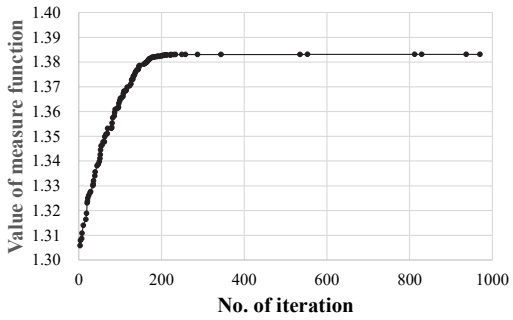


Fig. 11. Convergence of measure function.

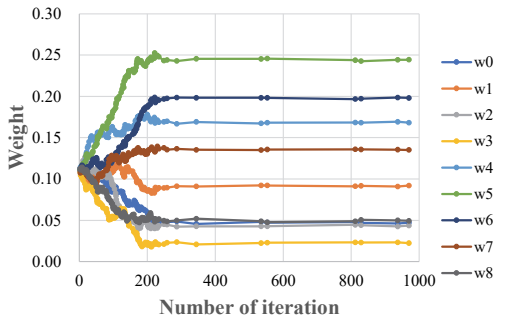


Fig. 12. Convergence of each weight.

3.5. Estimation of Vehicle Time Lag by Noise Reduction

To estimate the vehicle time lag by noise reduction, we first obtained the time lag for each travel time point by searching for a maximum cross-correlation value for each travel time point. Fig. 13 shows the time lag estimates when the weights for each cross-correlation function are equal, and Fig. 14 shows the results when the optimum weights are used. It is clear that a less noisy estimate of the time lag was made possible thanks to using the optimum weight.

The next step is to estimate the time lag profile from the noisy time lag estimates in Fig. 13 (equal weight) or Fig. 14 (optimum weight). Even though noise is eliminated to some extent, especially in the case of Fig. 14, it is still noisy.

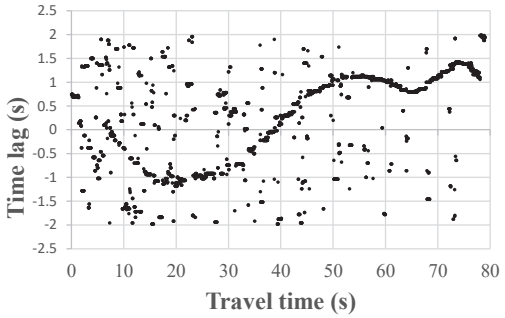


Fig. 13 Time lag estimate with equal weight before applying noise elimination filter.

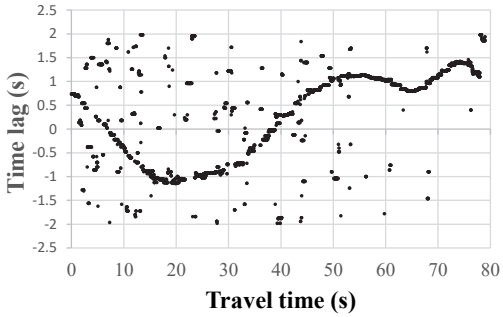


Fig. 14. Time lag estimate with optimum weight before applying noise elimination filter.

We therefore developed a simple but practical noise elimination filter for these noisy time lag data, as described by the pseudocode of Algorithm 1. Parameter θ is a positive threshold value to decide whether the time lag estimate $\hat{\tau}_{i+1}$ should be replaced by the previous estimate $\hat{\tau}_i$ or not. If the next time lag data τ_{i+1} differs from the current estimate $\hat{\tau}_i$ beyond the range limit θ, we keep the current value; otherwise, the time lag τ_{i+1} is accepted.

Algorithm 1. Noise elimination filter.

```
Begin
    for (i=0; 0<IMAX; i=i+1)
        if |τ_{i+1} − τ̂_i| < θ then τ̂_{i+1} = τ_{i+1}
        else τ̂_{i+1} = τ̂_i
        endif
End
```

Here τ_i is the time lag obtained by searching for the maximum value at each travel time, and it may differ significantly from the true value due to sensor noise or other factors. $\hat{\tau}_i$ is the estimated value of the time lag at time index i, and it is hopefully eased somewhat from the effect of noise. i is the time index of the travel time of the evaluation vehicle. IMAX is the quantity of the time series data. In this paper, θ is set to 0.4 s, which is not much strict, and almost the same results are obtained in the range of $0.3\ s < \theta < 0.4\ s$.

Fig. 15 show the time lag profile estimated from the optimally weighted cross-correlation functions after applying the noise elimination filter. The ground truth value of the time lag obtained by RTK-GNSS is also shown.

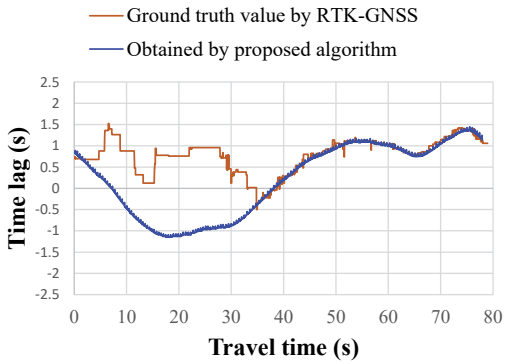


Fig. 15. Time lag estimate with equal weight after applying noise elimination filter.

Fig. 16 show the time lag profile estimated from the equally weighted cross-correlation functions after applying the noise elimination filter. The ground truth value of the time lag obtained by RTK-GNSS is also shown.

It is clear that the time lag profile obtained from the optimally weighted cross-correlation functions (Fig. 16) matches quite well to the ground truth data. In contrast, the time lag estimate from the equally weighted cross-correlation functions (Fig. 15) failed to accurately estimate the time lag in the range of 0 to 35 sec because of insufficient noise elimination. The ground truth of the time lag between the two vehicles was calculated by comparing the RTK-GNSS trace of each vehicle. RTK-GNSS was operating at 5 Hz and the traces were over-sampled at 50 Hz by introducing a linear interpolation of 5-Hz position data. It is reasonable that a vehicles velocity cannot be changed so rapidly within 0.2 seconds, so we consider the obtained 50-Hz resolution ground truth data to be sufficiently reliable. The RMS error of the time lag estimation in the optimum weight case was 0.091 s, which is a far more precise result than what we obtained with our earlier method [2, 3]. At the travel time of 50 s, the time lag estimate reached the maximum error of about 0.3 s.

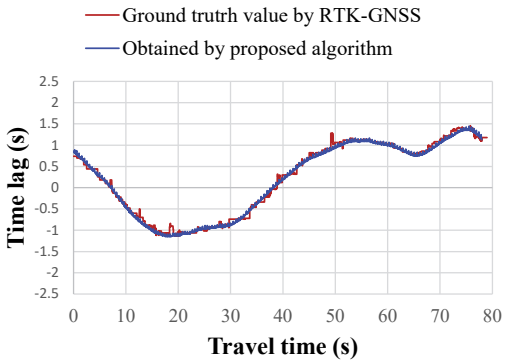


Fig. 16. Time lag estimate with optimum weight after applying noise elimination filter.

3.6. Estimation of Vehicle Position and its Evaluation

We utilized the proposed algorithm to precisely estimate the position error following the data flow in Fig. 17.

The algorithm input is the date and time the evaluation vehicle is running, and its corresponding MEMS sensor data are pattern matched to the reference MEMS sensor data. This pattern matching outputs the time lag between the evaluation vehicle and the reference vehicle. In our study, the reference vehicle was running in December 2021 and the evaluation vehicle was running in March 2022, which is a time lag of roughly three months. For simplicity, we reduced this lag to several seconds (note that this does not limit the generality of the proposed algorithm). The ground truth position of the evaluation vehicle is obtained by the RTK-GNSS and the estimated position is obtained by the position of the reference vehicle obtained by RTK-GNSS at the date and time of the reference vehicle that is obtained by the time lag estimate. The position error can be calculated from longitude and latitude values of the RTK-GNSS data of both vehicles.

Fig. 18 shows the position error along the travel time of the evaluation vehicle when the cross-correlation functions are equally averaged. As we can see, the maximum error is 9.6 m at the travel time of about 22 s. This is due to the failure of the time lag estimate, as shown in Fig. 15.

Fig. 19 shows the position error along the travel time of the evaluation vehicle when the cross-correlation functions are optimally weighted. In this case, the RMS position error is 0.37 m and the spike-like maximum error is 1.75 m. This result is superior

to any result of our previous study. The maximum error occurs at the travel time of 47 s, which is also where the time lag estimate error is apparent in Fig. 16.

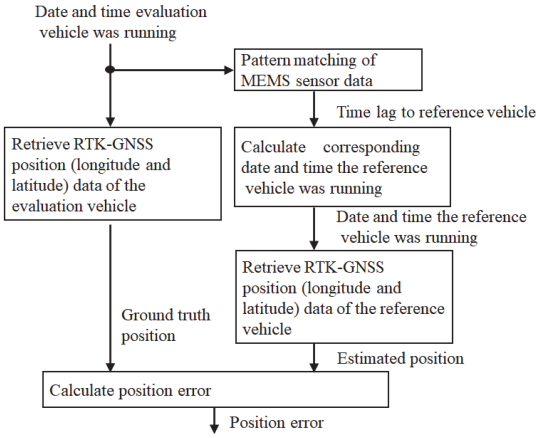


Fig. 17. Flow of position error evaluation.

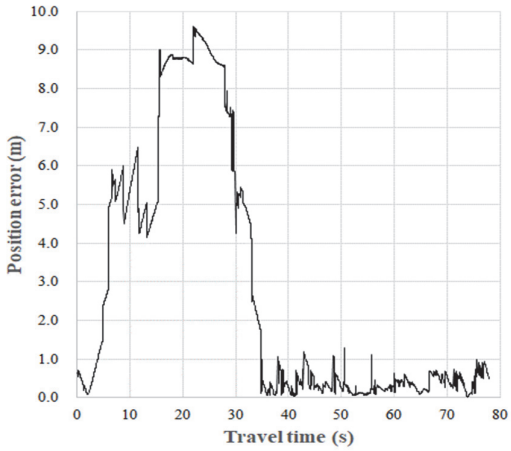


Fig. 18. Position error along evaluation travel time with equally averaged cross-correlation functions. Failure occurs in the range of 0–30 s. The maximum error is 9.6 m at the travel time of about 22 s.

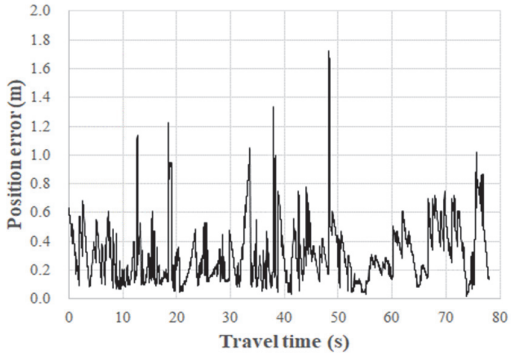


Fig. 19. Position error along evaluation travel time with optimally weighted cross-correlation functions. RMS error is 0.37 m. The maximum error is 1.75 m at the travel time of about 47 s.

Throughout this work, all software is written in C++ and Python for both Windows PC and Raspberry Pi 4B.

4. Conclusions

In this work, we presented a vehicle localization algorithm based on MEMS sensor data and demonstrated through evaluation tests on a real road that it can achieve highly precise position estimation. Our main contributions are summarized as follows.

1. We set up MEMS sensor data acquisition equipment operating at 50 Hz to calculate cross-correlation functions with a sufficient time resolution.
2. Among the nine cross-correlation functions we examined, the pitch rate, vertical magnetism, roll rate, and backward magnetism all played significant roles (74.7 %) in the vehicle localization. Among these types of sensor data, the pitch rate weight makes the largest contribution (24.4 %), which coincides with the fact that most terrain-based localization algorithms utilize pitch rate data as a major feature of roads.
3. After performing optimally weighted averaging of the nine cross-correlation functions using a noise elimination filter, the time lag estimation worked quite well; its estimation RMS error was 0.091 seconds.
4. The accuracy of the vehicle position estimation derived from the optimally weighted cross-correlation function outperformed that obtained from the equally weighted cross-correlation function.
5. Detailed position error analysis revealed that the position RMS error is 0.37 m and its maximum error is 1.75 m, both of which are far lower than our previous results.

Future extensions will involve combining a dead reckoning algorithm to stabilize the position estimation by eliminating the spike-like position errors and extending the algorithm to more general road networks by combining a map-matching algorithm [7]. We will also investigate the performance of the proposed algorithm when different vehicles and drivers are involved.

Acknowledgements

This work was supported by JSPS KAKENHI Grant number JP20K04723.

References

[1]. T. Yokota and T. Yamagiwa, Vehicle Localization by Optimally Weighted Use of MEMS Sensor Data, in *Proceedings of the 3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0 / 5.0 (ARCI 2023),* Chamonix-Mont-Blanc, France, Feb. 2023, pp. 79–84.

[2]. T. Yamagiwa and T. Yokota, Vehicle Localization Method Based on MEMS Sensor Data Comprising Pressure, Acceleration and Angular Velocity, in *Proceedings of the IEEE International Conference on*

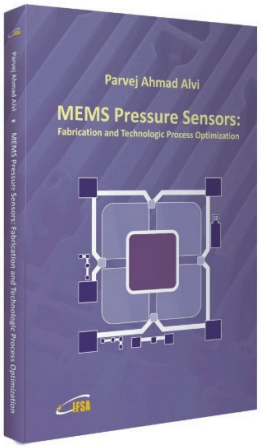
Mechatronics and Automation (ICMA), Guilin, Guangxi, China, 2022, pp. 786-791.
[3]. T. Yokota, Vehicle localisation by altitude data matching in spatial domain and its fusion with dead reckoning, *International Journal of Mechatronics and Automation,* Vol. 8, No. 4, Jan. 2021, pp. 208–216.
[4]. T. Yokota, Vehicle localization by dynamic programming from altitude and yaw rate time series acquired by MEMS sensor, *SICE Journal of Control, Measurement, and System Integration*, Vol. 14, No. 1, April 2021, pp. 78–88.
[5]. T. Yokota, Vehicle Localization Based on MEMS Sensor Data, in *Proceedings of the 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE' 2021)*, Sept. 2021, pp. 1094 - 1100.
[6]. T. Yokota, Localization Algorithm Based on Altitude Time Series in GNSS-Denied Environment, in *Proceedings of the SICE Annual Conference*, September 23–26, 2020, pp. 952–957.
[7]. T. Yokota, M. Okude, T. Sakamoto, and R. Kitahara, Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data, *IET Intell. Transp. Syst*., Vol. 13, Issue 11, 2019, pp. 1613–1623.
[8]. J. Tsurushiro and T. Nagaosa, Vehicle localization using its vibration caused by road surface roughness, in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, Yokohama, Japan, Nov. 2015, pp. 164–169.
[9]. A. J. Dean, R. D. Martini, and S. N. Brennan, Terrain-based road vehicle localisation using particle filters, *International Journal of Vehicle Mechanics and Mobility*, Vol. 49, Issue 8, 2011, pp. 1209–1223.
[10]. E. I. Laftchiev, C. M. Lagoa, and S. N. Brennan, Vehicle Localization Using In-Vehicle Pitch Data and Dynamical Models, *IEEE Transactions on Intelligent Transportation Systems,* Vol. 16, Issue 1, February 2015, pp. 206–220.
[11]. C. A. Gim, Inertial-Sensor-Based Non-Dead Reckoning Localization for Ground Vehicles, *IEEE Access*, Vol. 9, 2021, pp. 132766–132778.
[12]. X. Qu, B. Soheilian, and N. Paparoditis, Landmark-based localization in urban environment, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 140, June 2019, pp. 90–103.
[13]. https://www.boschsensortec.com/products/environmental-sensors/humidity-sensors-bme280/ (accessed on Jan. 27 2023).
[14]. https://www.u-blox.com/en/product/zed-f9p-module (accessed on April 6 2023).
[15]. https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/ (accessed on April 6 2023).

Published by International Frequency Sensor Association (IFSA) Publishing, S. L., 2023 (http://www.sensorsportal.com).

Parvej Ahmad Alvi

IFSA

MEMS Pressure Sensors:
Fabrication and Technologic Process Optimization



So far, no book has described the step by step fabrication process sequence along with flow chart for fabrication of micro pressure sensors, and therefore, the book has been written taking into account various aspects of fabrication and designing of the pressure sensors as well as fabrication process optimization. A complete experimental detail before and after each step of fabrication of the sensor has also been discussed. This leads to the uniqueness of the book.

Features include:
A complete detail of designing and fabrication of MEMS based pressure sensor.
- Step by step fabrication and process optimization sequence along with flow chart, which is not discussed in other books.
- Description of novel technique (lateral front side etching technique) in terms of chip size reduction and fabrication cost reduction, and comparative study on both the techniques (i.e. Front Side Normal Etching Technology and Front Side Lateral Etching Technology) for the fabrication of thin membrane.
- Discussion on issues of sealing of conical tiny cavity; because the range of pressure applied (i.e. greater or less than atmospheric pressure) can be decided by methodology of sealing of tiny cavity.
- A complete theoretical detail regarding aspects of designing and fabrication, and experimental results before and after each step of fabrication.

MEMS Pressure Sensors: Fabrication and Process Optimization will greatly benefit undergraduate and postgraduate students of MEMS and NEMS courses. Process engineers and technologists in the microelectronics industry including MEMS-based sensors manufacturers.

Hardcover: ISBN 978-84-616-2207-8
e-Book: ISBN 978-84-616-2438-6

Order: http://www.sensorsportal.com/HTML/BOOKSTORE/MEMS_Pressure_Sensors.htm