INDOOR LOCALIZATION OF AN OMNI-DIRECTIONAL WHEELED MOBILE ROBOT

Sasha Ginzburg and Scott Nokleby
Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Ontario, Canada
Email: sasha.ginzburg@uoit.ca; scott.nokleby@uoit.ca

Received March 2012, Accepted August 2013
No. 12-CSME-35, E.I.C. Accession 3355

ABSTRACT

This paper presents a localization system developed for estimating the pose, i.e., position and orientation, of an omni-directional wheeled mobile robot operating in indoor structured environments. The developed system uses a combination of relative and absolute localization methods for pose estimation. Odometry serves as the relative localization method providing pose estimates through the integration of measurements obtained from shaft encoders on the robot's drive motors. Absolute localization is achieved with a novel GPS-like system that performs localization of active beacons mounted on the mobile robot based on distance measurements to receivers fixed at known positions in the robot's indoor workspace. A simple data fusion algorithm is used in the localization system to combine the pose estimates from the two localization methods and achieve improved performance. Experimental results demonstrating the performance of the developed system at localizing the omni-directional robot in an indoor environment are presented.

Keywords: omni-directional; localization; mobile robot; data fusion.

LOCALISATION EN ENVIRONNEMENT INTÉRIEUR D'UN ROBOT MOBILE OMNIDIRECTIONNEL SUR ROUES

RÉSUMÉ

Cet article présente un système de localisation développé pour une estimation de la position d'un robot mobile omnidirectionnel sur roues, opérant dans un environnement structuré. Le système développé utilise une combinaison de méthode de localisation. Comme méthode de localisation relative, on se sert de l'odométrie. La localisation absolue se fait par un système innovateur, semblable au GPS, qui effectue la localisation de balises actives montées sur le robot mobile, en se basant sur la mesure de la distance avec les receveurs situés à des positions connues, fixés dans l'espace de travail du robot. Un simple algorithme de fusion des données est utilisé pour combiner l'estimation des positions par les deux méthodes de localisation et arriver ainsi à une amélioration du rendement. Des résultats expérimentaux démontrent la performance des systèmes présentés.

Mots-clés : omnidirectionnel ; robot mobile ; fusion des données.

1. INTRODUCTION

Localization is a fundamental problem for autonomous mobile robot navigation. An autonomous mobile robot needs to know its position and orientation relative to its environment at every instant, in order to move from one location to another. To provide the mobile robot with the location information that it needs for navigation, a localization system is needed. Such a system must be able to accurately track the motion of the mobile robot and provide pose updates to the robot's navigation system at a sufficiently high rate to allow for proper control of its motion.

Since localization is such a critical problem and a prerequisite for autonomous navigation, it is a topic that has attracted much interest in the mobile robotics research community. As such, numerous methods have so far been developed for localizing mobile robots both in indoor and outdoor environments. While the number of localization methods is quite extensive, all of these methods can be categorized as either relative or absolute localization methods [1]. Relative methods, also referred to as dead-reckoning, rely on sensors mounted on the mobile robot that only obtain measurements of the robot's motion. The pose of the robot is updated from a known starting location by integrating the sensor measurements over time as the robot moves. Since dead-reckoning methods use high frequency sensors, such as shaft encoders, they are able to update the robot's pose at a high rate; however, the process of sensor measurement integration inevitably leads to error accumulation in the pose estimates. In contrast, absolute localization methods use sensors, such as sonar and laser scanners, that obtain measurements from the surrounding environment. As a result, absolute methods are inherently more accurate than relative localization approaches and most importantly the error in the resulting pose estimates is bounded. However, in comparison to relative techniques, absolute methods update the robot's pose at a much slower rate.

Most localization systems developed for mobile robots do not rely on a single method for pose estimation, but instead use a combination of two or more methods with multiple sensors mounted on the robot. Cox [2] describes an autonomous robot vehicle that uses a combination of odometry and optical range sensing to sense its environment. The vehicle is provided with an a-priori map of its environment and uses an algorithm for matching the sensor data to the map. An additional algorithm is used to integrate the odometric and matched positions to obtain improved estimates of the vehicle's location. Leonard and Durrant-Whyte [3] developed a system that uses an Extended Kalman Filter (EKF) to match beacon observations to a navigation map to estimate the location of a mobile robot. The beacons were considered as naturally occurring environment features that could be reliably observed in sensor measurements. Chenavier and Crowley [4] describe a localization method that uses the EKF to combine the position estimates from odometry with observations of known vertical edges (e.g., doors, pipes, corners, etc.) from a camera onboard the mobile robot. To eliminate the task of a human having to select the landmarks, Thrun [5] developed an algorithm called BaLL that enables a mobile robot to learn a set of landmarks by itself and to learn how to recognize them using artificial neural networks. Seki et al. [6] developed a system for positioning a powered wheelchair using active ultrasonic beacons at known positions on a ceiling and two receivers on the wheelchair that measure the time-of-flight of ultrasonic pulses from the beacons. Another localization system that uses active ultrasonic beacons was developed by Kleeman [7]. In [7], an iterated EKF is applied to the beacon distance measurements and dead-reckoning data to estimate optimal values of a robot's position and heading. In addition to the localization systems mentioned here, there are many others in the literature. The reader is directed to [1], which provides a review of the various mobile robot localization systems.

The focus of this paper is on the development of a localization system for the Omnibot omni-directional wheeled mobile robot (OWMR) [8], shown in Fig. 1. This localization system is developed to serve as part of the Omnibot's autonomous navigation system. The navigation system is composed of four integrated subsystems: path-following, obstacle detection, velocity control, and localization. In the context of the overall navigation system, the purpose of the localization subsystem is to estimate the Omnibot's pose and

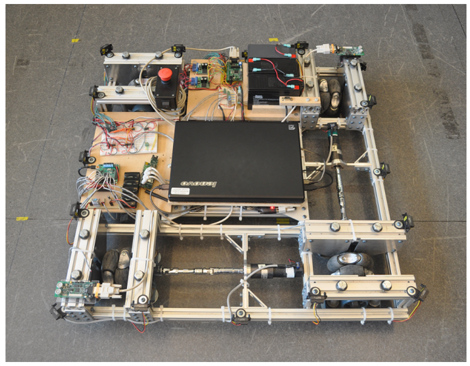


Fig. 1. Omnibot omni-directional four-wheeled mobile robot.

provide the computed pose estimates as feedback to the path-following controller to allow it to drive the Omnibot along predefined paths within an indoor structured office or factory-like environment. The pose of the Omnibot is defined by the set of variables $[x \quad y \quad \theta]^T$, where x and y are the position coordinates of the Omnibot's geometric centre and θ is its orientation angle, relative to a defined Cartesian global coordinate system.

This paper is organized as follows. In Section 2 the overall structure of the developed localization system is presented. Sections 3 and 4 describe the absolute and relative localization methods used in the localization system, respectively. The algorithm used to fuse the pose estimates from the two localization methods is described in Section 5. Experimental results are discussed in Section 6, and conclusions are provided in Section 7.

2. LOCALIZATION SUBSYSTEM STRUCTURE

The localization subsystem developed for localizing the Omnibot OWMR in an indoor environment is composed of several components, as shown in Fig. 2. In this subsystem, two different methods are used for estimating the Omnibot's pose; an absolute and a relative localization method. Absolute localization is performed by the modified Cricket system [9], while the relative localization is performed by the odometry system. Both the modified Cricket and odometry systems compute pose estimates, $[x \quad y \quad \theta]^T$, of the Omnibot independently of each other.

A data fusion algorithm is implemented in the localization subsystem, as shown in Fig. 2, to combine the pose estimates from the modified Cricket and odometry systems. The function of this data fusion algorithm is to use the absolute pose estimates from the modified Cricket system to periodically correct the error accumulated in the relative pose estimates of the odometry system. Through the application of the data fusion algorithm, the error in the odometry pose estimates is bounded; i.e., the error in the pose estimates is not allowed to grow boundlessly, which would occur if absolute pose updates are unavailable.

By combining an absolute localization method, the modified Cricket system, with a relative localization method, odometry, the performance of the localization subsystem is improved. The reason for this improvement in localization performance, is due to the complementary nature of the two systems. Since Cricket is an absolute method, it can produce very accurate pose estimates that are completely independent of each other; however, its main weakness is its relatively poor update rate. Odometry, on the other hand, has a much higher update rate in comparison with the Cricket system, but since it is a relative localization method its main weakness is an accumulation of error over time. The latter issue means that odometry can only provide good pose estimates over short-distances. Therefore, by periodically fusing the pose estimates from Cricket and odometry, it is possible to maintain the accuracy of the Cricket system, while at the same time

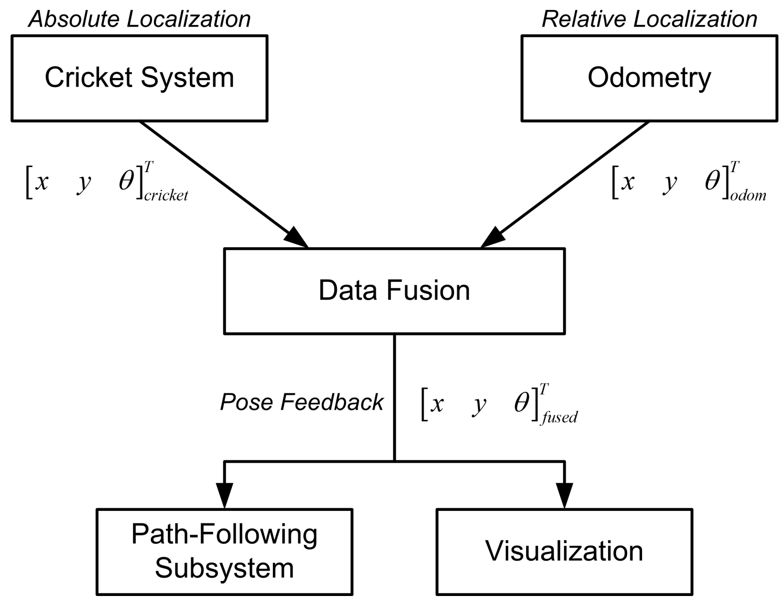


Fig. 2. Localization subsystem structure.

preserving the high update rate of odometry. The result is then a localization system that is both accurate and fast, and at the same time more reliable due to the redundancy of using two localization methods.

3. MODIFIED CRICKET SYSTEM

The absolute localization component of the Omnibot's localization subsystem is based on the Cricket indoor localization system developed in the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology [10–12]. The original Cricket system is an absolute indoor localization system that performs localization of mobile receivers based on distance measurements to active transmitters placed at known points in the surrounding environment. Cricket is similar to the widely used Global Positioning System (GPS), which localizes mobile GPS receivers based on distance measurements to orbiting satellites. However, whereas GPS relies solely on the use of radio frequency (RF) signals transmitted from satellites for range measurements, the Cricket system uses a combination of RF and ultrasonic (US) signals to perform the same function.

To make the original Cricket system more suitable for the task of mobile robot localization, modifications were made to the system. Although the original system has good indoor localization accuracy – on the order of a few centimeters – it lacks a sufficiently high update rate to allow for autonomous navigation of a mobile robot in an indoor environment; Cricket has an update rate of approximately 1 Hz. To correct this issue, the original system was modified with the goal of increasing its update rate, while maintaining the same level of accuracy. To achieve this goal, both the architecture and operation of the original system were modified.

3.1. Architecture

The hardware used in the Cricket system consists of Cricket nodes (see Fig. 3), small sensor units equipped with a microcontroller, an ultrasonic transmitter and receiver pair, an RF transceiver, and an RS-232 serial interface with a DB-9 connector. These Cricket nodes serve as the hardware for both the transmitters and receivers, respectively called beacons and listeners, in the Cricket system.

The modified system uses an active mobile architecture, illustrated in Fig. 4, in which listeners are deployed in the infrastructure as external references and active beacons are attached to the mobile robot that

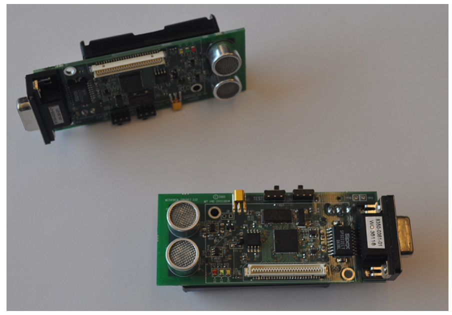


Fig. 3. Cricket node sensor units; can operate either as beacons (transmitters) or listeners (receivers).

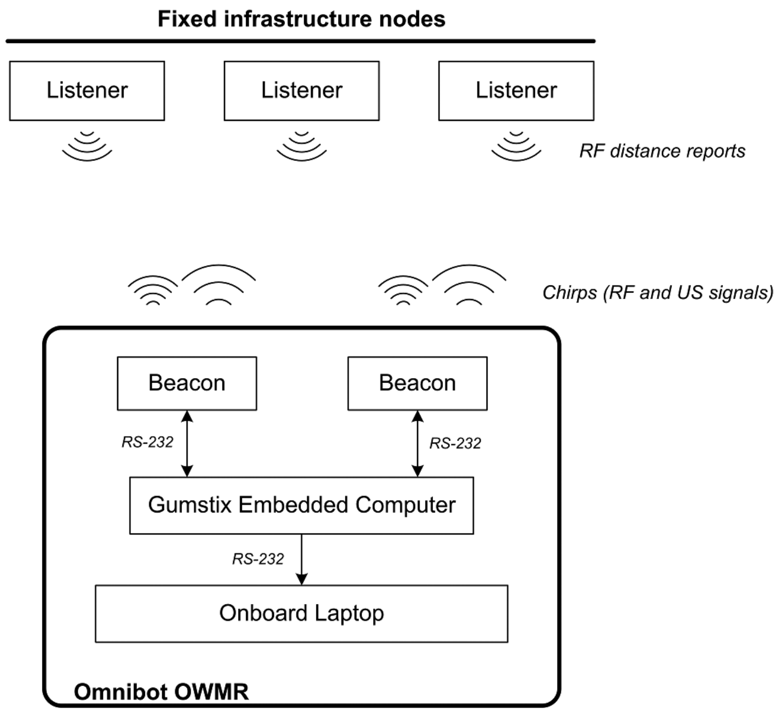


Fig. 4. Modified Cricket system architecture (active mobile).

needs to be located. The listeners in this configuration are fixed at known positions to a ceiling in the mobile robot's workspace, forming the Cartesian global coordinate system with respect to which the mobile beacons are positioned. The two actively transmitting beacons are mounted at the corners of the Omnibot mobile robot and move in a plane parallel and below the one formed by the ceiling mounted listeners. Both of the onboard beacons are interfaced to a Gumstix embedded computer using serial cables that connect their RS-232 ports to the corresponding ports on the embedded computer.

The Gumstix embedded computer is the central component of the Cricket system; its functions are to communicate with the onboard beacons, coordinate their transmissions, and compute estimates of their positions and those of the Omnibot by executing a position estimation algorithm. In addition to the beacons, the Gumstix computer is also interfaced to the laptop mounted on the Omnibot. This connection provides a link between the Cricket system and the other systems onboard the Omnibot. It is used for sending the computed pose estimates of the Omnibot from the Gumstix computer to the laptop for use in data fusion with odometry.

3.2. Operation

In the modified Cricket system, the Gumstix embedded computer coordinates the active beacon transmissions by successively triggering each of the two onboard beacons to transmit wireless signals (i.e., chirp). The Gumstix computer triggers a particular beacon to chirp by sending it a command over the serial interface. When a beacon is triggered, it simultaneously broadcasts an RF message and a US pulse for the listeners deployed on the ceiling to detect. Only those listeners that are within the ultrasonic range and have line-of-sight to the beacon will hear the broadcast.

Nearby listeners will first receive the RF signal – almost instantaneously since it travels at the speed of light – then a short time later detect the corresponding US pulse, propagating at the speed of sound. To compute the distance to the triggered beacon, the listeners measure the time interval between the start of the RF message and the arrival of the US pulse. This time interval is effectively the time-of-flight (TOF) of the US pulse and when combined with its known propagation speed – adjusted for the ambient temperature and pressure – allows the listeners to measure their distances to the beacon.

After measuring its distance to the beacon, each listener replies on the RF channel by transmitting an RF message containing its distance measurement and unique ID number. The triggered beacon onboard the Omnibot receives the listener RF distance reports and relays the data within the messages to the connected Gumstix computer via its RS-232 serial port. The Gumstix computer, acquires the set of simultaneous listener distance reports and processes the data for use in the position estimation algorithm. If distance measurements to three or more listeners with a-priori known positions (stored in the controller's memory) are obtained, the trilateration positioning algorithm is executed to calculate an estimate of the position coordinates $[x_B \quad y_B \quad z_B]^T$ of the triggered beacon relative to the global coordinate system.

Using the position estimates of the current and previous triggered beacons and their known placement on the Omnibot, the Gumstix also computes the position coordinates, x and y, of the Omnibot's geometric centre and its orientation angle, θ, relative to the global reference frame. Following these calculations, the Gumstix sends the Omnibot's pose estimate to the connected onboard laptop using serial communication and triggers the next beacon to chirp.

3.3. Position Estimation Algorithm

In the modified Cricket system, position estimates of the mobile beacons are computed based on the distance measurements to ceiling mounted listeners at a-priori known positions using the method of trilateration. To calculate a mobile beacon's position $[x_B \quad y_B \quad z_B]^T$ using trilateration requires a minimum of three simultaneous distance measurements from the mobile beacon to stationary listeners, as illustrated in Fig. 5.

For a given beacon transmission, if at least three valid distance measurements are obtained, then the trilateration positioning method involves selecting three of the listeners that replied, and finding the intersection of the surfaces of three spheres centred at the listeners with radii given by the distance measurements from these listeners to the mobile beacon. The position coordinates of the beacon are then obtained as the coordinates of the intersection point of the three spheres that is below the plane of the listeners (see Fig. 5).

Following this method, the position coordinates of a mobile beacon $[x_B \quad y_B \quad z_B]^T$ are computed by solving the following three simultaneous equations for $i = 1$, 2, and 3:

$$r_i^2 = (x_B - X_i)^2 + (y_B - Y_i)^2 + (z_B - Z_i)^2 \tag{1}$$

where r_i is the distance measurement to listener i, and X_i, Y_i, and Z_i are the position coordinates of listener i, for $i = 1$, 2, and 3, respectively.

The pose of the Omnibot OWMR, $[x \quad y \quad \theta]^T$, can be obtained from the computed position coordinates of the two beacons, B_1 and B_2, mounted on the robot. The two onboard beacons are mounted at diagonally opposite corners of the Omnibot's square-shaped frame, such that they are collinear with the Omnibot's

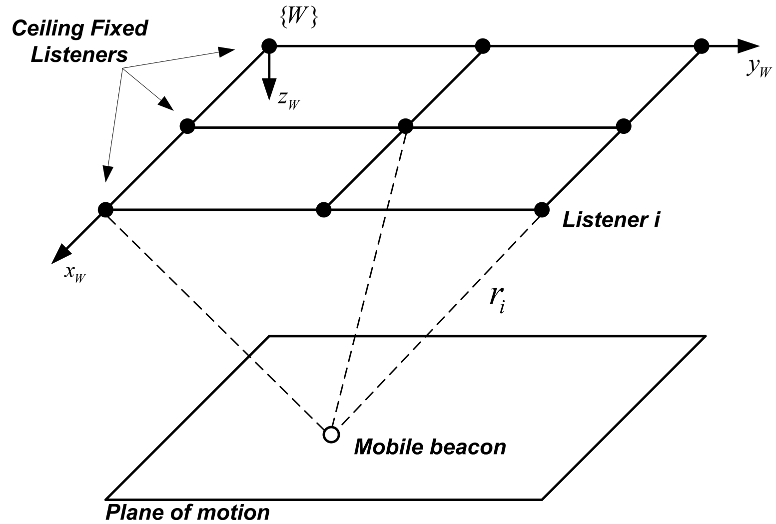


Fig. 5. Trilateration of the mobile beacon position.

geometric centre and equidistant from it. As such, the position of the Omnibot's geometric centre is the midpoint of the line segment joining the positions of the two onboard beacons and is obtained as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{x_{B_1}+x_{B_2}}{2} \\ \frac{y_{B_1}+y_{B_2}}{2} \end{bmatrix} \tag{2}$$

where (x_{B_1}, y_{B_1}) and (x_{B_2}, y_{B_2}) are the x and y position coordinates of beacons B_1 and B_2, respectively, determined from the trilateration positioning algorithm.

The orientation of the Omnibot, θ, can be obtained from the orientation of the vector, $\overrightarrow{B_1B_2}$, extending from B_1 to B_2, the positions of the two onboard beacons as

$$\theta = \text{atan2}\left[\frac{y_{B_2}-y_{B_1}}{x_{B_2}-x_{B_1}}\right] - 45^\circ \tag{3}$$

where 45° is the fixed orientation angle of the vector $\overrightarrow{B_1B_2}$ in the robot reference frame.

4. ODOMETRY

Odometry is a localization method that does not rely on external references for position estimation, but instead obtains location information by sensing the motion of the mobile robot using shaft encoders coupled to the drive motors. In odometry, the current pose of the robot is determined by measuring the robot's displacement from the previously known pose. Measurement of the robot's displacement is accomplished using motor shaft encoders, which sense the rotation of the motor shafts that drive the robot's wheels. The readings from the shaft encoders are used to compute the distances traveled by each of the robot's wheels from the previous robot pose. From the wheel distances, both the translational and rotational displacement of the robot can be computed using the odometry calculations.

Figure 6 depicts the Omnibot's omni-directional drive system which consists of four independently driven omni-wheels. To perform the odometry calculations for the Omnibot OWMR, its drive system is modeled as two orthogonal differential drive systems. The wheel pair (W_2, W_4) comprises the differential drive parallel to the Omnibot's x-axis; while the wheel pair (W_1, W_3) comprises the differential drive parallel to the Omnibot's y-axis (see Fig. 6).

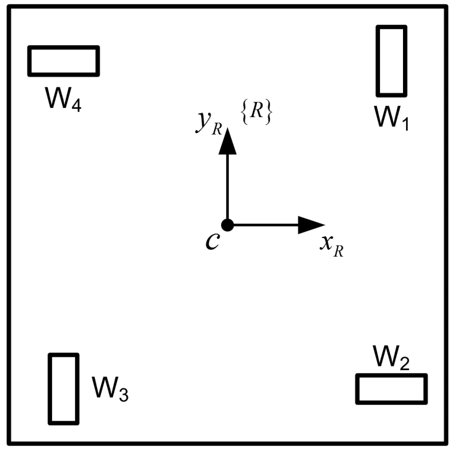


Fig. 6. Omnibot omni-directional drive system.

Since the two differential drives (W_2, W_4) and (W_1, W_3) are decoupled, the displacement of the Omnibot during a sampling period can be obtained as the sum of the displacements produced by the two differential drives. Thus, the odometry calculations for normal differential drive robots, such as those in [13] and [14], can be applied to determine the displacements resulting from the Omnibot's two differential drives.

The first step in performing odometry for the Omnibot is to compute the distances traveled by the four omni-wheels during the sampling period, $T = 50$ ms, as

$$\Delta d_i = \frac{n_i}{N} \cdot (2\pi R_{wheel}) \tag{4}$$

where Δd_i is the distance traveled by wheel i (for i= 1 to 4); the encoder resolution N is 184,320 pulses per revolution of the omni-wheel; and the omni-wheel radius R_{wheel} is 0.06 m.

The translational and rotational displacements resulting from differential drive (W_2, W_4) are obtained as

$$\Delta d_{(W_2,W_4)} = \frac{\Delta d_2 + \Delta d_4}{2} \tag{5}$$

$$\Delta \theta_{(W_2,W_4)} = \frac{\Delta d_2 - \Delta d_4}{W} \tag{6}$$

where the wheelbase length W is 0.597 m. The wheelbase is the distance between the pair of parallel wheels measured along their axis of rotation.

Similarly, the translational and rotational displacements resulting from differential drive (W_1, W_3) are obtained as

$$\Delta d_{(W_1,W_3)} = \frac{\Delta d_1 + \Delta d_3}{2} \tag{7}$$

$$\Delta \theta_{(W_1,W_3)} = \frac{\Delta d_1 - \Delta d_3}{W} \tag{8}$$

The rotational displacement of the Omnibot, $\Delta\theta$, during the sampling period is the average of the rotational displacements produced by the two differential drives:

$$\Delta \theta = \frac{\Delta \theta_{(W_2,W_4)} + \Delta \theta_{(W_1,W_3)}}{2} \tag{9}$$

The pose of the Omnibot at time instant $n+1$ (i.e., the current pose) relative to the global reference frame is computed as

$$x_{n+1} = x_n + \Delta d_{(W_2,W_4)} \cos\left(\theta_n + \frac{\Delta\theta}{2}\right) + \Delta d_{(W_1,W_3)} \cos\left(\theta_n + \frac{\pi}{2} + \frac{\Delta\theta}{2}\right) \tag{10}$$

$$y_{n+1} = y_n + \Delta d_{(W_2,W_4)} \sin\left(\theta_n + \frac{\Delta\theta}{2}\right) + \Delta d_{(W_1,W_3)} \sin\left(\theta_n + \frac{\pi}{2} + \frac{\Delta\theta}{2}\right) \tag{11}$$

$$\theta_{n+1} = \theta_n + \Delta\theta \tag{12}$$

5. DATA FUSION ALGORITHM

The pose estimates produced by the modified Cricket and odometry systems implemented on the Omnibot are fused together to achieve improved localization performance. Fusion of the pose estimates is performed by a simple data fusion algorithm. The purpose of this data fusion algorithm is to use the absolute pose estimates from the Cricket system to periodically correct the error accumulated in the odometry estimates in the time period between Cricket pose updates. The periodic updates from the Cricket system prevent the error in the odometry estimates from growing over time as the Omnibot navigates. In the time between Cricket pose updates, the localization system relies on the good short-term accuracy of the odometry system for pose estimation. The result of this fusion process is that the pose estimates from the localization system maintain the accuracy of the Cricket system and the update rate of odometry. In addition, since both Cricket and odometry provide redundant information, the fusion of the two systems improves the reliability of the localization system.

The first step of the fusion process is the initialization of odometry by the modified Cricket system. In order to perform the odometry calculations, the initial pose of the Omnibot must be known relative to the global coordinate system. This starting pose estimate is provided for the odometry algorithm by averaging the first 12 pose estimates computed by the Cricket system while the Omnibot is stationary; the number of estimates that are averaged is an adjustable parameter. With the initial pose of the Omnibot known, the odometry algorithm proceeds to estimate the Omnibot's pose at a rate of 20 Hz.

During the execution of the odometry algorithm, pose updates are periodically obtained from the modified Cricket system at a rate of $\sim$3.5 Hz. Whenever a new pose update is received from the Cricket system, the data fusion algorithm is executed to fuse the Cricket and odometry pose estimates.

In the fusion process, it is important to understand that the current Cricket pose estimate at time step n does not correspond to the newest odometry pose estimate, but instead to the odometry estimate calculated at the previous pose update at time step $n-1$. This is because the current Cricket pose estimate at time step n, corresponds to the Omnibot's location at the time when the last beacon chirped, which occurred right after the previous Cricket pose update was received at time step $n-1$. Basically, the Cricket system lags behind the odometry system, since it has a much slower pose update rate compared to odometry.

The current Cricket pose estimate at time step n, $\mathbf{P}_{cricket,n} = [x \quad y \quad \theta]^T_{cricket,n}$, is fused with the odometry pose estimate at time step $n-1$ (the previous Cricket update), $\mathbf{P}_{odom,n-1} = [x \quad y \quad \theta]^T_{odom,n-1}$, using a weighted average of the pose estimates as

$$\mathbf{P}_{fused,n} = w\mathbf{P}_{odom,n-1} + (1-w)\mathbf{P}_{cricket,n} \tag{13}$$

where the constant coefficient w is the weight of the weighted average and is set to 0.5. It should be noted that more complex fusion techniques, such as Extended Kalman Filters (EKFs), could be used, but for the purposes of this work, a simple weighted fusion was chosen.

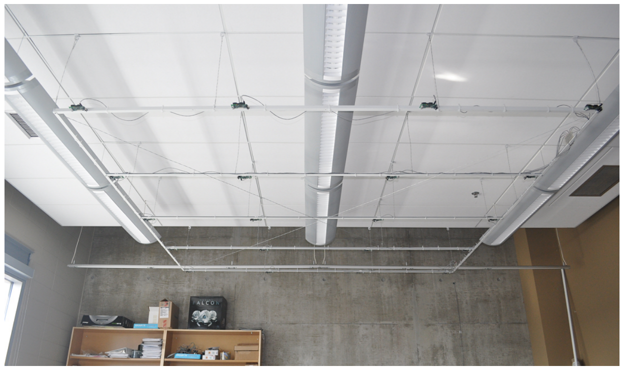


Fig. 7. Listeners attached to a ceiling suspended frame above the robot's workspace.

The displacement of the Omnibot measured by odometry during the time period from time step $n-1$ to n (i.e., the period between Cricket updates) is computed as

$$\Delta \mathbf{P}_{odom} = \mathbf{P}_{odom,n} - \mathbf{P}_{odom,n-1} \tag{14}$$

To account for the Omnibot's displacement during the time between consecutive Cricket updates, the result in Eq. (14) is added to the fused pose estimate computed in Eq. (13) as

$$\mathbf{P}_{fused,n} = \mathbf{P}_{fused,n} + \Delta \mathbf{P}_{odom} \tag{15}$$

The resulting fused pose estimate is then set as the starting pose for the next iteration of odometry as

$$\mathbf{P}_{odom,n} = \mathbf{P}_{fused,n} \tag{16}$$

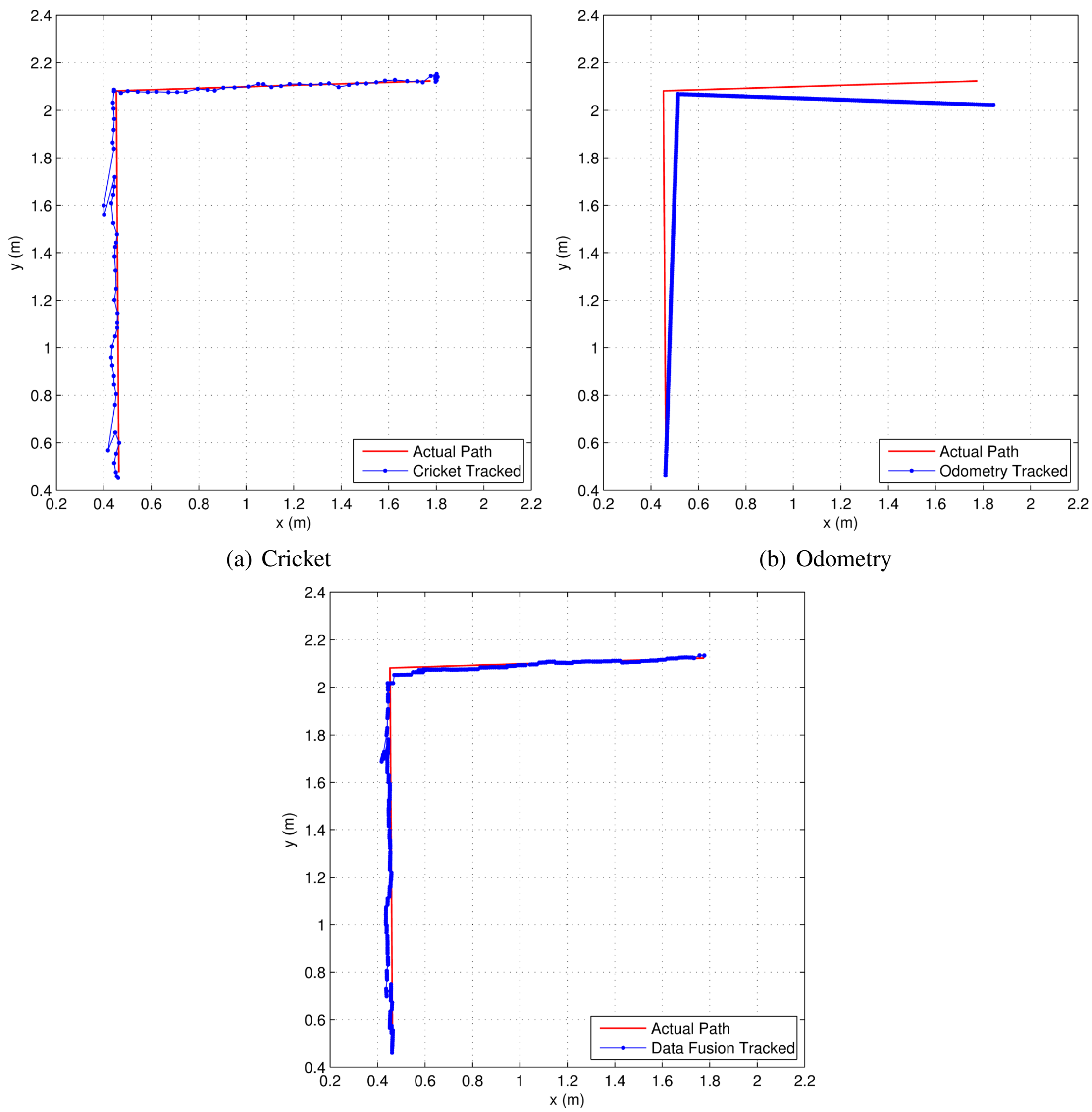
Through this process the odometry pose estimate at time step n has been corrected using the absolute pose estimate from the Cricket system. By continuing to fuse the Cricket and odometry pose estimates at a rate of $\sim$3.5 Hz, the error in the odometry estimates is not allowed to grow over time.

6. EXPERIMENTAL RESULTS

To evaluate the performance of the developed localization system at estimating the Omnibot OWMR's pose and tracking its motion, a series of tracking experiments were performed. The experiments were carried out in an indoor laboratory environment within a designated 3 m x 3 m work area, where the Omnibot moved on a smooth level floor. To provide localization coverage over this workspace, the modified Cricket system was deployed by mounting listener nodes onto a ceiling suspended frame, as shown in Fig. 7. A total of 20 listener nodes were mounted onto the frame in a grid pattern.

In the tracking experiments, the Omnibot was pre-programmed to execute a motion sequence in an open-loop manner without path-following control; only the closed-loop velocity control subsystem was used to control velocities of the Omnibot's wheels in order to execute the desired sequence of motions. A motion sequence that produced an L-shaped path was used for the tracking experiments. To evaluate the tracking performance of the localization system at different speeds of the mobile robot, a series of three experimental runs were performed corresponding to Omnibot translational speeds of 0.15, 0.3, and 0.5 m/s.

For all experimental runs, pose estimates were logged from both the overall localization system, which fuses the Cricket and odometry pose estimates, and from the modified Cricket and odometry systems individually. This means that for each experimental run a total of three data sets were logged simultaneously during run-time, corresponding to the pose estimates computed by the modified Cricket, odometry, and data



(a) Cricket

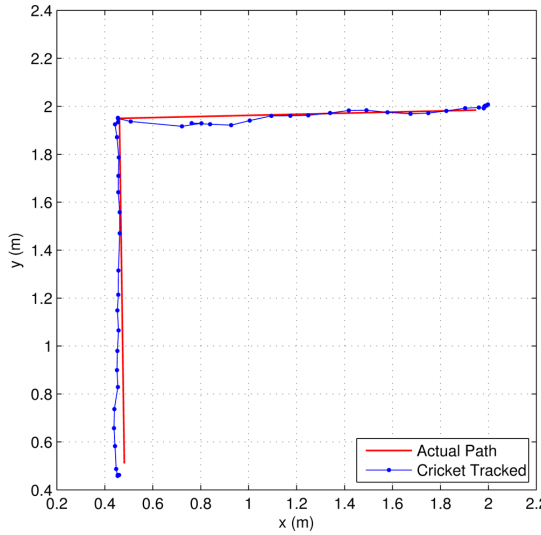
(b) Odometry

(c) Data Fusion

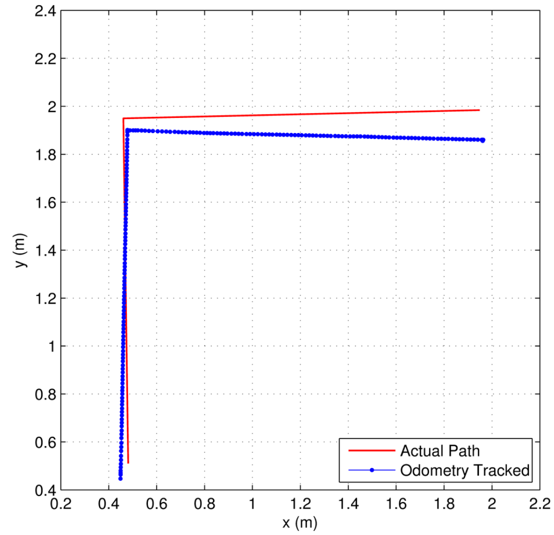
Fig. 8. Tracking results – Run 1 at 0.15 m/s.

fusion (i.e., overall) systems. The reason for logging the data from the modified Cricket and odometry systems in addition to the overall system, was to verify the effectiveness of the data fusion algorithm. It was important to determine whether the fusion of the Cricket and odometry pose estimates in real-time, resulted in improved localization performance as was expected prior to conducting the experiments.

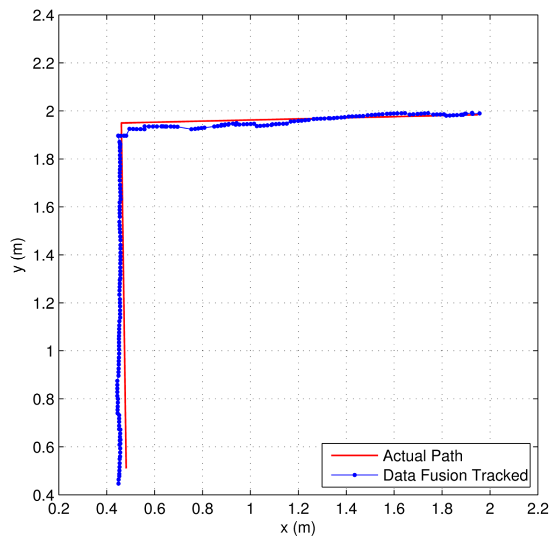
Evaluation of the tracking performance in these experiments was based on comparing the path tracked by the localization system to the actual path traveled by the Omnibot during each run. A marker attached to the front side of the Omnibot was used to draw a line on the floor as the Omnibot moved to indicate the traveled path for each experimental run. After each run, points were sampled from the line drawn on the floor by manually measuring their x and y coordinates in the defined global coordinate system. With these points, the Omnibot's actual path for each run was constructed and compared to the path tracked by the localization system.



(a) Cricket



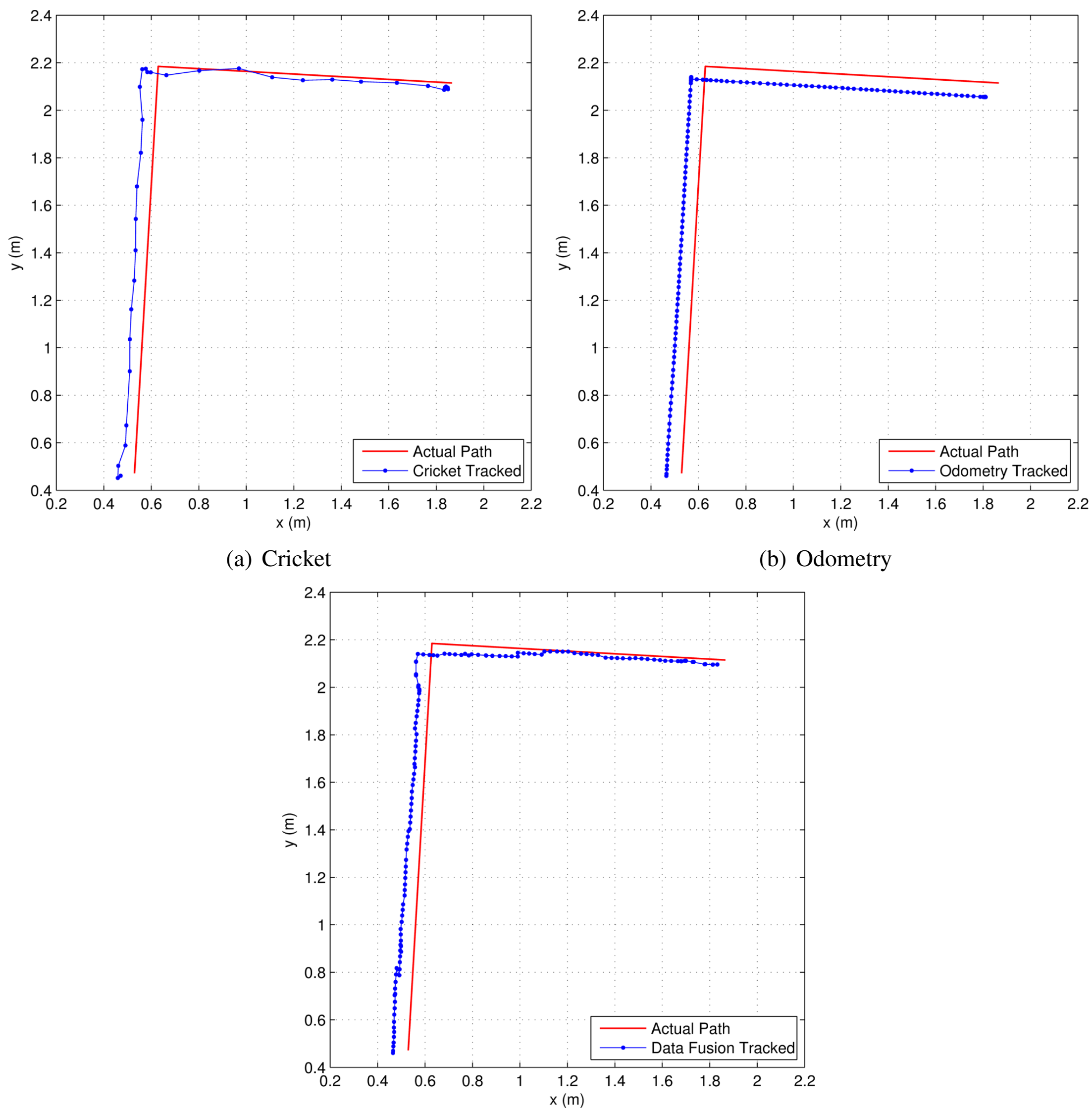
(b) Odometry



(c) Data Fusion

Fig. 9. Tracking results – Run 2 at 0.3 m/s.

The position tracking results for the three experimental runs at Omnibot speeds of 0.15, 0.3, and 0.5 m/s are shown in Figs. 8, 9, and 10, respectively. For each of these experimental runs, the position tracking error was measured as the distance from each position estimate to the actual path (i.e., the point to line distance). The mean tracking error for the three experimental runs is provided in Table 1. From these results, it is evident that the localization system is able to accurately track the Omnibot's motion in an indoor environment. It is observed from the data fusion graphs in Figs. 8(c), 9(c), and 10(c), that the periodic pose updates from the modified Cricket system were successful in correcting the error in the odometry pose estimates. In addition, the application of the data fusion algorithm reduces the effects of outliers in the Cricket data. This is especially apparent in the results in Figure 8. It should be noted that the improvement in the pose estimates decreases as the speed of the platform is increased. This limitation could be overcome through the use of a more advanced data fusion algorithm, such as the use of an EKF. Overall, the results



(a) Cricket (b) Odometry

(c) Data Fusion

Fig. 10. Tracking results – Run 3 at 0.5 m/s.

prove that the approach of fusing an absolute with a relative positioning method results in a localization system with improved performance.

7. CONCLUSIONS

This paper presented a localization system for estimating the pose of an omni-directional mobile robot operating in indoor structured environments. This system was developed for use in the mobile robot's navigation system, to allow for autonomous navigation of the robot. The approach used in developing this system was to combine absolute and relative localization methods and fusing their pose estimates to achieve improved localization performance. Absolute localization was achieved using the GPS-like modified Cricket system, while relative localization was performed using odometry. Fusion of the relative and absolute pose estimates was executed using a data fusion algorithm that uses a weighted average to combine the estimates.

Table 1. Mean tracking error (in cm) for the L-shaped motion sequence experiment.

	Cricket	Odometry	Data Fusion
Run 1 (0.15 m/s)	1.11	5.0	1.11
Run 2 (0.3 m/s)	1.64	6.53	1.46
Run 3 (0.5 m/s)	3.56	5.98	3.75

The functionality and effectiveness of the proposed localization system was verified through a series of tracking experiments for different mobile robot velocities. The experimental results demonstrated the good performance of the developed localization system at tracking the mobile robot in an indoor environment. It is expected that the use of a more complex data fusion algorithm will yield even better results. This will be investigated in future work.

REFERENCES

1. Borenstein, J., Everett, H.R., Feng, L. and Wehe, D., "Mobile robot positioning: Sensors and techniques", *Journal of Robotic Systems*, Vol. 14, No. 4, pp. 231–249, 1997.
2. Cox, I.J., "Blanche – An experiment in guidance and navigation of an autonomous robot vehicle", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 193–204, April 1991.
3. Leonard, J.J. and Durrant-Whyte, H.F., "Mobile robot localization by tracking geometric beacons", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 376–382, June 1991.
4. Chenavier, F. and Crowley, J.L., "Position estimation for a mobile robot using vision and odometry", in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2588–2593, Nice, France, May 1992.
5. Thrun, S., "Finding landmarks for mobile robot navigation", in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 958–963, Leuven, Belgium, May 1998.
6. Seki, H., Kobayashi, S., Kamiya, Y., Hikizu, M. and Nomura, H., "Autonomous/semi-autonomous navigation system of a wheelchair by active ultrasonic beacons", in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 1366–1371, San Francisco, CA, April 2000.
7. Kleeman, L., "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning", in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2582–2587, Nice, France, May 1992.
8. Bemis, S., Design and development of a novel omni-directional platform, Master's Thesis, University of Ontario Institute of Technology, Oshawa, Ontario, 2009.
9. Ginzburg, S., von Frankenberg, F. and Nokleby, S., "Design and implementation of an indoor localization system for the Omnibot omni-directional platform", *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 33, No. 4, pp. 715–729, 2009.
10. MIT Computer Science and Artificial Intelligence Laboratory, "The cricket indoor location system", http://cricket.csail.mit.edu/, 2006.
11. Priyantha, N., Chakraborty, A. and Balakrishnan, H., "The Cricket location-support system, in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 32–43, Boston, Massachusetts, August 2000.
12. Smith, A., Balakrishnan, H., Goraczko, M. and Priyantha, N., "Tracking moving devices with the cricket location system", in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pp. 190–202, Boston, MA, June 2004.
13. von der Hardt, H.J., Arnould, P., Wolf, D. and Dufaut, M., "A method of mobile robot localisation by fusion of odometric and magnetometric data", *The International Journal of Advanced Manufacturing Technology*, Vol. 9, No. 1, pp. 65–69, 1994.
14. Surrecio, A., Nunes, U. and Araujo, R., "Fusion of odometry with magnetic sensors using Kalman filters and augmented system models for mobile robot navigation", in *Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 1551–1556, June 2005.