

# Indoor vehicle tracking with a smart MEMS sensor

Radosław Cechowicz<sup>1</sup>, and Marcin Bogucki<sup>1</sup>

<sup>1</sup>Lublin University of Technology, Faculty of Mechanical Engineering, Dept. of Automation, Nadbystrzycka 36, 20-618 Lublin

**Abstract.** Indoor navigation and vehicle tracking require special measurement techniques. The reference points and routes used by classic AGV (Automated Guided Vehicle) systems are usually buried under floor surface or painted directly on the floor, thus limiting the set of possible transportation paths. However, the indoor environment of an industrial warehouse is dynamic, the number and location of objects inside are subject to frequent changes and these changes might not be reflected in the map of the area. In such conditions, navigation according to the on-board instruments (dead-reckoning) could provide valuable information about the position and orientation of the vehicle. This paper reports test results from a smart sensor using a 6-axis MEMS IMU unit and a self-calibrating procedure for indoor vehicle orientation tracking. The smart sensor, integrated with information from wheel encoders can produce 2D position coordinates suitable for navigation. Original data processing algorithm, applied in the sensor, was developed by the authors as a part of the research project on mobile robotics.

## 1 Introduction

Indoor navigation is an important problem in modern manufacturing systems. Automated transport systems that make use of self-guided vehicles, such as AGV (Automated Guided Vehicle), require a network of orientation points to navigate through the production floors and storage areas. In classic AGV systems, markings used for vehicle navigation are located just beneath the floor surface or painted (or attached) directly on the floor [1]. Such systems have a number of limitations which include: poor path flexibility and scaling (expansion) properties, low resistance to wear and high maintenance costs [2]. In classic AGV applications, the transportation paths used by mobile units cannot be shared with the pedestrian traffic or human-operated machines, furthermore, the space cannot be allocated to other tasks (like temporary storage).

A modern idea of industry automation, referred to as industry 4.0, favours small autonomous entities that are able to communicate, exchange data and co-operate with other actors [3]. It resembles the idea of holons and holonic systems where real and virtual actors share resources and co-operate to complete a global task. Following this idea, the transportation system in the manufacturing environment would be composed of autonomous vehicles that could share their location and distribute tasks between themselves according to some cost criterion. The system should be flexible and easy to reconfigure.

Detaching the transportation units from fixed paths requires a new approach to navigation. The minimum-effort approach, typical of line-following solutions, would have to be replaced with navigational tools allowing some degree of vehicle autonomy. If the

transport routes are part of the shared space, then the autonomous vehicle should also be equipped with path planning and decision-making procedures so as to adjust to the dynamic change of the environment (obstacles, route closure, etc.).

Classical navigation is a process where the outcome from the on-board devices (dead reckoning; log) and reference measurements from external sources (fix) are used to provide information about the position of a vessel. In indoor mobile robotics the external reference might be the outcome from the processing (e.g. triangulation) of data obtained from optical (e.g. direction to beacon, image feature extraction and analysis), electromagnetic (e.g. Wi-Fi; wave amplitude or time of flight) or mechanical (e.g. ultrasound wave propagation time [4]) sources. On-board devices used for dead-reckoning include: rotary encoders (odometry, angle measurement [5, 6]), speed gauges (velocity measurement), inertial devices (acceleration, rotational speed [7, 8]) and optical devices (rotational speed). Strap-on inertial devices, such as microelectromechanical accelerometers and vibratory gyroscopes (MEMS), are a common choice for dead-reckoning systems, as they provide measurements independent of wheel encoders and can help improve accuracy [8–10].

An original MEMS-based orientation sensor with auto-calibration capability is presented further in the article. The sensor has proved itself useful in a typical navigational task encountered in industrial transport systems. The proposed solution, unlike continuous measurement systems [11], profits from the fact that data received while the vehicle is moving differs significantly from the stationary state and uses it for sensor recalibration (known as zero-velocity update [12]).

\* Corresponding author: [r.cechowicz@pollub.pl](mailto:r.cechowicz@pollub.pl)

## 2 Smart MEMS gyroscope

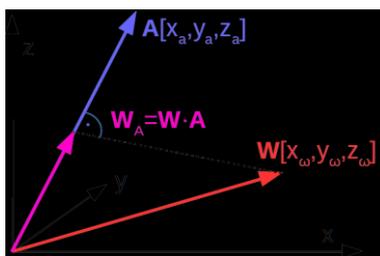
The properties of strap-on inertial gyroscope, as well as their application in industry, have been outlined in the work [13]. The smart sensor was a 6-axis IMU unit consisting of MEMS gyroscope and accelerometer (LSM6DS33). It was connected to the Odroid C2 single board computer (control unit) via the I2C interface, which was used for sensor data processing.

The intelligent sensor composed of the MEMS unit and the software was built for the floor (2D) navigation and had the following functions:

- it would be attached (strapped on) mechanically to a moving object (like a mobile robot or an industrial vehicle) in any orientation and in any position (no need for axis alignment),
- it would self-calibrate after power-on,
- it could detect rotation axis perpendicular to floor plane regardless of the IMU orientation,
- it could detect the stationary state (robot not moving) and use it for sensor recalibration (zero velocity update).
- it would broadcast the measured orientation angle (Yaw) through a standard interface (serial or Ethernet).

The initial self-calibration of the gyroscope was done as in the previous work [2]. The zero velocity detection was based solely on the signal from the gyroscope to enable the detection of movement events that were not triggered by the robot actuators, such as hits and collisions with external objects.

The primary use of the 3-axial accelerometer integrated into the sensor was to find the direction of the axis perpendicular to the floor plane.



**Fig. 1.** Arrangement of accelerometer and gyro axes.

Assuming that the mobile robot is not moving and no external force, except gravity, is acting on it, the vertical direction (A in Fig.1) can be obtained directly from the accelerometer readings:

$$\mathbf{A} = [x_a, y_a, z_a] \quad (1)$$

where:

$[x_a, y_a, z_a]$  are readings from accelerometer (axis x, y, z) in SI units.

Vector A can be normalised:

$$\hat{\mathbf{A}} = \frac{\mathbf{A}}{\|\mathbf{A}\|} \quad (2)$$

The readings from a 3-axial gyroscope make up vector W (Fig.1), which represents the momentary rotational speed of the sensor in the 3D space:

$$\mathbf{W} = [x_\omega, y_\omega, z_\omega] \quad (3)$$

where:

$[x_\omega, y_\omega, z_\omega]$  are readings from gyroscope (axis x, y, z).

The length of vector W can be obtained from:

$$\|\mathbf{W}\| = \sqrt{x_\omega^2 + y_\omega^2 + z_\omega^2} \quad (4)$$

Assuming that the sensor is strapped to an object moving on a horizontal floor, the rotational speed of the object can be obtained by finding the projection of vector W in the direction A:

$$\mathbf{W}_A = \mathbf{W} \cdot \hat{\mathbf{A}} \quad (5)$$

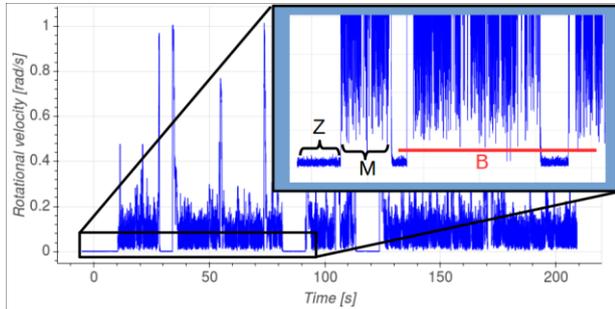
This works with any orientation angle, which is why the sensor can be strapped-on to the object at any position [1]. The limitation of this method is that the mobile robot must be placed, in its normal position, on the horizontal floor and must remain still during the initial calibration step. Initial self-calibration and detection of the vertical axis are done once, at the beginning of measurements. It is assumed that the vertical direction does not change while the robot is moving.

The bias of the MEMS gyroscope is changing during its operation ([14, 15]). The Allan variance test performed for a similar gyroscope ([2]) has shown that the angle random walk (ARW) was 0.011 deg/s<sup>1/2</sup> and the bias stability was 3.58 deg/h. The expected angular error is then about 40 deg/h (for averaging time equal to 1s), up to 360 deg/h (for averaging time 0.01s). It can be substantially decreased, as shown in the work [16], with a properly designed noise model and filters. However, in a sensor designed for 2D navigation in the industrial environment, the measurement error from bias change can be decreased with a technique called zero-velocity update (ZUPT [17]). This technique uses periods of time at which the vehicle is not moving to re-calibrate the sensor (update the sensor bias) and is used for enhancing the performance of automotive GPS/INS navigation systems ([12,18,19]), as well as indoor pedestrian tracking ([20–22]). Since a common movement pattern of industrial AGV includes stops and idle times either for control reliability ([23]) or because of resource constraints ([24, 25]), the sensor can be re-calibrated during normal operation of the vehicle using this solution. The recalibration algorithm in the smart sensor was based on the detection of zero-velocity state (lack of movement) from the analysis of the signal from all gyroscope axes.

It was shown in [2] that the amplitude of the signal from gyroscope can be used for detection of movement of SCARA robot arm. To determine whether a similar measure can be used for detecting the movement of a mobile robot, a series of experiments was performed. The robot, with the sensor attached at different angles, was following a test path similar to the one used in the experiment described in the work [1].

The analysis of the 3D gyroscope output has shown that the length of vector W (3) can be used for robot movement detection. The length of W (obtained using formula (4)) was significantly greater while the robot was moving (Fig. 2) than while it was stopped. Furthermore, the range histograms of  $\|\mathbf{W}\|$  (4) during the idle state (stopped) and during movement were detached for most tested sensor orientations (Fig. 3). A clear boundary (marked as B in Fig. 2 and in Fig. 3) could be drawn between the highest value of  $\|\mathbf{W}\|$  measured during robot stop (zone marked as Z in both figures) and the lowest value of  $\|\mathbf{W}\|$  measured during movement (zone marked as M). As it is shown in Fig. 4, a single

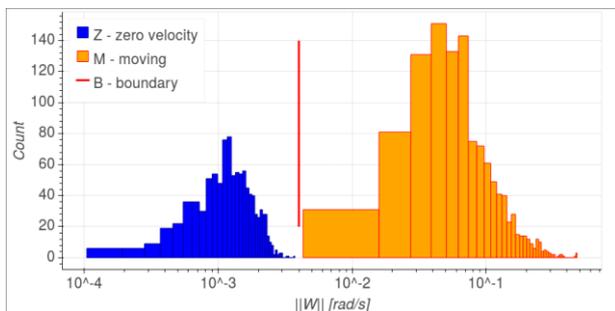
axis gyro measurement could not be used for motion detection because of large parts of the histograms for M and Z overlap. The measure  $\|W\|$  proved much more sensitive for robot motion detection than the single-axis measurement because the sensor could pick up horizontal and vertical vibrations from the motor and wheels during the movement of the vehicle.



**Fig. 2.** The length of  $W$  (3) during robot movement (segment M) and during vehicle stop (segment Z). A boundary B can be drawn between segments Z and M.

The histograms shown in Fig. 3 were made of data from initial sensor calibration (histogram Z) and 15 seconds of straight robot movement shown in Fig. 2 (segments Z and M). As the histograms are detached, a clear boundary value B could be used for movement detection.

The histograms shown in Fig. 4 were made of the same amount of data but the reading  $z_\omega$  from gyroscope axis Z was used instead of  $\|W\|$ . The detection of movement was still possible but the quality of the classification was poorer.



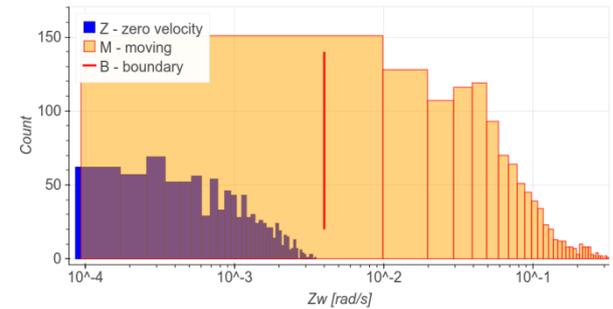
**Fig. 3.** The histograms of rotation vector length  $\|W\|$  during robot movement (M; right) and during vehicle stop (Z; left). B is the boundary value that separates both histograms. The horizontal (length) axis is logarithmic.

The movement detection procedure based on the above findings (using  $\|W\|$  as the norm) was included in the smart-sensor software. The recalibration of the sensor was performed if a detected period of robot stop was longer than  $T_i$ . Time  $T_i$  was chosen arbitrarily, based on the data from the experiments, and amounted to approx. 3 s during all test runs shown below.

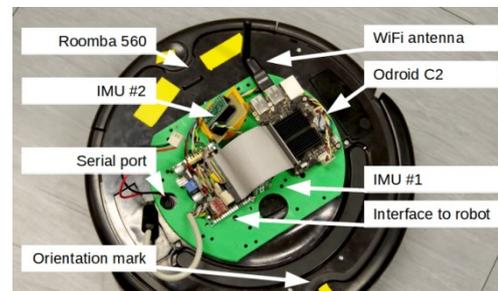
### 3 Verification of the Sensor Performance

The intelligent strap-on sensor was mounted on a modified iRobot Roomba robot (shown as IMU#2 in

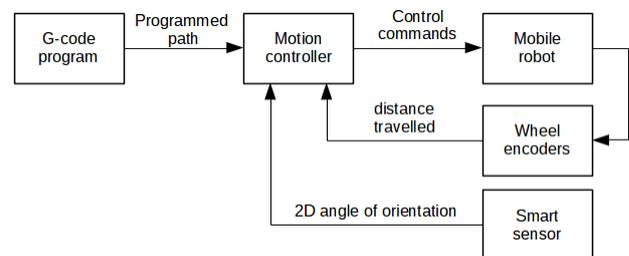
Fig. 5). The sensor was providing the angle of attitude feedback to the movement controller mounted on the robot. Linear translation of the robot was derived from the odometry data from robot wheels. Control commands and the feedback data from the wheel encoders were passed through the standard robot serial port. The Odroid C2 was used as the main controller and data processing unit for the IMU. The second sensor (shown as IMU#1) was used only as a reference.



**Fig. 4.** The histograms of rotation vector length  $Z_w$  during robot movement (M; right, transparent) and during vehicle stop (Z; left). B is the boundary from the previous figure drawn for reference. The horizontal (length) axis is logarithmic.



**Fig. 5.** Modified iRobot Roomba used in the experiment.



**Fig. 6.** Scheme of robot control during the experiments

The scheme of control during the test runs is shown in Fig. 6. The motion controller, which was a process running on the C2 computer, was reading the required path from a G-code program file.

The actual position of the robot was determined from dead-reckoning, using wheel encoders to find the distance travelled, and the smart sensor output to find the actual direction of movement. Wheel encoder counters were read approximately 3 times per second. The distance travelled since the last update  $dL$  was determined from the equation:

$$dL = \frac{dX_L + dX_R}{2} \quad (6)$$

where:

$dX_L, dX_R$  were the distances travelled by the left and

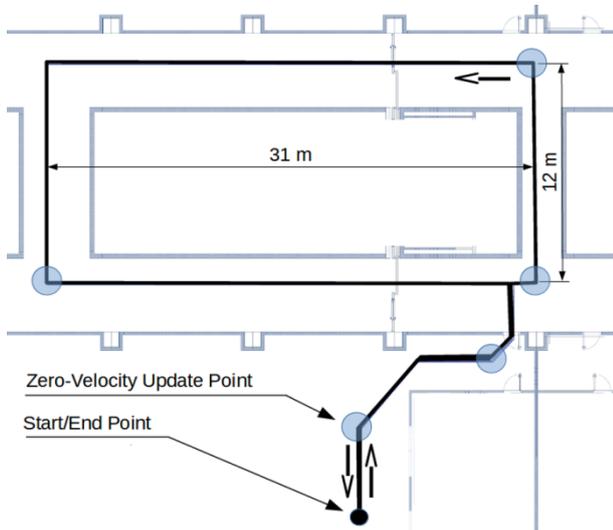
right wheel respectively (number of counts multiplied by rate ratio).

It was assumed, that there was no wheel slip during the test runs and certain countermeasures (limited torque, high friction floor surface) were taken to prevent this.

Wheel encoders were not used to determine the orientation of the robot. This method is known for poor accuracy and the information that could be extracted was unsuitable for navigation (as shown in Fig. 8). During the test runs the robot orientation control was relying solely on the MEMS gyroscope (smart sensor).

Standard control commands, defined in the robot interface, were used to control the movement. Robot position was updated 3 times per second.

The pre-programmed path (stored in the G-code file) is shown in Fig. 7. The path was designed to test the performance of navigation during long stretches of straight line movement, typical for industrial storage area environment, where accurate measurement of the orientation angle is important to keep the robot on track. During the test run, the robot was moving with a speed of approximately 70% of its maximum, covering the distance of 120m. A single test run took about 600 s. A series of 5 s stops, marked with blue circles in Fig. 7 was programmed to provide conditions for the zero-velocity update of the gyroscope.



**Fig. 7.** Programmed robot path during the tests (black line) on the contour map of the laboratory floor (blue). The image is not to scale.

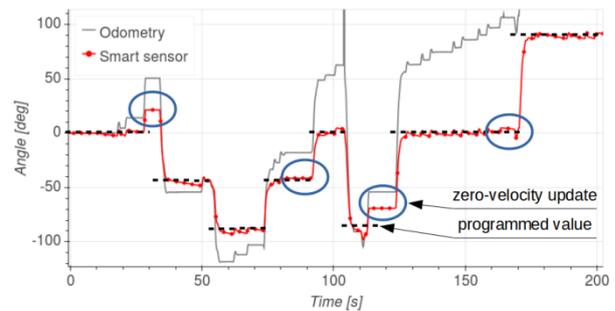
**Table 1.** The positioning accuracy after the test runs.

Test no	Absolute error (x,y) [m]	Distance error [m]
1	0.16, 0.15	0.22
2	0.05, 0.40	0.40
3	0.01, 0.44	0.44
4	0.09, 0.10	0.13

The results of robot positioning accuracy, shown in Table 1, were taken at the endpoint, after completing the programmed loop (Fig. 7). The absolute error was

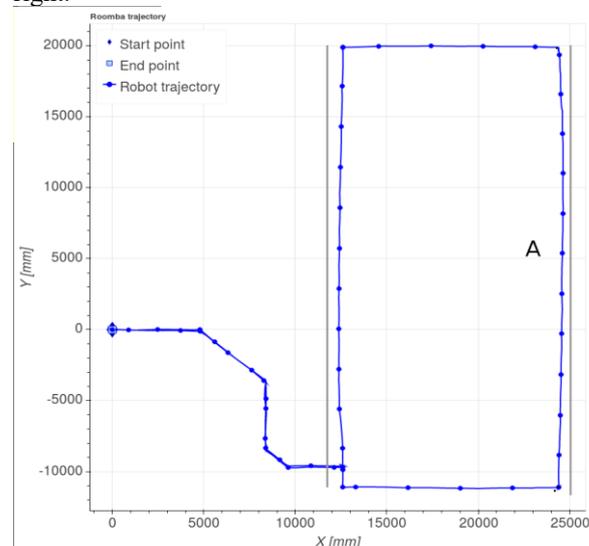
measured from the starting point along each coordinate. The distance error was the distance from the start point.

A comparison of the robot orientation angle produced by the smart sensor and the angle obtained from the odometry for the first 200 s of the test run is shown in Fig. 8. Using the output from the gyroscopic sensor allowed keeping the robot close to the programmed path in spite of the obstacles on the floor and the poor synchronisation of its motors (robot had a tendency to turn to the right). The test results show that the designed sensor can provide valuable readings during short-term incremental navigation.

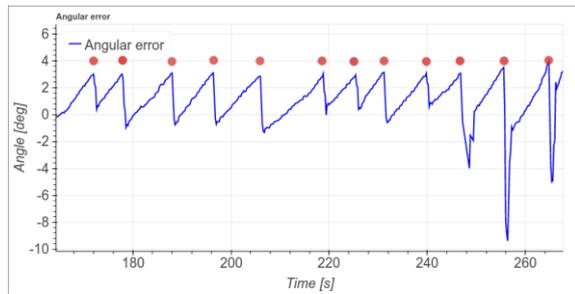


**Fig. 8.** The orientation angle from smart sensor (IMU) compared to orientation angle obtained from odometry (ODO) for the first 200s of the test run.

The maximum deviation was observed during the long straight line legs of the programmed path. As it is shown in Fig. 9, on long 31 m straight line legs, the robot was following a curved trajectory, with the maximum off-track error of approx. 300mm. However, at the end of these legs, the error was reduced to less than 100mm, which proved, that the main source of the error was within the robots' own control system and differences in effective wheel diameters. It was difficult to keep robot on track because of its natural tendency to turn to the right.



**Fig. 9.** Robot trajectory as seen by the motion controller (dead-reckoning position tracking). Dots are drawn every 50 position updates, the distance between dots is proportional to the speed of the robot. Drawn with gray colour are straight lines to provide the reference to the track curvature.



**Fig. 10.** Angular error during the movement along the long leg of the programmed path marked with “A” in Fig. 9. Red dots mark the execution of corrective action triggered by the C2 controller.

The plot of orientation error during the first straight line part of the track (marked with A in Fig. 9) is shown in Fig. 10. The rate of uncontrolled orientation change of the robot during straight-line motion was about 0.4 deg/s. The C2 controller was programmed to trigger corrective action (send a command via the robot’s serial interface; marked with red dots in Fig. 10) when the threshold value (3 deg) was exceeded. The correction of lower errors was not possible due to the lack of floating point capability of the robot on-board controller.

#### 4 Conclusions and Further Research

The smart sensor proved to be a suitable solution for short-term incremental navigation (dead reckoning) of the mobile robot moving on an even, horizontal floor. It was assumed, that the readings from wheel encoders were accurate, i.e. there was no wheel slip. The problem of detecting and handling wheel slip situations was outside of the scope of the experiment as the sensor was to measure the orientation of the robot body.

The measurement procedure for the orientation angle was designed with the assumption that no rotation other than around the vertical axis, parallel to Earth's gravity vector, is important. The sensor self-calibrated after power-on and automatically recalibrated during detected periods of no movement.

The motion detection procedure used in the sensor is rudimentary; the sensor cannot correctly detect rotation rates that overlap with the idle state, although the use of the  $\|W\|$  norm, defined by equation (4), is more selective than a similar norm using only one axis.

Further research will concentrate on the improvements of the motion detection procedure as it is crucial for accuracy and practical applications. To make the sensor more usable in the industrial environment, the software will be updated to facilitate proper angle measurements on ramps and rough surfaces where the rotation is made around the axis which is not parallel to the vertical direction.

#### References

1. R. Cechowicz, ITM Web of Conferences 15, 05010 (2017).
2. R. Cechowicz, Acta Mechanica et Automatica 11 no.2, 104 (2017).

3. J. Lee, B. Bagheri, and H.-A. Kao, Manufacturing Letters 3, 18 (2015).
4. P. Stączek, ITM Web of Conferences 15, 05008 (2017).
5. D. Herrero-Pérez, J. J. Alcaraz-Jiménez, and H. Martínez-Barberá, Int. J. Adv. Rob. Syst. 10, 292 (2013).
6. J. Borenstein and L. Feng, IEEE Trans. Rob. Autom. 12, 869 (1996).
7. N. Houshangi and F. Azizi, in 2006 World Automation Congress (2006), pp. 1–8.
8. B. Gersdorf and U. Freese, in ICINCO (1) (2013), pp. 388–395.
9. J. Borenstein, A. Borrell, R. Miller, and D. Thomas, in Unmanned Systems Technology Xii, edited by G. R. Gerhart, D. W. Gage, and C. M. Shoemaker (Spie-Int Soc Optical Engineering, 1000 20th St, Po Box 10, Bellingham, Wa 98227-0010 USA, 2010).
10. S. K. Hong, S. C. Leev, J. W. Han, and Y.-S. Ryuh, Int. J. Control Autom. Syst. 10, 1000 (2012).
11. S. K. Hong and S. Park, Sensors 8, 7287 (2008).
12. D. A. Grejner-Brzezinska, Y. Yi, and C. K. Toth, Navigation 48, 216 (2001).
13. V. M. N. Passaro, A. Cuccovillo, L. Vaiani, M. D. Carlo, and C. E. Campanella, Sensors 17, (2017).
14. D. Bhatt, P. Aggarwal, P. Bhattacharya, and V. Devabhaktuni, Sensors 12, 9448 (2012).
15. H. Hyyti and A. Visala, International Journal of Navigation and Observation 2015, 1 (2015).
16. A. G. Quinchia, G. Falco, E. Falletti, F. Dovis, and C. Ferrer, Sensors 13, 9549 (2013).
17. J. R. Huddle, in Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290) (1998), pp. 63–73.
18. W. Meiling, F. Guoqiang, Y. Huachao, L. Yafeng, Y. Yi, and X. Xuan, in 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES) (2017), pp. 103–108.
19. K.-W. Chiang, T. T. Duong, and J.-K. Liao, Sensors 13, 10599 (2013).
20. W. Yang, C. Xiu, J. Zhang, and D. Yang, Sensors 17, (2017).
21. M. Ren, K. Pan, Y. Liu, H. Guo, X. Zhang, and P. Wang, Sensors 16, (2016).
22. L. Zheng, W. Zhou, W. Tang, X. Zheng, A. Peng, and H. Zheng, Simulation Modelling Practice and Theory 65, 45 (2016).
23. T. Ganesharajah, N. G. Hall, and C. Sriskandarajah, Ann. Oper. Res. 76, 109 (1998).
24. T. C. E. Cheng, Int. J. Oper. Prod. Manage. 7, 62 (1987).
25. N. Kousi, S. Koukas, G. Michalos, and S. Makris, Int. J. Prod. Res. 1 (2018)