Omni-Directional Wheeled Autonomous Guided Vehicle Position Estimation Using Sensor Fusion

Jose Alfron Benedicto R. Batiles1, Calvin Alexander Y. Ng2, Francis Gregory L. Ng3

Mechanical Engineering Department

De La Salle University - Manila

Malate, Manila

jose\_alfron\_benedicto\_batiles@dlsu.edu.ph1, calvin\_alexander\_ng@dlsu.edu.ph2, francis\_gregory\_ng@dlsu.edu.ph3

**Abstract— The use of sensor fusion techniques to improve the localization of an omni-directional autonomous guided vehicle (AGV) was explored in this paper. The researchers combined the use of Encoder and the IEEE 802.15.4a to improve the position estimation of the mobile robot and used omni-wheels to improve its maneuverability and control. The sensors were used one at a time on the AGV as a sort of baseline for the performance of single-sensor use on the robot. Sensor fusion was performed through a Particle Filter Algorithm to improve the localization of the omni-directional AGV. Simulated tests and actual test runs were performed to verify the performance of the sensor fusion with AGV. The robotic design produced a higher performing omni-directional AGV that can be produced economically due to the availability and affordability of the components used.**

***Keywords— sensor fusion, omni-wheel, particle filter, Autonomous guided vehicle***

NOMENCLATURE

*Variables*

State vector of the particle

posterior density

unnormalized importance weight

normalized importance weight

noise

Proportionality factor for the signal strength and distance squared of the first and second WSN transmitter

measurement of the sensors

x-component of the particle position

y-component of the particle position

velocity of the particle

estimation of the actual state

measurement prediction based on particle state

history of sensor measurements

*Subscript*

particle

iteration

w process noise for the time update

v measurement noise for the weight update

1. INTRODUCTION

Manipulators that could replace human labor in performing repetitive or harmful work has been developed and widely searched for use in the industrial sector [1]. A significant reason for this proliferation of manipulators was because of the safety to humans and the accuracy and reliability of this approach. Effective manufacture systems that use mobile robots has been the development of the industry for the past years. Automated guided vehicles (AGV) are an example of these mobile robots that are wheel-based and can follow a path automatically; thus, AGVs have a simple development and control process [2]. AGVs have found uses in warehouses, distribution centers, and manufacturing plants for providing transport to large and heavy transport units for an automated internal material flow within the facility. Current research in the development of AGVs has been towards the affordable production of small AGVs for use in facility logistics [3].

An improvement to the design of AGVs would be the utilization of omnidirectional wheels as it has gained popular use in mobile robotics [4]. It is called omnidirectional because the wheel possesses three degrees of freedom in its plane of movement compared to the typical wheel design [5]. This is made possible through the design of the wheel that uses small wheels embedded in the construction of the omnidirectional-wheel. The design of the wheel allows it to freely slide in the direction of the axis of rotation [1]. An example of an omnidirectional robot can be seen in Figure 1. There have been several designs for omni-directional wheels that have been put into use in previous literature, such as Mecanum wheels, Swedish wheels, Ball-bot, and Omni-wheels. Mecanum wheels were used in a research that created an omnidirectional mobile robot with established motion control rules and a kinetic model of the model that had six degrees of freedom [5]. The Swedish wheels were used to create a floor cleaning robot that had proximity and ultrasonic sensors that gave the robot the ability to longitudinally and laterally while avoiding obstacles and impacts [6].



Fig. 1. Omnidirectional mobile robot [3]

Ballbots balance themselves on balls with only a single point of contact to the ground. Brush motors are used in conjunction with omni-wheels that make contact with the ball. A ballbot was designed with three to four actuators, omni-directional wheels, Bluetooth model to monitor the robot’s behavior, and a gyroscope and sensor to quantify its balance on the ball [7]. Omni-wheels are the last kind of omnidirectional wheels and were used to create an industrial manipulator similar to a Nokia PUMA 560 with and encoder and a DC drive gear for localization, and a three-wheeled omnidirectional mobile robot that uses a capture camera for localization [1]. The omni-wheel was the wheel of choice due to the availability and range of applications of the wheel.

For AGVs to traverse a course and accomplish its intended task, localization is required. Localization is defined as the position of the robot and can be classified either as global localization or as local localization. Global localization uses a work space as a reference frame for quantifying the absolute position of the mobile robot. On the other hand, local localization simply measures the position of the robot relative to a marker or known position. AGV’s either use global localization or a fusion of local and global localization [2]. The localization is achieved through sensors that translates physical data input into data signals that the processor can interpret. Sensor Fusion is an approach to localization that integrates the use of two complimenting sensors to improve localization. Several sensors and sensor fusion methods have been used across various studies for localization.

The first sensor fusion method used Wireless Sensor Networks (WSN) and Laser Range Finders. WSN, such as the IEEE.802.15.4a, are inexpensive and flexible for use in different applications but has relative low accuracy for localizing mobile robots, while Laser Range Finders can detect landmarks and possesses high accuracy compared to WSN; hence, the localization was accomplished through using the WSN as the global localization and the Laser Range Finders as the local localization [3]. An example of a WSN is provided in Figure 2. The second sensor fusion method used High Frequency Radio Frequency Identification Technology (HF-RFID) and Odometry. Passive Radio Frequency Identification (RFID) technology are commonly used for logistics, but the transponders can act as artificial landmarks to achieve self-localization. High Frequency RFID transponders were sensor fused with the odometry by using the wheel encoders to move from one RFID transponder to the next one [8].

The third and last sensor fusion method used a StarGazer localization sensor fused with an encoder. The localization sensor analyzes reflected infrared ray images that emanate from a passive landmark. Global localization can be attained; however, the misrecognition of landmarks causes errors in movement. Thus, the sensor fusion was performed with the encoder since it has no moving errors but accumulates errors [9]. All the aforementioned sensor fusion methods utilized Mecanum wheels in their prototypes; hence, the error due to slipping of the omnidirectional wheels has been addressed through the combined sensor application. For this study, WSN sensor fused with wheel encoders was used to resolve the accumulation of errors in the encoder and the relative low accuracy of the WSN method.

To achieve sensor fusion, sensor fusion algorithms were employed. The WSN sensor’s None-line-of-sight measurements and multi-path fading generate noise; thus, Bayesian filter is used to estimate the localization of the mobile robot through the probability density functions that provide a model for the residuals [3]. Kalman Filter was derived from the Bayesian Filter and optimally estimates the system state by repeatedly operating on the noise from the input data. Applications of this filter are normally for vehicular control and navigation [9]. On the other hand, the Particle Filter is also based on the Bayesian Filter, but this approach accounts for uncertainty in position and can function with non-Gaussian motion and sensor errors. The advantage of this filtering algorithm is that it can deal with global localization and position tracking [3]. In the Particle Filter approach, the required posterior density function is represented by a set of weighted particles that capture the posterior mean and covariance for any nonlinearity through true nonlinear system. There is significance in the number of particles considered to the performance of the algorithm [10]. However, either the estimation accuracy or computational load can be prioritized by simply manipulating the number of particles [11]. The Monte Carlo Particle Filter was used to account for the noise in the WSN and sensor fuse this sensor to the Encoder.

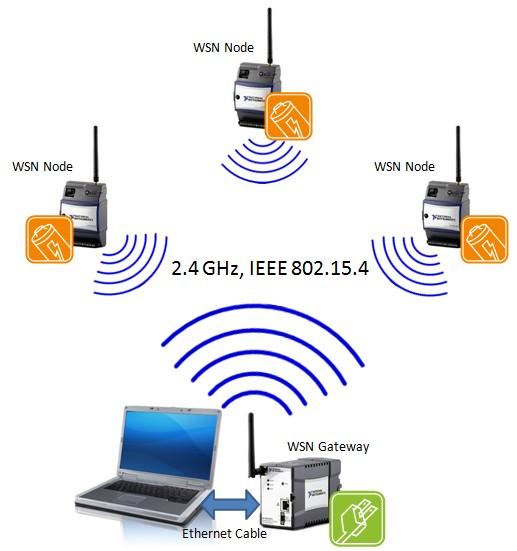


Fig. 2. Wireless Sensor Network [12]

The main objective of this study is to create a prototype to create an affordable design for a prototype omni-directional wheeled autonomous guided vehicle that can navigate around an indoor environment through position estimation using sensor fusion. An encoder used in measuring the rotation of the wheels was sensor fused with a WSN to create a highly accurate localization with minimal error due to drifting. The sensor fusion was accomplished through the use of a Monte Carlo Particle Filter. To achieve the main objective, the algorithm for the sensors in use were generated with reference to the Kinematics of an Omni-Wheel and the noise generated in the WSN sensor. The AGV was then designed and simulated the localization with the sensors individually, and then fused. The results were then compiled and analyzed. This can be seen in the following sections.

1. SENSOR FUSION ALGORITHM

The foundation of probability densities representation through particles is the Monte Carlo integration [13]. A multidimensional integral can be evaluated using the following equation:

By creating a significant number of independent samples while distributing them according to and evaluating weighted sum, an estimate of I using Monte Carlo’s approach to integration can be computed. This is expressed in the equation below.

The desired density normalizing factor cannot be identified; hence, normalization of the importance weights must be performed to attain this value. The importance weights can be expressed by the following equation:

Performing the normalization, the normalized importance weights can be defined by the following equation.

Thus, the Monte Carlo estimation of I can be expressed using the normalized importance weights with the following equation.

1. *Initialization*

Without knowing the actual behavior of the system, the uniform distribution is an acceptable initialization for the particles. The minimum and maximum values for the x and y components for the particle position values are the uniform distribution limits. The range of the particles could be improved if the position range is reduced; thus, providing better initialization results. For the initial velocities for the x and y components, two uniform distributions to generate dedicated particles for each component shall be used. After the end of the initialization phase, the succeeding steps enter into a loop [14]. Hence, the state vector for each particle is defined by the following equation.

1. *Measurement Update*

Majority of the sequential Monte Carlo filters have based their algorithms around the concept of sequential importance sampling which is mostly used to execute nonlinear filtering. Essentially, the required posterior density function can be represented through a random sample set with associated weights and computing estimates according to these samples [13].

Ideally, the samples should be extracted from the probability distribution which defines the actual position of the robot for robot localization; however, it is an unknown distribution state. On the other hand, the alternative probability distribution is known because it is the probability distribution for the measurements provided by the sensors.

A specific case for sequential importance sampling is taken into consideration where the importance density is dependent only on the previous state of the particle and measurement of the sensors. This case will be the assumption followed for the algorithm; thus, the modified equation for determining the weights can be defined by the following equation [13].

The transitional prior has gained popularity in use as the suboptimal choice with an additive zero-mean Gaussian process noise model considered for the nonlinear state dynamics.

The modified weight equation would result in higher weights being attained for particles that would lead to similar measurements from the sensors. For each weight in an iteration, the likelihood function would be multiplied to the previous weight and the difference between the real measurement from the observations introduced into the system and the predicted measurement. This would result in the following equation [14].

The predicted measurements for the incorporation of the WSN position of the AGV with respect to the transmitters in the corners of the room and the rotation of the wheels for the encoders is defined by the following equation and must be accomplished for each particle.

The normalization of the weights are attained after the weights for all the particles are computed by using the following equation:

1. *Estimation*

Upon the completion of the computations for the modified weights, the approximations for the posterior probability function can be computed using the following equation.

The estimation for the state is calculated using the weighted mean value.

1. *Resampling*

The performance of the filter is directly influenced by the resampling algorithm used. There have been several researches made in this field with certain criteria making the algorithms an effective choice. The algorithm should have many properties and select higher probability particles while still including sufficient lower probability particles for the filter to detect nonlinear behavior.

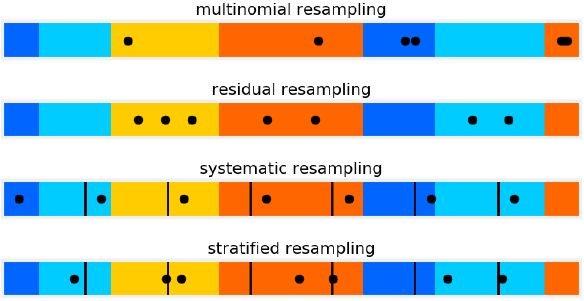


Fig. 3. Performance of different resampling algorithms

There are typically four types of resampling algorithms: multinomial, residual, stratified, and systematic resampling. Multinomial resampling is the conceptually easiest algorithm to use but not too efficient. It computes for the cumulative sum of the normalized weights and provides an array of increasing values from 0 to 1. Uniformity of the samples is an issue with the multinomial resampling algorithm; fortunately, there are algorithms that have properties that exhibit better uniformity among its samples.

Residual resampling is such an improvement over multinomial resampling in this regard with guaranteed uniformity across its sampling of its population of its particles. There is also the added benefit of improved run time with this algorithm. The algorithm works by multiplying the normalized weights by *N* and then taking the integer value of each weight to determine the number of samples of that particle would be taken where the residuals would be the fractional component of the weights. The remaining particles would be chosen by using a different sampling scheme on the residuals.

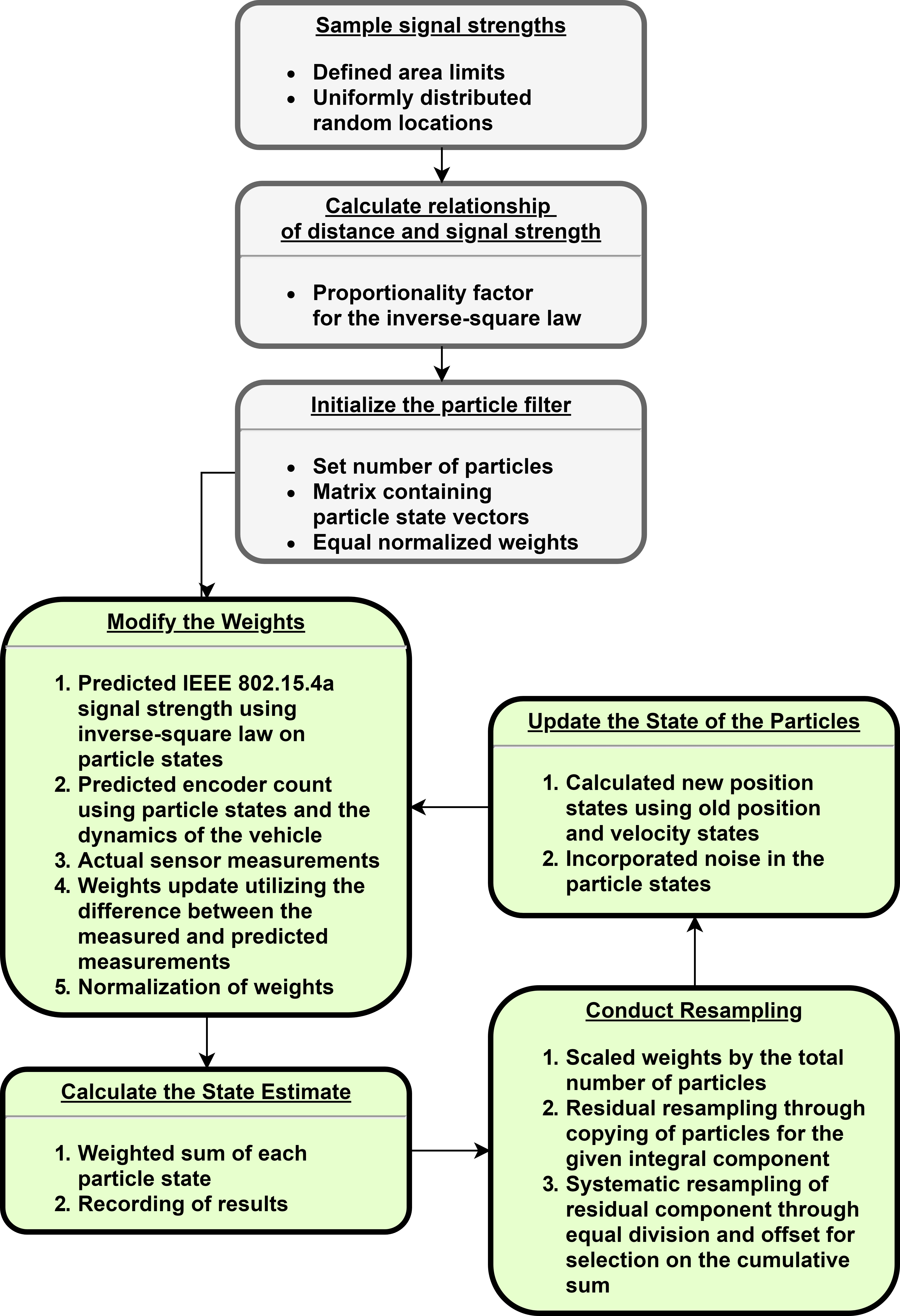
Stratified resampling divides the cumulative sum into *N* equal sections to select a particle randomly from each section and results in relatively consistent selections. Systematic resampling also divides the space into *N* divisions with a random offset chosen for all of the divisions to ensure that each sample is evenly spaced from one another. The performance of these two resampling methods are good where systematic resampling ensures all parts are proportionally sampled while stratified resampling guarantees that higher weights get higher resampling exposure compared to systematic. The performance of each resampling algorithm can be seen in Figure3. For the algorithm used in this paper, residual resampling is used with systematic resampling performed on the residuals.

1. *Time Update*

The predictions for the particles in the next time step through the proposal distribution or the dynamic model which is defined by the equation below where the next iteration of the state vector for each particle.

where,

Fig.4. Schematic Diagram of the Particle Filter Algorithm



sds

The new particles would be evaluated in the next iteration, but each particle must have the process noise added to it due to the noise probability function to generate the next iteration. The steps for the particle filter algorithm can be summarized in Figure 4 as seen below.

1. AGV DESIGN

Presented in this section of the paper is the schematic of the prototype to be used for the production of the AGV together with the components and their corresponding properties that were used for the prototype and taken into consideration for the simulations. The components used for the prototype are a 9V rechargeable battery, the motor controller, microprocessor with the raspberry pi, 3 omni-wheels with encoder, 5V DC-DC converter, 3 zigbee transmitter-receiver, NXT motor, I293 h-bridge, and miscellaneous connectors. The converter was used to convert the 9V battery voltage to a stable 5V for the raspberry bi.

1. *Schematic Diagram for the omnidirectional AGV*

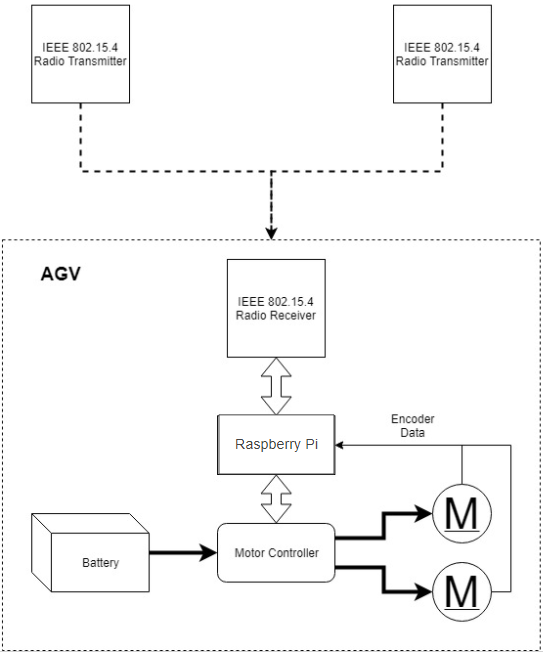


Fig. 5. Schematic Diagram of AGV

Fig. 5 shows the flow of the signals from the environment to the controller. The WSN transmitters will broadcast a signal that will be caught by the receiver. This information is processed by the controller that has the sensor fusion algorithm. This sends a signal to the motor controller that adjusts the rotation of the wheels that are monitored by the encoder. The encoder data is fed back into the controller to form a closed loop.

1. *Implementation of the AGV Design*

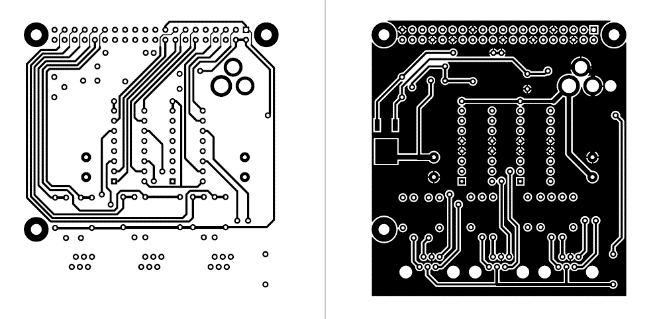


Fig. 6. Motor Driver Shield for Raspberry Pi

Considering the schematic diagram presented above, the components and circuit boards that were specifically used for this mobile robot together with the specifications are presented in the succeeding section. As shown in Fig. 6 is the design of the motor driver shield that was made to adjust the power entering the motor. The power being provided into the system is from a 9V rechargeable battery that could be bought in any convenience store.



Fig. 7. Omni-wheel for NXT compatible hub [15]

The Omniwheel for NXT compatible hub is used to provide omnidirectional movement to the AGV. For this specific application, the wheel has a diameter of 48 mm with 2 plates and 8 rollers where the rollers are perpendicular to the axis of rotation of the wheel [15]. The wheel can withstand a load capacity of 2 kg, which is more than enough for the application taken into consideration for this study.



Fig. 8. NXT Lego Motor [16]

The motor directly attached to the omniwheel is the 9V NXT motor as seen in Figure 8. The motor comes equipped with a built-in encoder that will serve as the second sensor for the sensor fusion. With a resolution of 1°, the rotation encoder can accurately monitor the rotation of the shaft. When the motor is loaded with 0.115 Nm and operated at 9V, the efficiency of the motor’s power control is at 70% of the unloaded performance of the motor. The 9V source is connected using a cable adapter to drive the motor. The curve presented in Figure 9 is the mechanical power vs torque curve of the motor with the maximum mechanical power attained at 0.15 Nm [17].

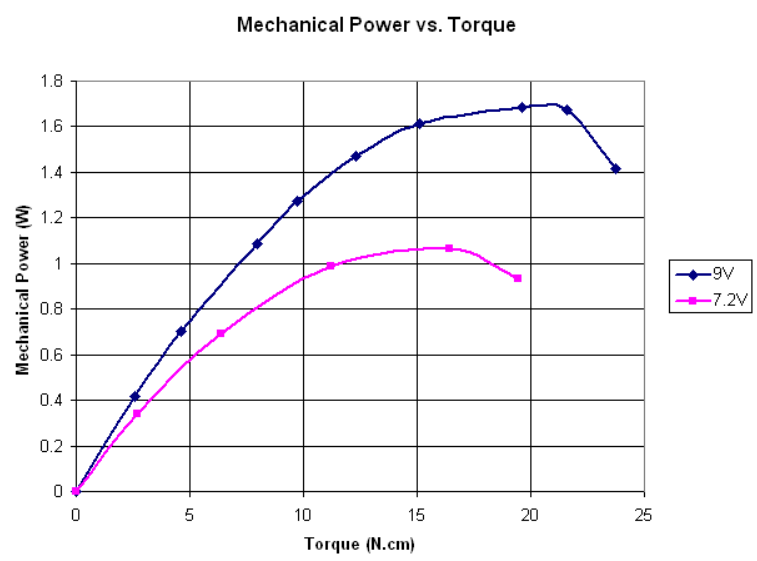


Fig. 9. Mechanical Power vs. Torque Curve of the NXT motor [17]

Other than the encoder present in the NXT Lego motor, the WSN can be found in the Arduino Zigbee Shield. Anny Arduino standard footprint development board can interface directly with the Zigbee Shield thanks to the shield form-factor that also equips the board with the wireless communication abilities. Other features of the board is that it directly regulates power from the 5V source into the 3.3V power used in the system. It also has on-board MOSFET level shifting. Any digital pin on the Arduino can be connected to the DIN and DOUT pins of zigbee shield. The presence of indicator LEDs for Power, DIN, DOUT, RSSI, and DIO5 ensures the user of the status of the board together with a reset button. Lastly, there is a 9x11 grid of 0.1” spaced prototyping holes. Three of the Arduino Zigbee shields were used for the research, two serving as transmitters placed in the corners of the room and one serving as the receiver on the AGV [18]. The Arduino Zigbee Shield used for the prototype can be seen in the figure below.

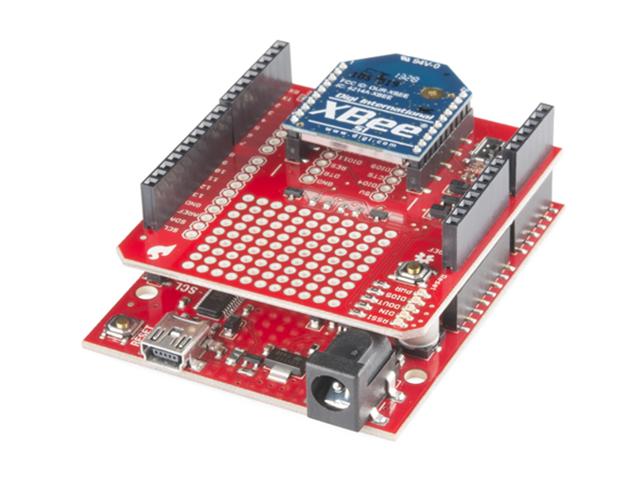


Fig. 10. Arduino Zigbee Shield [18]

With the sensors to be used for the sensor fusion identified, the processor for the particle filter algorithm is all that remains. The Raspberry Pi B+ was chosen because it is more capable than a typical Arduino because of its high processing power and clockspeed. Other features prevalent in the board is the presence of significantly more GPIO pins. There are 40 pins with the first 26 pins retaining the same pinout. There are also more USB ports where there are o 4 USB 2.0 ports with improved hotplug and overcurrent behavior. There is less power consumed by using switching regulators to reduce power consumption by 0.5 to 1 W. An improved form factor with a more aligned USB connector to the board edge and four squarely-placed mounting holes added to the board. Improved audio circuit with a low-noise power supply and repositioned composite video onto the 3.5mm jack is present. The storage of the board is now a more user-friendly push-push micro SD instead of the previous friction-fit SD [19]. An example of the Raspberry Pi B+ can be seen in Figure 11.



Fig. 11. Raspberry Pi B+ [19]

1. EXPERIMENTAL TESTS

To determine the improvement of the omnidirectional AGV performance due to the implementation of the sensor fusion algorithm on the WSN and the encoder, a series of base tests must be performed where the localization of the mobile robot with each sensor operating separately. Once the performance of the AGV with only a single sensor in use has been performed, the localization of the AGV with the incorporated sensor fusion was performed. These tests were executed through simulations and prototype developments.

1. *Simulation Test*

The simulated tests involved the use of a program that was developed to model the operation of the AGV using dynamics. The program was developed using PYTHON 3.6 to generate the baseline performance of the mobile robot with the individual use of the sensors without the presence of the sensor fusion algorithm. After the baseline tests, the program feeds information from each sensor into the particle filter algorithm to sensor fuse the two and perform a more accurate localization of the AGV. The initial conditions for the simulation tests are summarized in Table I.

1. simulation parameters

| Simulation Properties | Equivalent Parameters |
| --- | --- |
|
| Vehicle wheel radius, mm | 24 |
| Vehicle mass, kg | 2 |
| Friction coefficient between vehicle and road | 0.8 |
| Vehicle motor speed, rad/s | 30 |
| Random Noise for WSN sensor | Normal distribution |
| Mean of 0 |
| Std. Dev. of 0.05m |
| Random Noise for Encoder | Normal distribution |
| Mean of 0.01 |
| Std. Dev. of 0.05m |

1. *Prototype Test*

The prototype would be assembled using the components indicated in the AGV Design Section of the paper with the body of the robot built using the NXT Lego Mindstorm set. Once the automated vehicle was complete, the prototype would perform localization with a single sensor first to form the baseline, and then the sensor fusion run would be performed to determine the actual localization capacity of the sensor fusion algorithm.

However, to accomplish this test, special consideration for the operation of the WSN system was taken into consideration. The transmitters were placed in the corner of the rooms so that the sensors would only read one unique position for the vehicle in the room. One transmitter was considered as the point of origin while the other was considered as the boundary limit to one of the axes passing through the point of origin. The data gathered from all the tests were compiled and analyzed in the following section of the paper.

1. RESULTS & DISCUSSIONS

The simulated test runs were performed using the aforementioned software and the particle filter algorithm integrated into the Raspberry Pi B+ board. The results are shown in the figures that follow.

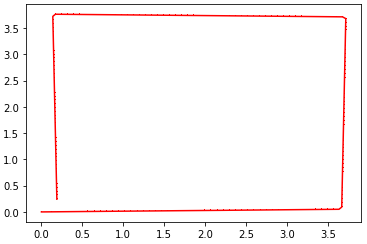


Fig. 12. Actual path taken by simulated AGV

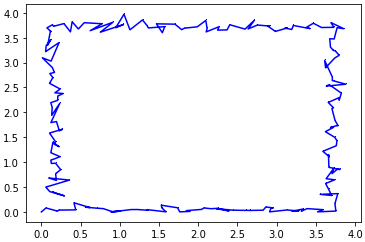


Fig. 13. Simulated AGV Localization using WSN only

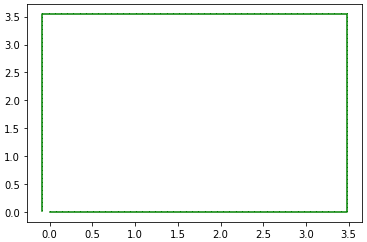


Fig. 14. Localization using Encoder Only

From Figure 12, the actual path taken by the simulated AGV was calculated using dynamics given the parameters of the vehicle. The individual sensor results shown in Figure 13 and 14 show how the sensors tend to deviate away from the actual path that the AGV needs to take. Hence, the reason for executing the particle filter algorithm.

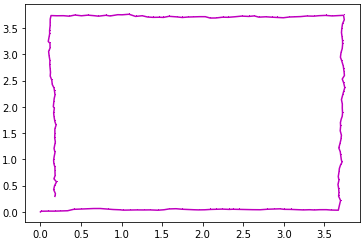
Fig. 15. Localization through Particle Filter Algorithm

Figure 15 shows the performance of the particle filter algorithm in fusing the two sensors. Notably, the localization of the sensor fusion combines the best qualities of the sensors used in the individual localization tests. It was able to obtain the accuracy of the WSN sensor while exhibiting the precision of the encoder during localization. This is a reflection and verification of the nature of sensor fusion to acquire the best qualities of the sensors and use them in conjunction to improve the performance, which was localization of the AGV for this case. The improvement of the AGV localization can be clearly seen in Figure 16, where the sensor fusion localization basically overlapped with the actual path taken by the simulated AGV.

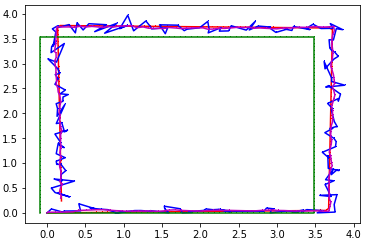


Fig. 16. Overlapping Localization of simulated test runs

1. CONCLUSION

In sum, the paper has demonstrated a particle filter approach for sensor fusion of IEEE 802.14 and encoder. To verify the performance of the fusion algorithm, numerical simulation was made by derivation of mathematical model of omni-wheel AGV. A mathematical model was derived using Newton's 2nd law.

The use of the particle filter algorithm to accomplish sensor fusion was a considerable success, with the localization of the sensor fusion nearly overlapping the actual path of the robot. Nonetheless, the researchers recommend further research to be done to investigate possible optimizations regarding the constants and weights of the particle filter to further improve the estimation performance. Additionally, further research can be done to investigate the effect of the number of particles being simulated to further optimize the algorithm. Further studies involving the localization of AGVs with sensor fusion algorithms should consider testing more complicated trajectories to test the resilience of the algorithm.

REFERENCES

|  |  |
| --- | --- |
| [1] | K. Ignatiev, M. Kopichev and A. Putov, "Autonomous Omni-Wheeled Mobile Robots," in *2nd International Conference on Industrial Engineering, Apllications and Manufacturing*, Saint Petersburg, Russia, 2016. |
| [2] | I. Lee, J. Kim, J. Lee, J. Kim and S. Kim, "Kalman Filter-based Sensor Fusion for Improving Localization of AGV," *Advanced Materials Research,* pp. 1818-1822, 2012. |
| [3] | C. Rohrig, D. HeB, C. Kirsch and F. Klinemund, "Localization of an Omnidirectional Transport Robot Using IEEE 802.15.4a Ranging and Laser Range Finder," in *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010. |
| [4] | K. Krinkin, E. Stotskaya and Y. Stotskiy, "Design and Implementation Raspberry Pi-based Omni-wheel Mobile Robot," in *AINL-ISMW FRUCT Conference*, St. Petersburg, Russia, 2016. |
| [5] | Y. Liu, H. Li, L. Ding, L. Liu, T. Liu, J. Wang and H. Gao, "An Omnidirectional Mobile Operating Robot Based on Mecanum Wheel," in *2nd International Conference on Advanced Robotics and Mechatronics*, China, 2017. |
| [6] | X. Gao, K. Li, Y. Wang, G. Men and D. Zhou, "A floor cleaning robot using Swedish Wheels," in *IEEE International Conference on Robotics and Biomimetics*, Sanya, China, 2007. |
| [7] | I. Calciu, L. Vasilescu, A. Draghici, D. Rosner and M. Patrascu, "Three Omni-wheel ballbot optimum implementation," in *RoEduNet International Conference - Networking in Education and Research*, Craiova, Russia, 2015. |
| [8] | C. Rohrig, A. Heller, D. Heb and F. Kunemund, "Global Localization and Position Tracking of Automatic Guided Vehicles using passive RFID Technology," in *Conference ISR Robotik*, Berlin, Germany, 2014. |
| [9] | S. Yoon, S. Park and J. Kim, "Kalman Filter Sensor Fusion for Mecanum Wheeled Automated Guided Vehicle Localization," *Journal of Sensors,* pp. 1-7, 2015. |
| [10] | N. Yang, W. Tian, Z. Jin and C. Zhang, "Particle filter for sensor fusion in a land vehicle navigation system," *Meas. Sci. Technol,* pp. 677-681, 2005. |
| [11] | F. Caron, M. Davy, E. Duflos and P. Vanheeghe, "Particle Filtering for Multisensor Data Fustion with Switching: Observation Models: Application to Land Vehicle Positioning," *IEEE Transactions on Signal Processing,* pp. 2703-2719, 2007. |
| [12] | C. Ankita, "Wireless Sensor Networks," electroSome, 6 April 2018. [Online]. Available: https://electrosome.com/wireless-sensor-networks/. [Accessed 18 April 2018]. |
| [13] | B. Ristic, S. Arulampalam and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, London, United Kingdom: Artech House, 2004. |
| [14] | J. Sillero, "Sensor fusion methods for indoor navigation using UWB radio aided INS/DR," KTH Electrical Engineering, Stockholm, Sweden, 2012. |
| [15] | RobotShop Inc., "48mm Omniwheel for LEGO NXT, Servo," RobotShop Inc., 2018. [Online]. Available: https://www.robotshop.com/en/48mm-omniwheel-lego-nxt-servo.html. [Accessed 18 April 2018]. |
| [16] | Amazon, "LEGO MINDSTORMS Interactive Motor (9842)," Amazon, 2018. [Online]. Available: https://www.amazon.com/LEGO-4297095-MINDSTORMS-Interactive-Motor/dp/B000PM8I1G. [Accessed 18 April 2018]. |
| [17] | philohome.com, "NXT motor internals," philohome.com, [Online]. Available: http://www.philohome.com/nxtmotor/nxtmotor.htm. [Accessed 18 April 2018]. |
| [18] | Sparkfun Electronics, "SparkFun XBee Shield," Sparkfun Electronics, [Online]. Available: https://www.sparkfun.com/products/12847. [Accessed 18 April 2018]. |
| [19] | Raspberry Pi Foundation, "Raspberry Pi 1 Model B+," Raspberry Pi Foundation, July 2914. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-1-model-b-plus/. [Accessed 18 April 2018]. |