IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 6, 2019, accepted January 23, 2020, date of publication February 3, 2020, date of current version February 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971169

Low-Cost Reduced Navigation System for Mobile Robot in Indoor/Outdoor Environments

EHAB I. AL KHATIB[1], MOHAMMAD ABDEL KAREEM JARADAT[2,3], AND MAMOUN F. ABDEL-HAFEZ[2], (Senior Member, IEEE)

[1]Mechanical Engineering Department, Southern Methodist University, Dallas, TX 75205, USA
[2]Department of Mechanical Engineering, American University of Sharjah, Sharjah 26666, UAE
[3]Department of Mechanical Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan

Corresponding author: Mohammad Abdel Kareem Jaradat (author@mjaradat@aus.edu)

ABSTRACT This paper presents a low-cost based approach for solving the navigation problem of wheeled mobile robots to perform required tasks within indoor and outdoor environments. The presented solution is based on probabilistic approaches for multiple sensor fusion utilizing low-cost visual/inertial sensors. For the outdoor environment, the Extended Kalman Filter (EKF) is used to estimate the robot position and orientation, the system consists of wheel encoders, a reduced inertial sensor system (RISS), and a Global Positioning System (GPS). For the indoor environment, where GPS signals are blocked, another EKF algorithm is proposed using low cost depth sensor (Microsoft Kinect stream). EKF indoor localization is based on landmarks extracted from the depth measurements. A hybrid low-cost reduced navigation system (HLRNS) for indoor and outdoor environments is proposed and validated in both simulation and experimental environments. Additionally, an input-output state feedback linearization (I-O SFL) technique is used to control the robot to track the desired trajectory in such an environment. According to the conducted validation simulation and experimental testing, the proposed HLRNS provides an acceptable performance to be deployed for real-time applications.

INDEX TERMS Extended kalman filter, Kinect depth sensor, low-cost navigation, mobile robot, sensor fusion.

I. INTRODUCTION

The aim of this paper is to provide a comprehensive solution to the navigation problem of Mobile Robots, by developing a robust technique that enables robots to face the numerous challenges arising in indoor and outdoor environments. This research focuses on analyzing localization techniques using commercially available sensors with the integration of trajectory tracking algorithm and with its implementation on the Wheeled Mobile Robot (WMR).

Mobile robot navigation covers a broad spectrum of Mechatronic systems. However, the most challenging problem is obtaining the exact knowledge of the position of the robot. In mobile robot navigation systems, there are two types of localization sensors. The first type is the onboard sensor which includes encoders and inertial measurement units (IMUs). These sensors measure the robot's linear and angular velocities and acceleration along the robot's body axes. By integrating sensor measurements, the mobile robot position and direction can be predicted, which is known as dead-reckoning [1]; the dead-reckoning is not applicable for long-term navigation because of the unavoidable accumulated error due to the integration steps. On the other hand, the second type is based on external absolute sensors; these sensors include cameras and Global Positioning System (GPS). These sensors can be used to measure the absolute position and direction of the robot and can used in the measurement equation for correction.

In [2], the authors classified two groups of sensors based on measurement techniques used: relative positioning sensors such as wheel odometry [1], visual odometry [3] and inertial sensor [4]. On the other hand, absolute positioning sensors such as Global Positioning Systems [5], [6], landmark navigation [7], [8], active beacons [9], ultrasound range sensor [10], light detection and ranging (LiDAR) [11], cameras [12] and magnetic compasses [1]. The two groups have pros and cons. The absolute sensors have accurate readings with a low update rate. On the other hand, the relative

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tan.

positioning sensors have a fast update rate but with drifting bias [13], [14].

Developers of mobile robots usually combine the two methods using a sensor fusion algorithm to provide high-accuracy estimates with high update rates. The basic problem of multi-sensor data fusion is determining the best algorithm for combining the multi-sensor data input [15]. Another important issue with navigational systems that needs to be dealt with is the unpredictability that exists in the operating environment. The unpredictability could be due to operation in harsh environments or due to corruption in the sensor's output due to noise or bias. The level of uncertainty arises if the robot lacks critical information for carrying out its tasks [16]. To address such issues, researchers often come up with algorithms that can improve the overall robustness of the system. The goal is to develop robust algorithms that enable robots to face the various challenges arising from operation in dynamic environments [17]–[19]. The main advantage of a probabilistic approach is that explicit stochastic models are used to describe the various relationships between sensors and other information sources, taking into account the uncertainties [15]. The most common algorithms for sensor fusion are Kalman Filter (KF) based approaches [20], [21], intelligent-based filters [22]–[25], and Particle Filters (PF) [26], [27].

To overcome the navigation problem, we need to answer three key questions [28]:

- Where am I?
  The robot's ability to determine its position in its reference environment, which is referred to as localization.
- Where am I going?
  The robot's ability to plan a path towards the goal location. This process usually consists of pre-planning a path that is optimized for the shortest distance by taking point obstacles and dangerous areas into consideration.
- How should I get there?
  The robot's ability to accurately follow the planned path.

Solving the localization problem is the first step to solve the navigation problem followed by path planning and trajectory tracking. The localization problem has to be solved for outdoor and indoor environments. For outdoor localization, a low-cost navigation system suitable for outdoor commercial mobile robots was described in [29]. The proposed navigational algorithm uses two GPS receivers, an IMU and wheel encoders to provide measurements for a 7-state KF that is used in estimating the state of the robot. The real-time control of the robot occurs in NI sbRIO. The NI sbRIO platform has a 40 MHz FPGA unit which allows the robot to parse all the critical low-level data such as GPS and IMU onboard to avoid latency and communication problems. When using low-cost off-the-shelf sensors for navigation, the optimality of classic KF cannot be guaranteed. Thus, methods such as [30] can be applied to the system to improve the estimation result. In [29], a similar approach was described with a 3-state discrete EKF with the measurement taken from odometry, fiber-optic gyroscope, and the angular measurements to the ground markers (obtained from the video frames taken during motion). The authors suggested that the overall filtering performance can be improved by online adjustment of the dynamics and measurement covariance matrices based on the statistical properties of the incoming data. They also proposed to extend the state of the filter to include the translational and rotational velocities.

On the other hand, for indoor localization, researchers studied the mapping and localization problems. They focused on finding a solution to the two problems of mapping and localization independently. Robotic Mapping is the problem of building an accurate map of the environment given accurate knowledge of the robot's position [31]. Also, much work has been done to estimate the robot's pose with an existing complete representation of the environment. These algorithms require a pre-defined map to define the reference frames and the structure of the world. Knowledge of the surrounding world may be displayed in many different forms; either in a full geometric map representation or in the knowledge of landmarks and their locations [31]. The author describes the application of the EKF in indoor localization [32]. Mobile robot localization addresses the problem of estimation of a mobile robot's pose given a map based only on sensor measurements and controls. Microsoft Kinect was used as a navigation sensor in many indoor localization methods [33], [34]. The work in [34] investigates the suitability of the Xbox Kinect optical sensor for navigation and simultaneous localization and mapping (SLAM). The prototype presented uses the 3D point cloud data captured by Kinect to create a 3D model of the environment. Then, it projects the 3D model to a 2D plane for 2D localization. RGBD SLAM [33] was used for 3D SLAM but found to be very slow, thus the robot speed was reduced for the algorithm to work in real-time. The presented prototype was compared against traditional solutions with a laser scanner (Hokuyo URG-04LX) in terms of SLAM performance and suitability as a navigation sensor. The results show that the use of the Kinect sensor is applicable.

In addition, a localization system that works in indoor and outdoor environments is presented in [35]–[38]. In [35], the authors developed a localization system that works for an urban site to track a robot's location, using two methods. The first one is based on odometry, a compass, and a GPS. An EKF fuses the sensor data and keeps track of the uncertainty associated with it. When the uncertainty of the first method becomes large, the algorithm switches to the second method that is based on camera pose estimation done by matching features in the image. On the other hand, the same approach was done for outdoor environments in [36]. MCL based on Kinect depth measurement was used to estimate the robot's pose when the GPS signal was blocked. In [37], a vision-based approach was presented for path following, where the video stream is taken as input to a heterogeneous landmark-based visual navigation algorithm to get the landmarks and then used to improve the robot localization accuracy. Gaussian process models are presented in [38] to

produce a measurement likelihood function for mobile robot localization in the absence of accurate sensor modeling.

Finally, from a control point of view, due to the perfect rolling constraints, wheeled mobile robot (WMR) is a typical example of a nonholonomic system. In addition to the non-linearity of the system, this leads to raising the complexity of the controller. Many studies were conducted on trajectory tracking control [39]–[42]. In [42], the authors addressed the output feedback trajectory tracking problem in the presence of parameter uncertainties and external disturbances. The quality of the proposed approach was tested in simulation environment. On the other hand, in [43] and [44], a common approach called Input-Output State Feedback Linearization (I-O SFL) is used in controlling non-linear systems to solve the trajectory tracking problem on a Pioneer differential drive robot. The kinematic model of this robot is based on the unicycle model [45]. Input-Output linearization is a well known systematic approach to the design of trajectory tracking controllers. It should be emphasized that the reference trajectory may exhibit a path with a discontinuous geometric tangent without the need for the robot to stop and correct its orientation.

In this paper, a novel hybrid low-cost reduced navigation system (HLRNS) is used to integrate between indoor and outdoor low-cost sensors based on EKF localization methods. The used low-cost sensors are commercially available, off-the-shelf sensors [30], [46]. The outdoor EKF is based on GPS, RISS, and encoders, whereas the indoor EKF is based on Microsoft Kinect depth sensor, RISS, and encoders. The proposed method was tested and validated using an autonomous vehicle in various indoor/outdoor environments. In addition, the performance of the fusion architecture is compared against a commercial off-the-shelf (MIDG) solution.

The outline of this paper is as follows: in Section II, the used wheeled mobile robot model is derived. Section III presents the trajectory controller design while section IV presents the outdoor navigation algorithm. The indoor navigation algorithm is presented in section V. Section VI presents an indoor/outdoor navigation algorithm followed by simulation and experimental results in Sections VII and VIII, respectively. Finally, conclusion remarks are given in Section IX.

II. WHEELED MOBILE ROBOT MODEL

The derivations of the kinematics of the mobile robot are based on the assumption that the robot is made up of a rigid frame equipped with non-deformable wheels that move on a horizontal plane [45]. In order to identify the robot position on the plane, two reference frames are used; the global reference frame on the plane and the local reference frame on the robot. These frames are shown in Figure 1. The axes X_I and Y_I define an arbitrary inertial basis on the plane as a global reference frame from an origin O. To specify the position of the robot, a point P on the robot's chassis is chosen as its position reference. The basis X_R and Y_R defines two axes passing through P on the robot's chassis and is the robot's local reference frame. The position of P in the global

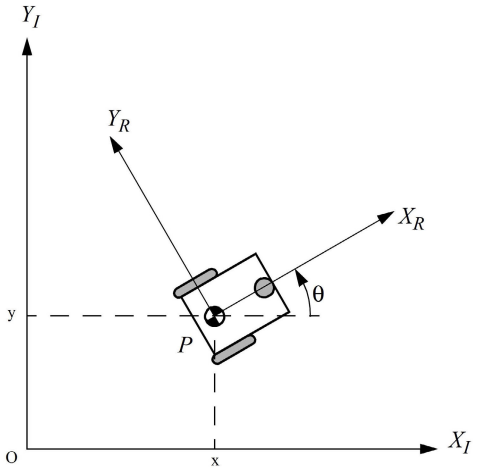


FIGURE 1. The global reference frame and the robot local reference frame [45].

reference frame is specified by coordinates x and y, and the angular difference between the global and local reference frames (i.e. the angle between X_R and X_I) is given by θ. The robot position is then fully identified by the three variables x, y, and θ. A 3×1-vector ζ is defined to describe the robot state with respect to the global frame and ζ_R with respect to the local frame. Also, the orthogonal rotation matrix $R(\theta)$ is defined as follows.

$$\zeta = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{1}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

So $R(\theta)$ defines the relation between $\dot{\zeta_I}$, and $\dot{\zeta_R}$ as follows.

$$\dot{\zeta_R} = R(\theta)\dot{\zeta_I} \tag{3}$$

The differential drive robot has two wheels, each with a diameter r. Given a point P centered between the two drive wheels, each wheel is at a distance l from P. Given r, l, θ, and the spinning speed of each wheel (ϕ_1) and (ϕ_2), a forward kinematic model for the robot's overall speed in the global reference frame can be described as

$$\dot{\zeta_I} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T = f(l, r, \theta, \dot{\phi}_1, \dot{\phi}_2) \tag{4}$$

Before presenting the derivation of the kinematic model of the WMR, two constraints will be presented for every fixed wheel. Figure 2 shows a wheel A and describes its pose with respect to the local reference frame, where α and β determine the orientation of the wheel with respect to the local frame. The first constraint enforces the concept of rolling contact, i.e. that the wheel must roll when motion takes place in the appropriate direction; represented by Equation (5). The second constraint enforces the concept of no lateral slippage i.e. that the wheel must not slide orthogonal to the wheel plane; represented by Equation (6) [45].

$$\begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta \end{bmatrix} R(\theta)\dot{\zeta_I} - r\dot{\phi} = 0 \tag{5}$$

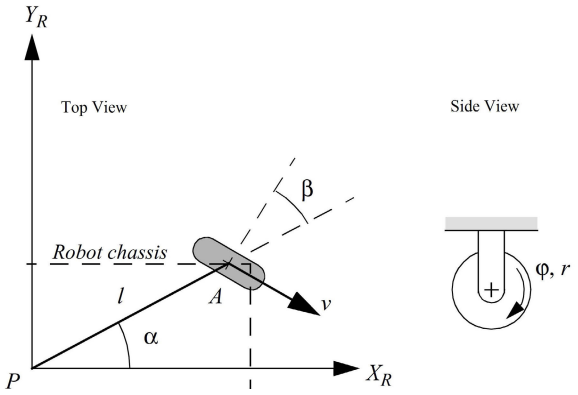


FIGURE 2. A fixed standard wheel and its parameters [45].

$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\zeta}_I = 0 \tag{6}$$

The combination of the wheel rolling and sliding constraints for all wheels of the robot describes the kinematic behavior and the resulting equation is:

$$\dot{\zeta}_1 = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix} \tag{7}$$

The linear velocity (v) of the robot is always along with X_R direction due to the nonholonomic constraint, and the angular velocity (ω) is the rotation speed of the local reference frame with respect to the global reference frame [47]. By writing Equation (7) in terms of v and ω we obtained Equation (8)

$$\dot{\zeta}_1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\,sin\theta \\ \omega \end{bmatrix} \tag{8}$$

where,

$$v = \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \tag{9}$$

$$\omega = \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \tag{10}$$

III. TRAJECTORY TRACKING

To perform trajectory tracking of WMRs, kinematic models are used to design feedback laws. The output from the controller is then used by the existing low-level PID controller on the platform as the velocity reference for each wheel. The basic control scheme of the system is made of a high-level controller, which is based on the kinematic model of the robot that has to take into account the constraints introduced by the wheels. The complexity of the high-level controller is raised by the nonlinearity that exists in the system. The control system for trajectory tracking using a WMR is made of a high-level controller which computes the linear and angular velocity which then sends them to a low-level PID controller as an input. The low-level controller then controls the motors in a way that the desired linear and angular velocity is achieved.

In order to perform trajectory tracking of the WMR, input-output state feedback linearization is presented by considering a point B (x_b,y_b) outside the wheel axle of the unicycle model with distance b, as a point of reference for the WMR, which is shown in Figure 3. In doing so, it is possible to control the robot's motion with a constant linear velocity regardless of the path curvature [44].

From Figure 3 we obtained

$$\begin{aligned} x_b &= x + bcos\theta \\ y_b &= y + bsin\theta \end{aligned} \tag{11}$$

The derivatives of Equation (11) are shown as follows.

$$\begin{aligned} \dot{x}_b &= \upsilon cos\theta - \omega bsin\theta \\ \dot{y}_b &= \upsilon sin\theta + \omega bcos\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{12}$$

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} cos\theta & -bsin\theta \\ sin\theta & bcos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{13}$$

$$\begin{vmatrix} cos\theta & -bsin\theta \\ sin\theta & bcos\theta \end{vmatrix} = b \neq 0 \tag{14}$$

For the matrix in Equation (13) to be invertible, b should not be equal to zero as shown in Equation (14). Subsequently, the linear and angular velocity is as given in Equation 15.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{b}\sin\theta & \frac{1}{b}\cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \end{bmatrix} \tag{15}$$

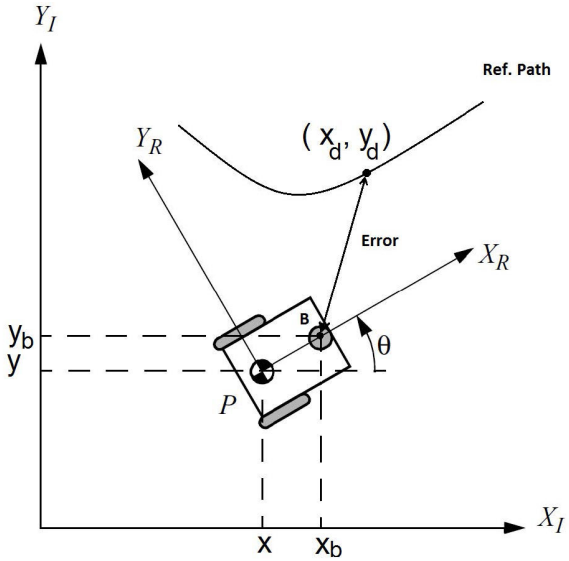


FIGURE 3. Control problem description.

Given a trajectory (x_d, y_d) as shown in Figure 3, it is possible to find input $\dot{x}_b$ and $\dot{y}_b$ that guarantee

asymptotic tracking. This can be seen as follows.

$$\dot{x}_b = \dot{x}_d + k_1 e_x$$
$$\dot{y}_b = \dot{y}_d + k_2 e_y \tag{16}$$

where,

$$e_x = x_d - x_b$$
$$e_y = y_d - y_b \tag{17}$$

By rearranging Equation (16), the following equations are obtained

$$\dot{e}_x + k_1 e_x = 0$$
$$\dot{e}_y + k_2 e_y = 0 \tag{18}$$

By solving Equation (18), we reach.

$$e_x = c_1 e^{-k_1 t}$$
$$e_y = c_2 e^{-k_2 t} \tag{19}$$

For positive k_1 and k_2,

$$\lim_{t\to\infty} e_x(t) = 0$$
$$\lim_{t\to\infty} e_y(t) = 0 \tag{20}$$

Then the system is asymptotically stable, and the error always converges to zero.

In addition, the controller performance is enhanced by adding a derivative term into control law Equation 16 to derive Equation (21)

$$\dot{x}_b = \dot{x}_d + k_1 e_x + k_3 \dot{e}_x$$
$$\dot{y}_b = \dot{y}_d + k_2 e_y + k_4 \dot{e}_y \tag{21}$$

where,

$$\dot{e}_x = \dot{x}_d - \dot{x}_b$$
$$\dot{e}_y = \dot{y}_d - \dot{y}_b \tag{22}$$

By rearranging Equation 21, the following equations are obtained.

$$\dot{e}_x + \frac{k_1}{1+k_3} e_x = 0$$
$$\dot{e}_y + \frac{k_2}{1+k_4} e_y = 0 \tag{23}$$

By solving Equation (23) for e_x and e_y, we obtained

$$e_x = c_3 e^{-\frac{k_1}{1+k_3} t}$$
$$e_y = c_4 e^{-\frac{k_2}{1+k_4} t} \tag{24}$$

For positive k_1,k_2,k_3 and k_4,

$$\lim_{t\to\infty} e_x(t) = 0$$
$$\lim_{t\to\infty} e_y(t) = 0 \tag{25}$$

Then the system is asymptotically stable and the error always converges to zero. The overall control scheme of WMR can be summarized in Figure 4.

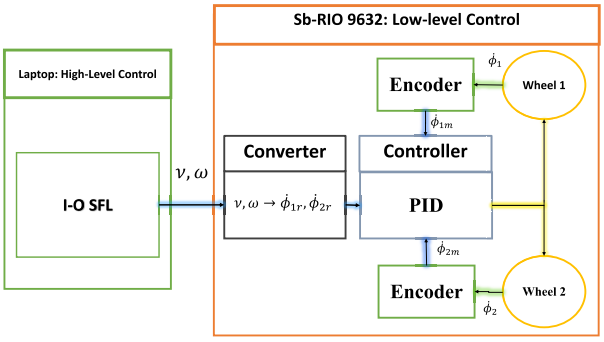


FIGURE 4. The overall control scheme of WMR.

IV. OUTDOOR NAVIGATION

In the outdoor environment, EKF will be derived and used for mobile robot localization and navigation, EKF algorithm will utilize the encoder, compass, IMU and GPS measurements to estimate robot position and orientation. The outdoor system structure can be summarized as shown in Figure 5.

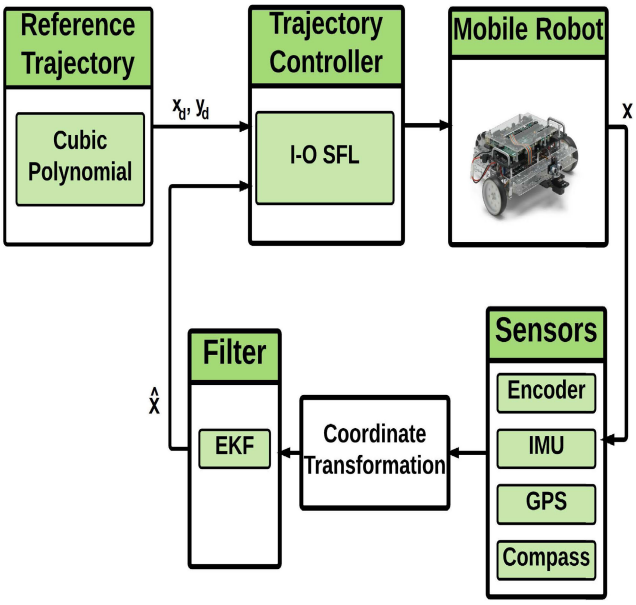


FIGURE 5. The outdoor system structure.

A. THE DYNAMIC MODEL

The first step of the filter design is to choose the system state whose elements represent the parameters that need to be estimated:

$$X = \begin{bmatrix} x & y & \theta & v & \omega \end{bmatrix}^T \tag{26}$$

The motion model Equation (8) derived in Section II can be used to model the system as follows:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \\ 0 \\ 0 \end{bmatrix} \tag{27}$$

Equation (27) is discretized, with a sampling time (Δt), to give the discrete-time model as:

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v\Delta t \cos\theta_{k-1} \\ y_{k-1} + v\Delta t \sin\theta_{k-1} \\ \theta_{k-1} + \omega\Delta t \\ v_{k-1} \\ \omega_{k-1} \end{bmatrix} \tag{28}$$

The system model and measurement models are given in Equation (29) and Equation (30), respectively.

$$X_k = f(X_{k-1}) + w_{k-1} \tag{29}$$

$$z_k = h(X_k) + \upsilon_k \tag{30}$$

where f and h represent the nonlinear system and measurement models, respectively. The dynamic system noise w_{k-1}, and measurement noise, υ_k are both zero mean Gaussian noise with associated covariance Q_k and R_k matrices, respectively.

$$w_k \sim N(0, Q_k) \tag{31}$$

$$\upsilon_k \sim N(0, R_k) \tag{32}$$

In order to use nonlinear functions f and h, the functions must be linearized. By taking the Jacobeans of f and h at the operating point, X_k during each time step. These matrices are:

$$F_k = \frac{\partial f(X)}{\partial X}|_{\hat{X}_k^-} \tag{33}$$

$$F_k = \begin{bmatrix} 1 & 0 & -v_k\Delta t \sin\theta_k & \Delta t \cos\theta_k & 0 \\ 0 & 1 & v_k\Delta t \cos\theta_k & \Delta t \sin\theta_k & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{34}$$

Prediction:

$$\hat{X}_k^- = f(\hat{X}_{k-1}^+) \tag{35}$$

$$P_k^- = F_k P_{k-1} F_k^T + Q_k \tag{36}$$

Update:

$$K_k = P_k^- H_k^T (H_k P_k^- H^T + R_k)^{-1} \tag{37}$$

$$\hat{X}_k^+ = \hat{X}_k^- + K_k(z_k - h(\hat{X}_k^-)) \tag{38}$$

$$P_k^+ = (I - K_k H_k) P_k^- \tag{39}$$

where $\hat{X}_k^-$, $\hat{X}_k^+$, P_k^- and P_k^+ represent priori, posteriori state estimate and their covariance matrices, respectively. K_k represents the Kalman gain and H_k represents measurement matrix.

B. THE MEASUREMENT MODEL

The measurement model is the mathematical representation of the sensors' measurements as a function of the system's state. For example, the encoder measures the wheel's velocity, the IMU measures the angular velocity and acceleration rate in gyroscope and accelerometer, respectively, the compass measures the heading, and the GPS measures the absolute position. The measurement equations are derived for each sensor as follows:

- **Encoder**
  Two encoders are mounted on each wheel of the differential mobile robot. Encoders measure the wheel's spinning velocity by counting the pulses per sampling period. This measurement is then converted to wheel linear velocity. Therefore, it will give the information about the displacement of the robot relative to its last known position, which in our case, will be calculated by multiplying the state vector given in Equation (28) with the following measurement matrix:

$$H_{encoder} = \begin{bmatrix} 0 & 0 & 0 & 1 & \frac{l}{2} \\ 0 & 0 & 0 & 1 & -\frac{l}{2} \end{bmatrix} \tag{40}$$

  where l is the distance between the wheels.
- **Compass**
  A compass provides the direction relative to the Earth's magnetic poles. Therefore, it can be used to measure heading θ. Then the associated measurement matrix is:

$$H_{compass} = \begin{bmatrix} 0 & 0 & 1-\theta_0 & 0 & 0 \end{bmatrix} \tag{41}$$

  where θ_0 is the deference angle between robot orientation and north pole.
- **GPS**
  The GPS is used to measure the location of the robot in x and y. Therefore, with the GPS receiver's antenna fixed on the center of the robot's coordinate system, the measurement matrix would be as simple as:

$$H_{GPS} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{42}$$

- **IMU**
  The IMU is an electronics module which measures angular velocity and linear acceleration along the body's three perpendicular axes. The IMU contains three sensors: accelerometer, gyroscopes, and magnetometers.
  - **Gyroscope**
    A gyroscope is a device that measures the angular rate ω. Then the measurement matrix is:

$$H_{gyro} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{43}$$

    And by the integration, the heading θ can be obtained. Then the measurement matrix is given by:

$$H_{heading} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{44}$$

  - **Accelerometer**
    An accelerometer is a device that measures specific force, by integration, the linear velocity v can be obtained. Then the measurement matrix is:

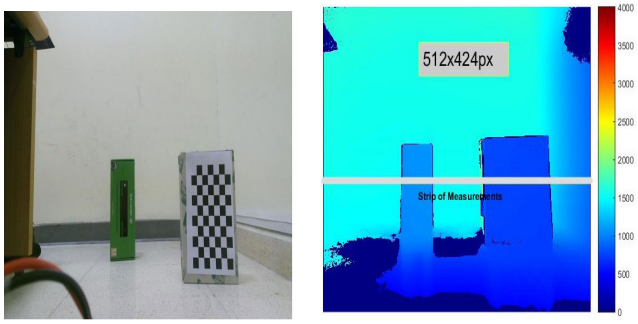
$$H_{acc.} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{45}$$

V. INDOOR NAVIGATION

The navigation solution that was discussed in the last section relies on position information received by the GPS. It is useless when the robot is in an indoor environment. Hence, separate solutions have been proposed in this section that can accurately estimate the robot's pose. Here, we mainly discuss the localization algorithms EKF for the indoor environment using Kinect V2 measurements given a predefined map. Feature map representation is adopted, and boxes placed in the environment are assigned as features.

Microsoft Kinect provides RGB and depth images as shown in Figure 6a and Figure 6b, respectively. The technical specifications of depth sensor is shown in Table 1. An alternative approach is to extract features from raw measurements. In this research, the depth data was used to localize the robot. The beam range finder model is based on raw sensor measurements, this may lead to huge computational cost [33].

The feature extractor $f(z_t)$ extracts the feature from range measurement. Most feature extractors extract a small number of features from high-dimensional sensor measurements. The advantage of this approach is the massive reduction of computational load. An additional step for computational load reduction in our system is converting the 3D Kinect's image to 2D by taking a strip of the image parallel to the robot height assuming the height of the obstacles are equal or greater than the robot height as shown in Figure 6b.



(a) RGB image. (b) Depth image.

FIGURE 6. Kinect images.

TABLE 1. Kinect V2 technical specifications [48].

Camera resolution		512 x 424 pixels
Framerate		30 frames per second
Field of view	**Horizontal**	70 degrees
	Vertical	60 degrees
Operative measuring range		From 0.5 m to 4.5 m
Depth technology		Time of flight (ToF)

After converting the depth stream from 3D to 2D, the measurement data in polar form is $L_i = [r_i, \phi_i]$, where r_i is the value of horizontal distance between the reflecting point and the Kinect, and ϕ_i is the angular position which describes the single laser beam (where ϕ is between $-35°$ to $35°$ and i is the index of laser beam distribution from 1 to 512, i.e. the width of depth image.) The 2D data are passed through the clustering algorithm [49] based on difference calculation as follows

$$\Delta r_i = r_{i+1} - r_i \tag{46}$$

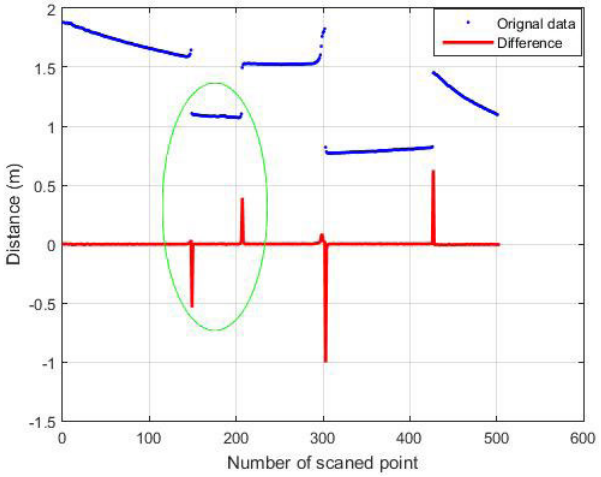


FIGURE 7. The difference vector and raw scan data.

where Δr_i is the difference vector representing the range between two neighboring laser beams L_i and L_{i+1}. As shown in Figure 7, significant changes will occur in the difference vector when the edges of objects are detected. After the difference calculation, the distinct objects are separated from the background in the vector curve as shown in Figure 7. Then the values of the vector are compared with depth threshold R_{max}, which is the maximum difference in the ranges that allows separating object from the background or another object. If two adjacent differences Δr_m and Δr_n satisfy the constraint shown in Equation (47), the depth cloud will be selected as an independent feature.

$$\begin{aligned} \Delta r_m &> R_{max} \\ \Delta r_n &< -R_{max} \\ N_{min} &< n - m < N_{max} \end{aligned} \tag{47}$$

where N_{max} is the greatest acceptable width of the object and N_{min} is the smallest acceptable width of the object. The laser points from the *m-th* to the *n-th* belong to one object and the others belong to the background or another object. Figure 8 shows the extracted features from the raw data. In robotics applications, features or landmarks correspond to distinct objects in the space. For example, landmarks may be room corners or table legs or boxes. The landmark model assumes that the sensor can measure the range and the bearing of the landmark relative to the robot's local coordinate frame. If we denote the range by r and the bearing by ϕ, the landmark vector is given by

$$f(z_t) = \left\{f_t^1, f_t^2, \ldots\right\} = \left\{\begin{pmatrix} r_t^1 \\ \phi_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \end{pmatrix}, \ldots\right\} \tag{48}$$

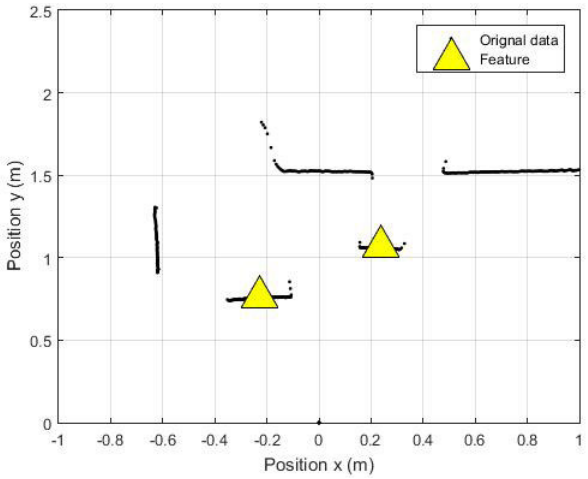


FIGURE 8. The extracted feature from depth steam.

The number of features identified at each time step is varied. However, we assume conditional independence between landmarks as shown in Equation 49. Under the conditional independence assumption, one feature can be processed at a time. This assumption helps in filter design.

$$p(f(z_t)|x_t, m) = \prod_i p(r_t^i, \phi_t^i|x_t, m) \tag{49}$$

Formulation of landmark measurement models are usually defined only for feature-based maps, $m = \{m_1, m_2, \ldots\}$. The location of a feature in global frame denoted as $m_{i,x}$ and $m_{i,y}$ for x and y coordinates respectively.

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2(m_{j,y} - y, m_{j,y} - x) - \theta \end{pmatrix} \tag{50}$$

The resulting measurement model is devised for the case where the $i - th$ feature at time t corresponds to the $j - th$ landmark in the map, with robot pose $x_t = (x \;\; y \;\; \theta)^T$.

In this section, an EKF is designed to fuse odometry, IMU and Kinect depth information to realize localization. The successful implementation of the EKF localization algorithm is demonstrated in Figure 9. The input parameters are $\bar{\mu}_{t-1}$, $\bar{\Sigma}_{t-1}$,u_t, z_t and m where $\bar{\mu}_{t-1}$ and $\bar{\Sigma}_{t-1}$ are respectively the estimated pose and covariance matrix of the pose at time $t-1$. z_t is the measurement vector and m is the map. EKF can be split into two different steps, prediction and update step.

- **Prediction step** (Lines 1-6): EKF localization algorithm uses the motion model derived in section II. G_t in line 2 is obtained by taking the Jacobian of motion model with respect to x, y, θ, receptively. V_t in line 3 is derived by taking the derivative of motion model with respect to the control input. In line 4, the dynamic noise covariance matrix is determined. Finally, the pose and its covariance matrix are predicted in line 5 and 6, respectively.
- **Update step** (Lines 7-20): The steps in lines 7-13 allow for obtaining a high-accuracy state estimate. For each measurement, we first calculate some quantities for all landmarks k in the map; they enable us to determine the most likely correspondence. The correspondence variable $j(i)$ is then assigned in line 14, by maximizing the likelihood of the measurement z_t^i given any possible landmark m_k in the map. As can be seen, in lines 9 and 10 we have the measurement model. In line 13, H is the Jacobian of the predicted measurement with respect to the robot location and is computed about the predicted pose $\bar{\mu}_t$.

Algorithm V.1: **EKF**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$

$1: \theta = \mu_{t-1,\theta}$

$2: G_t = \begin{bmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta + \omega\Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega\Delta t) \\ 0 & 0 & 1 \end{bmatrix}$

$3: V_t = \begin{bmatrix} \frac{-\sin\theta + \sin(\theta+\omega\Delta t)}{\omega t} & \frac{v_t(\sin\theta - \sin(\theta+\omega\Delta t)}{\omega_t^2} + \frac{v_t(\cos(\theta+\omega\Delta t)\Delta t}{\omega_t} \\ \frac{-\cos\theta + \cos(\theta+\omega\Delta t)}{\omega t} & -\frac{v_t(\cos\theta - \cos(\theta+\omega\Delta t)}{\omega_t^2} + \frac{v_t(\sin(\theta+\omega\Delta t)\Delta t}{\omega_t} \\ 0 & \Delta t \end{bmatrix}$

$4: M_t = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{bmatrix}$

$5: \bar{\mu}_{t-1} = \mu_{t-1} + \begin{bmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega\Delta t) \\ \omega_t \Delta t \end{bmatrix}$

$6: \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$

$7:$ **for** all observed features $z_t^i = (r_t^i \phi_t^i)$ *do*

$8:$ **for** all landmarks k in the map m **do**

$9: \quad q = (m_{k,x} - \bar{\mu}_{t,x})^2 + (m_{k,y} - \bar{\mu}_{t,y})^2$

$10: \quad \hat{z}_t^k = \begin{bmatrix} \sqrt{q} \\ atan2(m_{k,y} - \bar{\mu}_{t,y}, m_{k,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \end{bmatrix}$

$11: \quad H_k = \begin{bmatrix} -\frac{m_{k,x} - \bar{\mu}_{t,x})}{\sqrt{q}} & -\frac{m_{k,y} - \bar{\mu}_{t,y})}{\sqrt{q}} \\ \frac{m_{k,y} - \bar{\mu}_{t,y})}{q} & -\frac{m_{k,x} - \bar{\mu}_{t,x})}{q} \end{bmatrix}$

$12: \quad S_t^k = H_t^k \Sigma_t [H_t^k]^T + Q_t]$

$13: \quad endfor$

$14: j(i) = \text{argmax det } (2\pi S_t^k)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t^i - \hat{z}_t^k)^T [S_t^k]^{-1} (z_t^i - \hat{z}_t^k)\right\}$

$15: K_t^i = \bar{\Sigma}_t [H_t^{j(i)}]^T [S_t^{j(i)}]^{-1}$

$16: \bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z_t^{j(i)}})$

$17: \bar{\Sigma}_t = (I - K_t^i [H_t^{j(i)}]) \bar{\Sigma}_t$

$18: endfor$

$19: \mu_t = \bar{\mu}_t$

$20: \Sigma_t = \bar{\Sigma}_t$

return (μ_t, Σ_t)

FIGURE 9. The EKF localization algorithm with unknown correspondences [17].

The EKF is used to localize the robot in a feature-based map at any time t the robot observes information, z_t, as ranges and bearing of nearby features. This algorithm uses the feature-based model as the measurement model. It will start with mean and covariance from the previous time step $t - 1$, then the Jacobians G_t and V_t are obtained from the motion model. Similarly, the dynamics noise covariance matrix is determined. Subsequently, the predicted pose and covariance are calculated as $\bar{\mu}_t$ and $\bar{\Sigma}_t$, respectively. The core of the correction step is a loop through all features i observed at time t.

The hardest problem in the implementation of an EKF localization algorithm in feature-based map localization is measurement correspondences, which means how the

algorithm determines which measurement corresponds to which landmark. To solve this problem, maximum likelihood estimator is used. Maximum likelihood estimator determines the most likely value of the correspondence variable that maximizes the data likelihood as shown in Equation (51).

$$\hat{c}_t = \underset{c_t}{\operatorname{argmax}}\, p(z_t|c_{1:t}, m, z_{1:t-1}, u_{1:t}) \tag{51}$$

where c_t is the correspondence at time t and u is the input vector to WMR contains linear and angular velocities. The indoor system structure can be summarized in Figure 10.

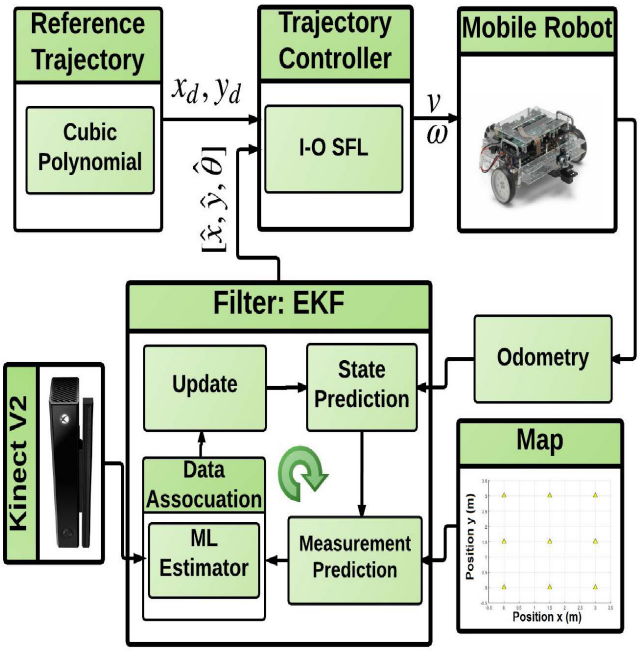


FIGURE 10. EKF indoor system structure.

VI. INDOOR/OUTDOOR NAVIGATION

Many solutions presented in this paper are proved to be useful in their environments. This section discusses the integration of the two systems and provides a solution for both indoor and outdoor environments.

The challenge in the hybrid system is how the robot decides which algorithm to use and when to use it. The proposed solution requires information for the whole map (i.e. the outdoor and indoor regions are specified). By having such information, switching safely between indoor and outdoor algorithms will become handy. The environment's regions are shown in Figure 11.

For example, suppose that the robot starts from an outdoor environment and the robot uses the algorithm discussed in section IV to localize itself. When the robot reaches the indoor region, it will automatically switch to the indoor algorithm that was discussed in section V. By doing so, the robot will use the last pose estimate from the outdoor algorithm as the initial pose estimate in the indoor algorithm.

When the robot nears to the indoor environment, we encounter new issues in deciphering the correct algorithm.

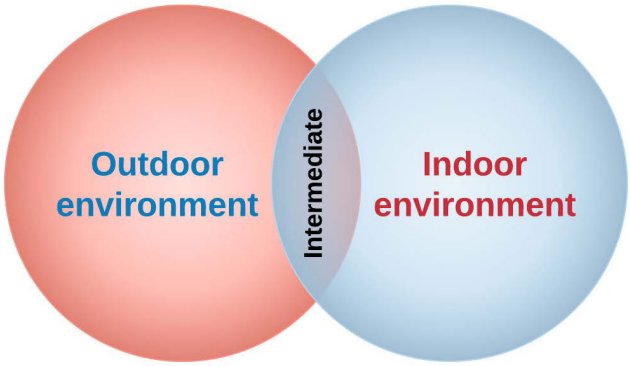


FIGURE 11. The environments regions.

Therefore, we exclude the use of the GPS in the intermediate region by skipping the update step in the EKF algorithm. Moreover, this can be considered when the GPS signals are blocked or scattered as seen when the difference between the GPS reading at current time step and previous time step is greater than a specific threshold. Additionally, the covariance matrix of the estimate is monitored and if the value exceeds a specific threshold λ, the algorithm will turn off the outdoor localization algorithm. By doing so, in the intermediate region, the robot's position is estimated based on encoder and IMU measurements only. The indoor/outdoor algorithm can be summarized in the chart shown in Figure 12.

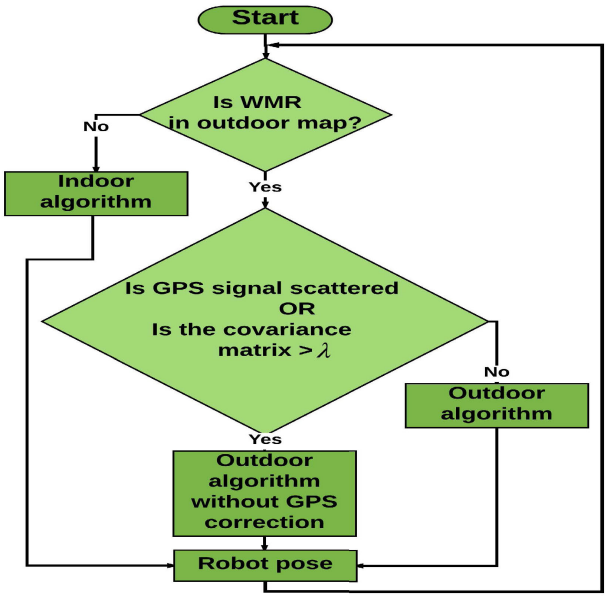


FIGURE 12. Indoor/outdoor algorithm.

VII. SIMULATION RESULTS

The proposed solution for the hybrid system is tested in scenarios where the robot, sensors, and the environment are simulated. The robot starts from an outdoor environment where the pose estimate is obtained using EKF, which combines GPS, compass and encoder measurements. Once the robot

nears to the indoor environment, the GPS signal is blocked by the near building. The hybrid localization algorithm should switch to the indoor algorithm that relies on landmarks that are placed on the corners. Therefore, there is no need for the GPS signal to localize the robot.

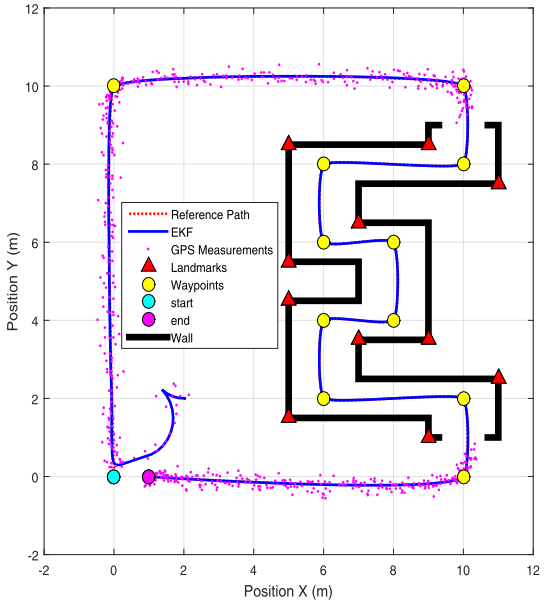


FIGURE 13. Indoor/outdoor Simulation Robot path in indoor/outdoor environments.

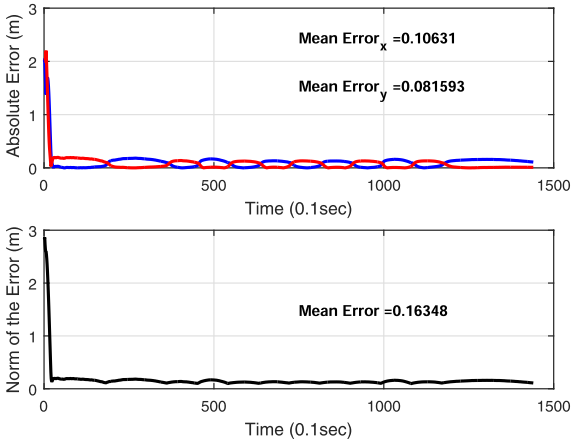


FIGURE 14. Indoor/outdoor Simulation: error results.

Figure 13 shows the performance of the hybrid system where the robot tracks the reference path both in indoor and outdoor environments, where the dotted red and the blue lines represent the reference trajectory and the EKF path, respectively. Additionally, the magenta dots, and red triangle and thick black line represent the simulated GPS signal, landmarks, and the walls, respectively. Moreover, a Monte Carlo simulation is performed, and the absolute error and distance error between the robot and the reference trajectory are obtained in Figure 14. The figure shows a promising result where the mean error of the norm is 0.163 m.

VIII. EXPERIMENTAL RESULTS

To test the performance of the proposed HLRNS, DaNI 2.0 robot is used with IMU (e.g MIDG), GPS antenna and Microsoft Kinect 2.0. Note that the MIDG is a GPS aided inertial navigation system (INS). The INS fusion filter uses measurements from the internal GPS receiver, internal magnetometer, and an external heading source to produce accurate estimates of position, velocity, and attitude [50]. The robot and the sensors are shown in Figure 15, where (1) is the MIDG (which includes IMU, Compass and GPS antenna), (2) is a Microsoft Kinect 2.0, (3) is DaNI 2.0 robot. To track the reference trajectory, I-O SFL controller is used. In order to observe the robot functioning during the navigation process, a Graphical User Interface (GUI) is designed for this purpose. As shown in Figure 16, GUI displays the sensors' information and the robot pose. Panel 1 is the information about the serial ports status and its configurations; likewise, panel 2 is the sensors (encoders and MIDG) readings. Panel 3 represents robot states during the process, while panel 4 represents the robot control method such as the trajectory tracking and the pose control. Finally, panel 5 represents the robot graphical representation on the grid.

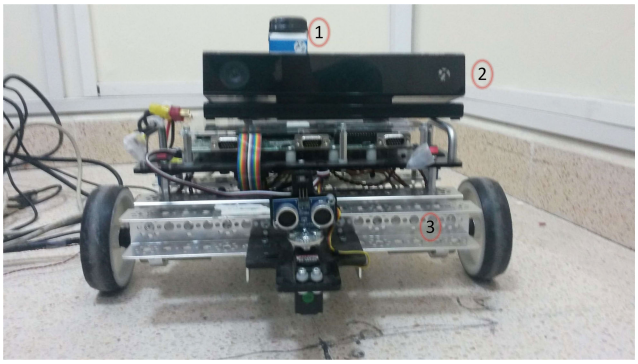


FIGURE 15. DaNI robot with MIDG and Kinect.

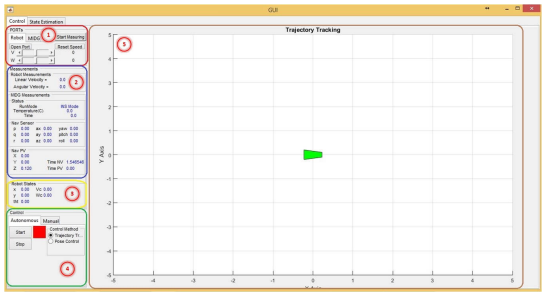


FIGURE 16. Indoor/outdoor algorithm.

The HLRNS was tested on the bridge between the two engineering buildings at the American University of Sharjah (AUS) as shown in the red circle in Figure 17. For the indoor environment, eight boxes were placed in a specific known location and were used as landmarks, see Figure 18.

Figure 19 shows the experimental results where the red, green, blue and dashed cyan lines represent the reference

FIGURE 17. The experiment environment at the bridge between the two engineering buildings in AUS.

(a) From outdoor to indoor. (b) From indoor to outdoor.

FIGURE 18. Indoor/Outdoor Experimental environment.

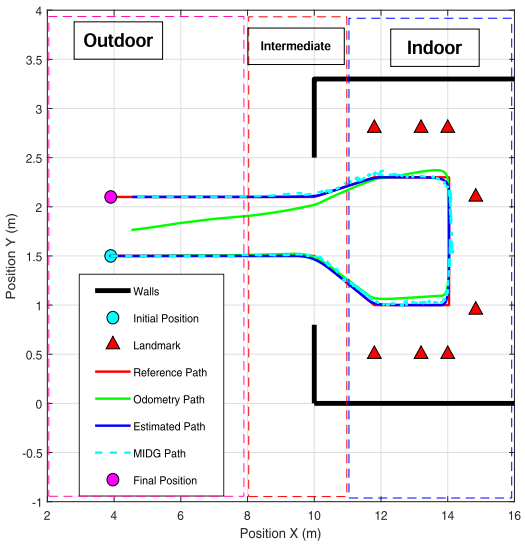


FIGURE 19. Robot path in indoor/outdoor environments.

trajectory, odometry path, estimated path (e.g the proposed method) and MIDG path, respectively. Similarly, the red triangle and thick black line represent the landmarks and the building walls, respectively. Furthermore, the cyan and magenta circles represent the initial and final positions, respectively. As shown in Figure 19, there are three regions: outdoor region, where the GPS signal is available and reliable; indoor region, where the landmarks are placed properly; and the intermediate region, which is the region that cannot

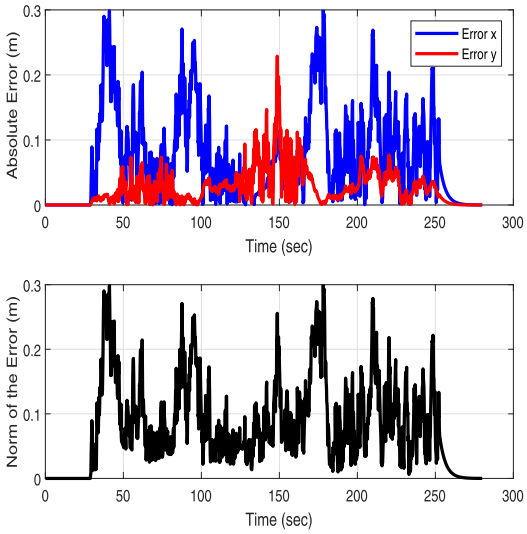


FIGURE 20. Indoor/outdoor error results.

be considered as an outdoor nor indoor environment. As can be seen in Figure 19, DaNI robot started from the outdoor region and fused the available measurements (encoder, IMU and GPS) using the outdoor EKF algorithm to localize itself while tracking the reference trajectory. Once the robot entered the intermediate region, the GPS signal started scattering and the associated covariance increased rapidly. Therefore, the best solution in this region was to combine the encoder with the compass using the EKF while neglecting GPS information until the robot entered the indoor environment. In the indoor environment, the robot started observing the surrounding environment and extracting the features to use as landmarks in the indoor EKF algorithm. When the robot exited the indoor region and enters the intermediate region leading to the outdoor region, the robot started to receive the GPS signal again. Hence, the algorithm combined the GPS signal with the other measurements to localize the robot. For comparison, we used the MIDG solution as a performance reference (e.g ground truth). The MIDG solution uses tight coupling in its Unscented Kalman Filter (UKF) structure where the raw pseudo-ranges are utilized. The UKF includes various flags in the output to determine the quality of the solution [46]. Figure 20 shows the absolute error and norm of the error between the estimated position of the proposed method and the MIDG solution. As can be seen from the figure, the robot successfully tracked the desired trajectory in indoor and outdoor environments within 0.3 accuracy.

IX. CONCLUSION

In this paper, a hybrid low-cost reduced navigation system (HLRNS) which combines indoor and outdoor algorithm based on EKF was developed for a low-cost sensors system. The proposed approach was tested in both simulation and experimental environments. The trajectory tracking problem was solved using I-O SFL and was enhanced by adding the derivative term. For the outdoor environment, EKF was

used to estimate the robot's pose based on wheel encoders, a reduced inertial sensor system (RISS), and Global Positioning System (GPS). For the indoor environment, where GPS signals are blocked, the Microsoft Kinect V2 sensor was used in the EKF algorithm to localize the robot in feature-based map representation with the other low-cost sensors. EKF was used to estimate the robot's pose relative to landmarks extracted from the map. The results were verified in experimental testing. The performance of the fusion architecture is compared against a commercial off-the-shelf (MIDG) solution that uses UKF as a fusion engine. It was seen that the robot can successfully track the desired trajectory within 0.3 m accuracy.

REFERENCES

[1] W. Lv, Y. Kang, and J. Qin, "Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 6, pp. 1241–1253, Jun. 2019.

[2] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robot. Syst.*, vol. 14, no. 4, pp. 231–249, Apr. 1997.

[3] H. Azartash, N. Banai, and T. Q. Nguyen, "An integrated stereo visual odometry for robotic navigation," *Robot. Auto. Syst.*, vol. 62, no. 4, pp. 414–421, Apr. 2014.

[4] B. Barshan and H. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 11, no. 3, pp. 328–342, Jun. 1995.

[5] Y. S. Khraisat, M. Al-Khateeb, Y. Abu-Alreesh, A. Ayyash, and O. Lahlouh, "Gps navigation and tracking device," *Int. J. Interact. Mobile technol.*, vol. 5, no. 4, pp. 39–41, 2011.

[6] C. Magnusson, K. Rassmus-Gröhn, and D. Szymczak, "Navigation by pointing to GPS locations," *Pers. Ubiquitous Comput.*, vol. 16, no. 8, pp. 959–971, Dec. 2012.

[7] R. Zhu and H. A. Karimi, "Automatic selection of landmarks for navigation guidance," *Trans. GIS*, vol. 19, no. 2, pp. 247–261, Apr. 2015.

[8] C. Premachandra, M. Murakami, R. Gohara, T. Ninomiya, and K. Kato, "Improving landmark detection accuracy for self-localization through baseboard recognition," *Int. J. Mach. Learn. Cyber.*, vol. 8, no. 6, pp. 1815–1826, Dec. 2017.

[9] W. Eom, J. Park, and J. Lee, "Hazardous area navigation with temporary beacons," *Int. J. Control Autom. Syst.*, vol. 8, no. 5, pp. 1082–1090, Oct. 2010.

[10] K. Nakajima, C. Premachandra, and K. Kato, "3D environment mapping and self-position estimation by a small flying robot mounted with a movable ultrasonic range sensor," *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 2, pp. 289–298, Sep. 2017.

[11] Y. Xu, Y. S. Shmaliy, Y. Li, X. Chen, and H. Guo, "Indoor INS/LiDAR-based robot localization with improved robustness using cascaded FIR filter," *IEEE Access*, vol. 7, pp. 34189–34197, 2019.

[12] K. Warnakulasooriya, B. Sudantha, and C. Premachandra, "A color mask and trained image set for the creation of new technique for indoor robotic navigation," in *Proc. 3rd Int. Conf. Inf. Technol. Res.(ICITR)*, Dec. 2018, pp. 1–4.

[13] M. F. Abdel-Hafez, "On the development of an inertial navigation error-budget system," *J. Franklin Inst.*, vol. 348, no. 1, pp. 24–44, Feb. 2011.

[14] M. A. K. Jaradat and M. F. Abdel-Hafez, "Enhanced, delay dependent, intelligent fusion for INS/GPS navigation system," *IEEE Sensors J.*, vol. 14, no. 5, pp. 1545–1554, May 2014.

[15] H. B. Mitchell, *Multi-Sensor Data Fusion: An Introduction*, 1st ed. New York, NY, USA: Springer, 2007.

[16] J. G. N. D. Carvalho Filho, E. A. N. Carvalho, L. Molina, and E. O. Freire, "The impact of parametric uncertainties on mobile robots velocities and pose estimation," *IEEE Access*, vol. 7, pp. 69070–69086, 2019.

[17] S. Thrun, W. Burgard, D. Fox, R. C. Arkin, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. The Cambridge, MA, USA: MIT Press, 2005.

[18] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field," *Soft Comput.*, vol. 16, no. 1, pp. 153–164, Jan. 2012, doi: 10.1007/s00500-011-0742-z.

[19] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 135–149, Feb. 2011, doi: 10.1016/j.rcim.2010.06.019.

[20] E. I. Al Khatib, M. A. Jaradat, M. Abdel-Hafez, and M. Roigari, "Multiple sensor fusion for mobile robot localization and navigation using the extended Kalman filter," in *Proc. 10th Int. Symp. Mechatronics Appl. (ISMA)*, Dec. 2015, pp. 1–5.

[21] X. Bai, Z. Zhang, L. Liu, X. Zhai, J. Panneerselvam, and L. Ge, "Enhancing localization of mobile robots in distributed sensor environments for reliable proximity service applications," *IEEE Access*, vol. 7, pp. 28826–28834, 2019.

[22] M. A. K. Jaradat and R. Langari, "A hybrid intelligent system for fault detection and sensor fusion," *Appl. Soft Comput.*, vol. 9, no. 1, pp. 415–422, Jan. 2009.

[23] M. Lin, C. Yang, D. Li, and G. Zhou, "Intelligent filter-based SLAM for mobile robots with improved localization performance," *IEEE Access*, vol. 7, pp. 113284–113297, 2019.

[24] M. A. K. Jaradat and M. F. Abdel-Hafez, "Non-linear autoregressive delay-dependent ins/gps navigation system using neural networks," *IEEE Sensors J.*, vol. 17, no. 4, pp. 1105–1115, Feb. 2017.

[25] K. Saadeddin, M. F. Abdel-Hafez, M. A. Jaradat, and M. A. Jarrah, "Optimization of intelligent approach for low-cost INS/GPS navigation system," *J. Intell. Robot. Syst.*, vol. 73, nos. 1–4, pp. 325–348, Jan. 2014.

[26] J. Woo, Y.-J. Kim, J.-O. Lee, and M.-T. Lim, "Localization of mobile robot using particle filter," in *Proc. SICE-ICASE Int. Joint Conf.*, Oct. 2006, pp. 3031–3034.

[27] P. Zingaretti and E. Frontoni, "Vision and sonar sensor fusion for mobile robot localization in aliased environments," in *Proc. 2nd IEEE/ASME Int. Conf. Mechatronics Embedded Syst. Appl.*, Aug. 2006, pp. 1–6.

[28] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[29] B. E. Hughes, "A navigation subsystem for an autonomous robot lawn mower," Ph.D. dissertation, Dept. Elect. Eng., Case Western Reserve Univ., Cleveland, OH, USA, 2011.

[30] K. Saadeddin, M. F. Abdel-Hafez, M. A. Jaradat, and M. A. Jarrah, "Optimization of intelligent-based approach for low-cost INS/GPS navigation system," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2013, pp. 668–677.

[31] G. Lakemeyer and B. Nebel, Eds., *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA, USA: Morgan Kaufmann, 2003.

[32] F. Alkhawaja, M. Jaradat, and L. Romdhane, "Techniques of indoor positioning systems (IPS): A survey," in *Proc. Adv. Sci. Eng. Technol. Int. Conferences (ASET)*, Mar. 2019, pp. 1–8.

[33] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual slam with a hand-held RGB-D camera," in *Proc. RGB-D Workshop 3D Perception Robot. Eur. Robot. Forum*, Västerås, Sweden, vol. 180, 2011, pp. 1–15.

[34] A. Oliver, S. Kang, B. C. WÜnsche, and B. Macdonald, "Using the Kinect as a navigation sensor for mobile robotics," in *Proc. 27th Conf. Image Vis. Comput. New Zealand (IVCNZ)*. New York, NY, USA: ACM, 2012, pp. 509–514.

[35] A. Georgiev and P. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 851–864, Oct. 2004.

[36] M. Roigari, "A navigation system for indoor/outdoor environments with an unmanned ground vehicle (UGV)," M.S. thesis, Dept. Mechtron. Program, Amer. Univ. Sharjah, Sharjah, United Arab Emirates, 2015. [Online]. Available: http://library.aus.edu/

[37] Y. Lu and D. Song, "Visual navigation using heterogeneous landmarks and unsupervised geometric constraints," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 736–749, Jun. 2015.

[38] A. Brooks, A. Makarenko, and B. Upcroft, "Gaussian process models for indoor and outdoor sensor-centric robot localization," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1341–1351, Dec. 2008.

[39] S. Sun, T. Endo, and F. Matsuno, "Iterative learning control based robust distributed algorithm for non-holonomic mobile robots formation," *IEEE Access*, vol. 6, pp. 61904–61917, 2018.

[40] F. Yan, B. Li, W. Shi, and D. Wang, "Hybrid visual servo trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 6, pp. 24291–24298, 2018.
[41] C.-L. Hwang, H.-M. Wu, and W.-H. Hung, "Software/hardware-based hierarchical finite-time sliding-mode control with input saturation for an omnidirectional autonomous mobile robot," *IEEE Access*, vol. 7, pp. 90254–90267, 2019.
[42] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access*, vol. 6, pp. 43414–43424, 2018.
[43] E. I. Al Khatib, W. M. F. Al-Masri, S. Mukhopadhyay, M. A. Jaradat, and M. Abdel-Hafez, "A comparison of adaptive trajectory tracking controllers for wheeled mobile robots," in *Proc. 10th Int. Symp. Mechatronics Appl. (ISMA)*, Dec. 2015, pp. 1–6.
[44] B. d'Andréa-Novel, G. Campion, and G. Bastin, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *Int. J. Robot. Res.*, vol. 14, no. 6, pp. 543–559, Dec. 1995.
[45] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2011.
[46] G. H. Elkaim, M. Lizarraga, and L. Pederseny, "Comparison of low-cost GPS/INS sensors for autonomous vehicle applications," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2008, pp. 1133–1144.
[47] E. Al Khatib, "A navigation and control system for a robot in indoor/outdoor environments," M.S. thesis, Dept. Mechtron. Program, Amer. Univ. Sharjah, Sharjah, United Arab Emirates, 2016.
[48] J. Jiao, L. Yuan, W. Tang, Z. Deng, and Q. Wu, "A post-rectification approach of depth images of kinect v2 for 3D reconstruction of indoor scenes," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 11, p. 349, Nov. 2017.
[49] J. Kong, X. Ding, J. Liu, L. Yan, and J. Wang, "New hybrid algorithms for estimating tree stem diameters at breast height using a two dimensional terrestrial laser scanner," *Sensors*, vol. 15, no. 7, pp. 15661–15683, Jul. 2015.
[50] *MIDG II Operating Modes*, Microbotics, Hampton, VA, USA, 2004.

EHAB I. AL KHATIB received the B.S. degree in mechanical engineering with a minor in mechatronics from the Jordan University of Science and Technology, Irbid, Jordan, in 2014, and the M.S. degree in mechatronics from the American University of Sharjah, Sharjah, UAE, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering with Southern Methodist University, Dallas, TX.

His current research interests include mechatronics, robotics, nonlinear control, navigation, and localization.

MOHAMMAD ABDEL KAREEM JARADAT received the B.Sc. degree from the Jordan University of Science and Technology, Jordan, and the M.Sc. and Ph.D. degrees in mechanical engineering from Texas A&M University, College Station, TX, USA. He is currently with the American University of Sharjah and the Jordan University of Science and Technology. During his accumulated experience, several projects, prototypes, and publications are conducted, specialized on the following research areas such as robotics, artificial intelligent systems, mechatronics system design, sensor fusion, fault diagnostics, intelligent nano-systems, intelligent control, and embedded control systems.

MAMOUN F. ABDEL-HAFEZ (Senior Member, IEEE) received the B.S. degree in mechanical engineering from the Jordan University of Science and Technology, in 1997, the M.S. degree in mechanical engineering from the University of Wisconsin, Milwaukee, in 1999, and the Ph.D. degree in mechanical engineering from the University of California at Los Angeles (UCLA), in 2003. He served as a Postdoctoral Research Associate with the Department of Mechanical and Aerospace Engineering, UCLA, in 2003, where he was involved in a research project on fault-tolerant autonomous multiple aircraft landing. He is currently with the Department of Mechanical Engineering, American University of Sharjah. His research interests are in stochastic estimation, control systems, sensor fusion, and fault detection.