computers

MDPI

Article

State Estimation and Localization Based on Sensor Fusion for Autonomous Robots in Indoor Environment

Mamadou Doumbia * and Xu Cheng

School of Computer Science and Electronic Engineering, Hunan University, Changsha 420082, China; cheng_xu@yeah.net

* Correspondence: cadip8@yahoo.fr; Tel.: +86-151-7312-4908

Received: 1 October 2020; Accepted: 13 October 2020; Published: 16 October 2020

Abstract: Currently, almost all robot state estimation and localization systems are based on the Kalman filter (KF) and its derived methods, in particular the unscented Kalman filter (UKF). When applying the UKF alone, the estimate of the state is not sufficiently precise. In this paper, a new hierarchical infrared navigational algorithm hybridization (HIRNAH) system is developed to provide better state estimation and localization for mobile robots. Two navigation subsystems (inertial navigation system (INS) and, using a novel infrared navigation algorithm (NIRNA), Odom-NIRNA) and an RPLIDAR-A3 scanner cooperation to build HIRNAH. The robot pose (position and orientation) errors are estimated by a system filtering module (SFM) and used to smooth the robot's final poses. A prototype (two rotary encoders, one smartphone-based robot sensing model and one RPLIDAR-A3 scanner) has been built and mounted on a four-wheeled mobile robot (4-WMR). Simulation results have motivated real-life experiments, and obtained results are compared to some existent research (hardware and control technology navigation (HCTNav), rapid exploring random tree (RRT) and in stand-alone mode (INS)) for performance measurements. The experimental results confirm that HIRNAH presents a more accurate estimation and a lower mean square error (MSE) of the robot's state than those calculated by the previously cited HCTNav, RRT and INS.

Keywords: odometry; INS; RPLIDAR-A3 scanner; tightly hybridization technique; UKF; 4-WMR

1. Introduction

Today, we can almost say that robots are used in all areas of human life. From the military, industrial and even domestic fields, robots are deployed in all those fields, and these deployments continue to increase every day. These robots are different from each other, depending on the fields in which they are used and their tasks to be performed. These differences can be described by their shape, size and performance, among other traits. Some of them are statics and others are dynamics. One can meet these dynamic robots, especially in public areas such as at airports, in hotels, in hospitals and even in public transportation stations. Mobile robots are a kind of robot which helps human beings to be more efficient and productive in daily life activities.

In addition, the motion of these robots is a difficult task to perform, because they should avoid some obstacles along the road to their destination. Performing object avoidance when moving from one position to another is a complex and composite task for mobile robots, since this task involves scanning the surrounding environment, detection of obstacles, path planning and navigation to the desired destination and dock to achieve a specific task, such as auto-recharging their batteries when needed. Usually, many of these obstacles are static, but often some of them can be dynamic. In this case, the complexity level of a robot's navigation task is increased. As such, it is useful to make these robots more accurate. Knowing that a simple error of a mobile robot can lead to collisions and financial losses, the mobile robots need more space for free movement in their navigation times. Thus, it is essential to make the mobile robots operate appropriately to maximize space utilization and prevent accidents. Doing so will save financial loses to both the robot's developers (companies) and its users (customers).

Actually, to be more economic in the field of robotics, micro-electromechanical sensors (MEMSs) are a good solution to replace the existing expensive and huge sensors used in mobile robots. These MEMSs are embedded in almost all modern edge devices. In this context, the authors of [1–3] used acoustic signals to develop lightweight health monitoring AI systems. The developed technologies, based on edge devices with a bi-level optimization approach, can be used efficiently in on-board diagnostics (OBD) and smartphones. A basic platform for the design of a lightweight AI system is provided, which utilizes its built-in microphone for the health monitoring of agriculture machines. The adopted strategies considerably reduce the bulky data transmission on the Internet. Therefore, they provide very lightweight and economic artificial neural networks (ANNs), which are innovative frameworks and consist of a new roadmap to develop autonomous agriculture machines.

Additionally, several approaches are developed under different modeling assumptions to improve the robots' navigation information. For example, to name a few in the motion planning problem, one can refer to [4–6]. We also have the connectivity graphs, which are used to offer multipath possibility to the robots. Several studies are done to find the optimal shortest path among these multipaths for the robots so that an active simultaneous localization and mapping (SLAM) framework is developed in [7], which exploits a graph structure in order to improve the exploration time and accuracy. This framework is helped by an online algorithm based on least squares optimization for compensating the most common sources of errors, allowing the robot to reconstruct a more accurate graph. James et al. [8] also present four methods to adjust the connectivity of a networked system. To do so, a basic algorithm to track a desired connectivity profile through the addition and deletion of a sequence of single connections between two unmanned aerial vehicles (UAV) is developed.

The cell decomposition method consists of a kind of connectivity graph by dividing each dimension of the space into multiple parts. As the resulting path does not satisfy non-holonomic constraints, C. Zhang et al. [9] proposed trajectory planning and tracking for autonomous vehicles, based on a state lattice and model predictive control. To find feasible continuous plans, D. Zeng et al. [10] employed smooth cubic curvature polynomials in their investigation to ensure algorithm completeness and pick out the best trajectory, taking smoothness, comfort and economy into account. In the field of mobile robotics, navigation is an essential task classified into global navigation and local navigation. In global navigation, many methods have been developed such as those in [11,12]. To complete these, the authors of [13,14] discussed and developed some popular methods used in the local navigation class. Various researchers solved their navigation problems by successfully using the above two classes of navigation methods.

To further improve the accuracy of the motion information of robots, many filtering approaches exist and continue to be developed in the literature. Many applications are using the unscented Kalman filter (UKF) in various domains nowadays, ranging from target tracking [15] to multi-sensor fusion [16,17]. Another form of sensor fusion research to improve the performance of existing mobile robots is found in [18], where two methods (Dempster–Shafer theory and Kalman filtering) are used to integrate a global positioning system (GPS) and an inertial measurement unit (IMU), and the obtained results allowed for selecting the most accurate method for robot localization at an appropriate cost. In addition to completing the governing equations of the robot, the authors implemented a proportional–derivative controller to control and evaluate the kinematic and localization algorithms of the robot.

A similar work is [19], in which the encoder, compass, IMU and GPS measurements are used in combination with extended Kalman filter (EKF) to study and discuss the localization and navigation algorithms of the mobile robots. In this study, the proposed method contains three main approaches. In each of them, the method combines the robot controller with the measurements of the considered approach using sensor fusion, which combines the on-board sensor and GPS measurements through EKF. The three approaches were verified in a simulation, and the performance of the proposed algorithms was demonstrated when a fault in the encoder was considered. In the same field of research, two filtering approaches had been used by the authors of [20] to analyze the localization performance of SLAM (SLAM with a linear Kalman filter (KF) and SLAM with EKF). The simulation results of the proposed SLAM-based algorithms were evaluated and compared, and the results outperformed other algorithms regarding SLAM. In addition to presenting good accuracy, the proposed SLAM algorithms also gave a sensible computational complication.

Other examples of SLAM research can be found in [21], where an overview of the existing SLAM approaches is presented, with a focus on novel hybridized light detection and ranging (LiDAR) camera solutions. The authors first presented a short theory behind the SLAM process, concerning current, state-of-the-art LiDAR camera solutions. Then, they discussed visual SLAM with a monocular and stereo camera, as well as modern red green blue-depth (RGB-D) and event cameras. Therefore, all of the above research allows us to deepen our understanding regarding SLAM and its contributions to the artificial intelligence built in mobile robots. Three main contributions are done in this research paper:

- First, a new navigation algorithm, based on IR sensors for mobile robots, is created and named the novel IR navigation algorithm (NIRNA). This algorithm facilitates the robot's navigation to dock to the charger in the docking station.
- The second contribution consists of integrating NIRNA into an odometric system to build an Odom-NIRNA navigation system. This system greatly increases the quality of the classical odometer data.
- The navigation systems of the inertial navigation system (INS), Odom-NIRNA and the KF-based estimation system are combined to develop a new estimation approach, based on a hybridization technique named hierarchical infrared navigational algorithm hybridization (HIRNAH), to improve the accuracy of the current estimation systems for four-wheeled mobile robot (4-WMR) localization.

The build for HIRNAH is based on the principle of Kalman filters (KFs) for nonlinear systems, such as UKFs. This technique is a tight hybridization technique, which contains three hierarchical levels and thus provided a better robot state estimation. In the proposed system, each navigation system processes separately the robot state information and then, based on these results, the errors in the robot state are calculated. These state errors and the localization data from the RPLIDAR-A3 scanner (measurement unit) are used as inputs into the system filtering module (SFM) to produce the estimated errors of the robot state. Based on the obtained estimated errors, the robot's optimal state estimation is calculated, which is much more accurate than the robot's state estimation from some previous research.

The remainder of the paper is structured as follows: Section 2 describes the experimental configurations (parameters, setup and implementation) based on a real robot, while Section 3 is devoted to presenting the results and discussion of the experiments (statistical evaluation analysis and comparison of results). The future works then end this section. Section 4 describes the HIRNAH system proposed to improve the location of the robot in detail. Finally, the conclusion is presented in Section 5.

2. Experimental Configurations

To achieve the objective of this research, which consists of increasing the accuracy of the robot's localization using NIRNA and verifying the applicability of our approach, several tests were conducted on real experimentation in our laboratory.

2.1. Experiment Setups and Implementations

A test space (docking space) of a 3 × 3 m^2 flat floor was defined, which contained the robot's docking station and four obstacles (landmarks). In this docking station, the robot battery's charger was positioned at the middle of the upper side borderline of the test space. It broadcasted six separate IR signals from its three infrared transmitters (IRT) to guide the robot in its navigation (docking operation). These three IRTs were called the left IRT, central IRT and right IRT. They were positioned so that all of them were transmitting in different directions, and the angles separating the central IRT and the other two (left IRT and right IRT) were 35 degrees for each one. Finally, each IRT had a coverage angle of 30 degrees and defined its own covered area. Together, these covered areas defined the whole docking space. Figure 1 below illustrates the experiment docking space.

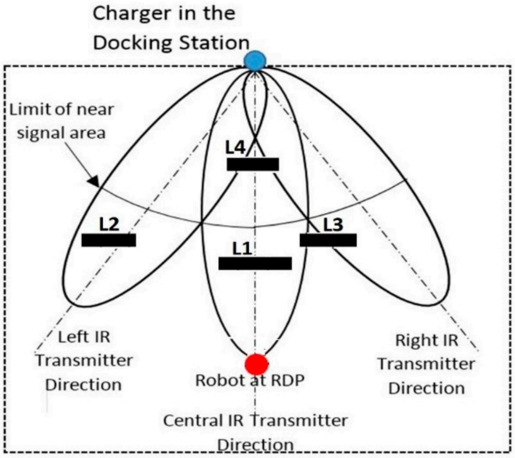

Figure 1. A four-wheeled mobile robot (4-WMR), charger and the four landmarks in the experiment docking space.

The role of these landmarks (L1, L2, L3 and L4) was twofold in this test space. First, they were used as references for the sensor measurement RPLIDAR-A3 scanner. Secondly, they interfered (to block) the broadcast IR signals from the charger to the robot. The robot departure position (RDP) was defined as the experiment starting position. From this RDP, the robot ran Algorithm 1 until finishing its docking operation. In addition to its infrared receiver (IRR), the robot was equipped with two encoders and one IMU module (smartphone-based sensor model) to provide Odometer and INS navigation data, respectively. These odometers provided wheel rotation rates, while the INS through the smartphone provided the acceleration force and the angular velocity to determine the robot's orientation.

Algorithm 1. Working principle of Odom-NIRNA

```
Input: IR signals, direction, $v$, $\omega$, $\theta$ for initial heading
Output: $P_{OdomN}(xR, yR, \theta R)$, CFlag Robot well docked to the charger
1: repeat
2: Call Algorithm 2
3: Calculate robot pose $POS_{Odom}(k+1)$
4: Update Robot pose to $P_{OdomN}(k)$ by using Equation (2)
5: Move forward at more 1 m
6:     if Robot not well connected then
7:             goto line 2
8:     else the Robot reaches the goal
9:         CFlag = True
8:         return $P_{OdomN}(xR, yR, \theta R)$, CFlag
9:     end if
10: until the Robot reaches the goal (End of Docking process)
```

The robot (4-WMR), using its IRR, followed the received IR signals to dock to the charger in the docking station. In addition, its RPLIDAR-A3 scanner was implemented to get the observation data of the robot. Below, Table 1 presents the main specifications of the 4-WMR used, and some of its experimentation steps are illustrated in the Figures 2–5.

Table 1. 4-WMR model parameters.

Symbol	Value	Quantity
d [cm]	40	Distance between the Two Back Wheels
L [cm]	45	Distance between the Wheels' Axles
r [cm]	12	Wheels Radius
N	2/1	Gear Ratio
ν [m s^{-1}]	[0.01; 0.05]	4-WMR Linear Velocity
ω [rad s^{-1}]	[0.1; 0.66]	4-WMR Angular Velocity

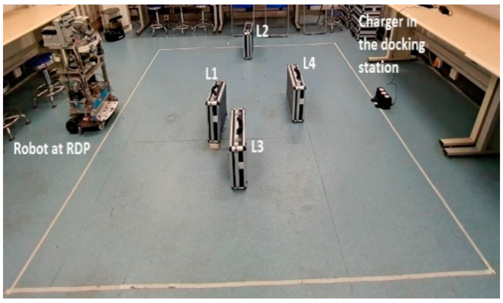

Figure 2. 4-WMR at the robot departure position (RDP).

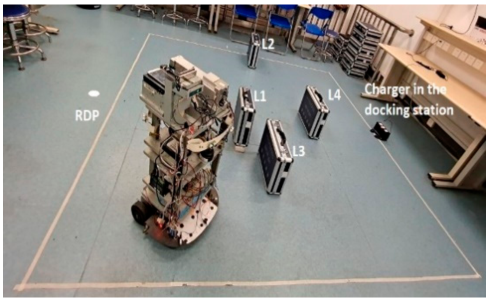

Figure 3. 4-WMR after turned 45° and moved forward 1.24 m.

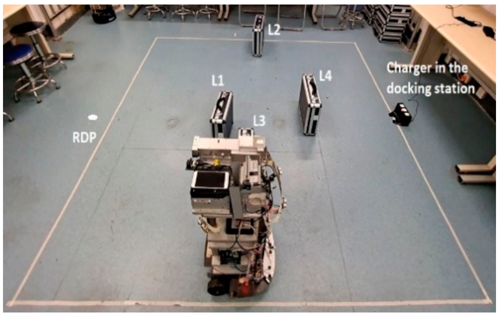

Figure 4. 4-WMR receiving only the right infrared transmitter (IRT) and traveling 2.08 m.

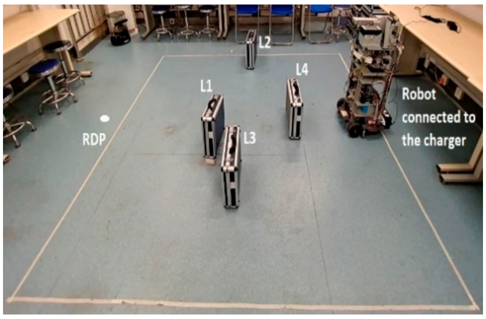

Figure 5. 4-WMR connected to the charger.

2.2. *Experiment Parameters and Performance Measurements*

The performance criterion was to determine the effect of NIRNA on the odometry localization approach used in this research. This was done by experimenting with our built system to identify the smallest pose errors of the robot. Recall that HIRNAH is a system based on an improved implementation of the classical UKF. This improvement came from the input data into the SFM, which in turn was based on the effect of NIRNA in the Odom-NIRNA navigation system. To realize this, the RDP was placed at 3.25 m from the charger on the main transmission line of the central IRT. In each experiment, the robot's linear speed v parameters (minimum and maximum) were set to be 0.01 m/s and 0.05 m/s respectively, and its angular velocities ω (minimum and maximum) were set to be 0.1 rad/s and 0.66 rad/s respectively, as indicated in Table 1 above.

To determine accurate measurements of the robot's final pose, pose errors were defined and used as measurement units in this experiment. Pose errors for each run were defined to be the absolute values of the differences between the actual pose and the calculated pose for each performance measurement (HIRNAH, hardware and control technology navigation (HCTNav), rapid exploring random tree (RRT) [22] and INS(IMU)), defined in Table 2 below. For each performance measurement, ten experiments were conducted.

Table 2. The definitions of the performance measurements.

Performance Measurements	Definitions
HIRNAH	Hierarchical Infrared Navigational Algorithm Hybridization (our proposed system)
HCTNav	Hardware and Control Technology Navigation
RRT	Rapid Exploring Random Tree
INS(IMU)	Inertial Navigational System (Inertial Measurement Unit)

3. Comparison Analysis and Statistical Evaluation of the Results

The produced errors of the position and orientation are presented in Figures 6 and 7, while the statistical analysis based on mean square error (MSE) are presented in Table 3 below.

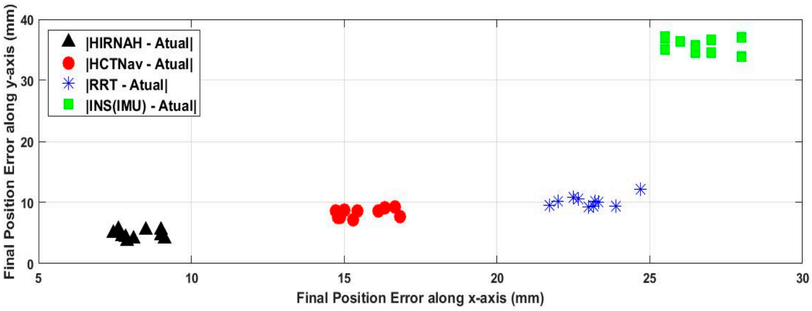

Figure 6. Robot's final position for ten runs.

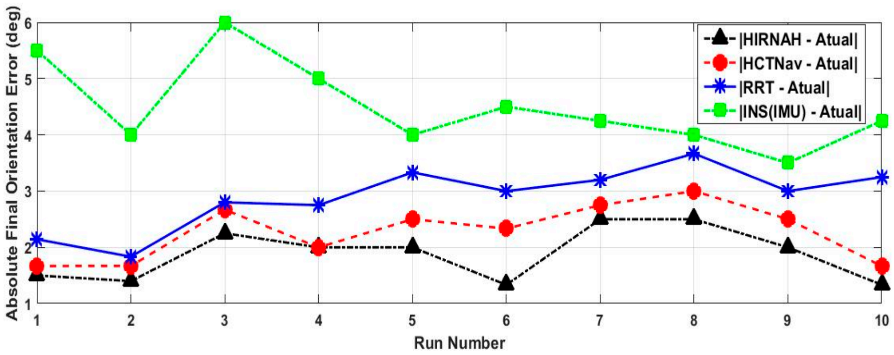

Figure 7. Robot's final orientation error.

Table 3. The mean square errors (MSEs) (mm) over 10 runs. Errors stated with respect to the true robot's state (position (x, y) and orientation (θ)).

MSE	Performance Measurements			
	HIRNAH	**HCTNav**	**RRT**	**INS(IMU)**
x-axis (mm)	0.36	0.60	0.69	0.82
y-axis (mm)	0.44	0.47	0.68	1.30
Heading (degree)	0.20	0.22	0.28	0.54

In these experimentation tests, the robot's travel path consisted of reaching the charger from the RDP by implementing successively the system based on NIRNA, HCTNav, RRT and INS (IMU) and then comparing the results. Implementing the system based on NIRNA, HCTNav or RRT consisted of successfully using as a navigation algorithm in the Odom-NIRNA module (see Figure 8) NIRNA, HCTNav or RRT. While implementing INS (IMU), the system was helped by camera data for navigation. The robot at the RDP facing the charger began to find the shortest path to the charger using the system.

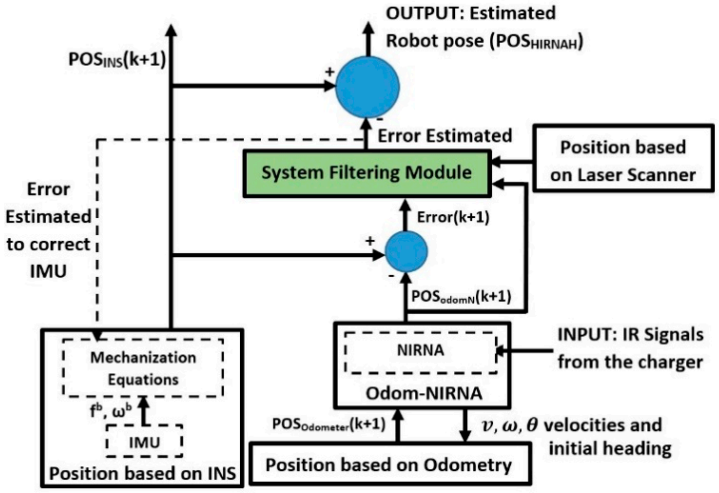

Figure 8. The block scheme of the hierarchical infrared navigational algorithm hybridization (HIRNAH) architecture.

From Figures 6 and 7 and Table 3, one can see that HIRNAH (the system when using NIRNA) provided more accurate positions and orientations, which was better than the system results when HCTNav, RRT or INS (IMU) were considered. In Figure 6, ten runs in each performance measurement of the robot's final poses are shown. In this figure, HIRNAH was the best with the lowest errors on average along the x-axis (8.22 mm) and along the y-axis (4.64 mm), followed successively by HCTNav and RRT. For HCTNav, the average errors along the x-axis and y-axis were 15.60 mm and

8.31 mm, respectively. That was slightly better than those of RRT, which were 23.02 mm and 10.20 mm, respectively. Finally, the worst-case performance measurement was given by INS(IMU), with the more erroneous information, on average, 26.55 mm errors along the x-axis and 35.8 mm errors along the y-axis. In Figure 7, one can notice that HIRNAH presents the best performance measurement (in terms of the robot's orientation errors for ten runs) with the lowest curve, followed successively by the curves for HCTNav, RRT and INS (IMU).

Throughout the ten experimental runs, HIRNAH provided less errors than the other performance measurements, except in run number four, where HIRNAH and HCTNav had the same errors. This is an exception which doesn't appear often; otherwise (if it appears several times) it means the above supremacy of HIRNAH can be reversed under some conditions. Finally, the worst-case orientation errors of the robot were provided when the system used INS (IMU) in standalone mode.

Moreover, a statistical analysis based on the mean square error (MSE) metric was also done to evaluate the performance of our proposed method. The MSE values for the different estimation methods used are summarized in Table 3. Recall that a low MSE implies high confidence for the localization and states estimation methods. From these results in Table 3, the proposed HIRNAH method presents the most accurate results when compared with the others (HCTNav, RRT and INS (IMU)) used in this research. The large values of the MSE for INS (IMU) were due to its accumulated drifts during a long period of operation in the computation of the state variables. When HCTNav is considered, low MSE values were provided, compared with those of RRT and INS (IMU). This describes the effectiveness of this navigation algorithm. For RRT, the random choice of the next node made it perform worse, with somewhat high MSE values. Finally, HIRNAH provided smaller (and therefore more precise) values in terms of MSE along the three parameters of the robot state variables, thanks to the low noise associated with the robot's pose when using NIRNA and given the history of measurements that can affect the accuracy of the robot's state. Therefore, the proposed HIRNAH method, which uses a filtering technique and NIRNA, can significantly reduce the MSE of the robot state.

4. Hierarchical Infrared Navigational Algorithm Hybridization (HIRNAH)

The HIRNAH system presented in this paper contains two navigation systems (one Odom-NIRNA and one INS) and an RPLIDAR-A3 scanner as an observation measurement unit (module). Below in Figure 8, the block scheme of the HIRNAH architecture is shown. These two navigation systems are combined to profit from their complementation. From the first navigational system, NIRNA and Odometer are handled together in order to produce the first navigational data, while in the second navigational system, an INS using an IMU provides the second navigation data. These navigational data are used to compute the robot state error, called error. This error, in addition to the Odom-NIRNA data (POS_{OdomN}) and the localization data from the RPLIDAR-A3 scanner (measurement unit), are used as inputs for the SFM to calculate the state estimated errors of the robot. These estimated errors in turn are used to compute the current optimal estimated state of the robot (POS_{HIRNAH}) and to correct the INS mechanization equations for the next loop. Independently, the system provides the INS navigation data ($POS_{INS(k+1)}$).

Let us consider a 4-WMR with the inertial reference frame $\{X_I, Y_I\}$ and robot body frame $\{X_R, Y_R\}$. In the Cartesian coordinate system and inertial frame, our robot's pose is expressed by $POS[x\ y\ \theta]^T$. The robot body frame $\{X_R, Y_R\}$ is selected so that x is forward and y is lateral, with the origin located at the robot's kinematics center. The inertial reference frame $\{X_I, Y_I\}$ is stationary and attached to the initial position of the robot. Thus, by applying a UKF as a localization algorithm for a 4-WMR, the state variables considered are for its pose $POS(x,\ y,\ \theta)$, which can be defined as

$$[POS_{UKF}] = \begin{bmatrix} x_{UKF}(k) \\ y_{UKF}(k) \\ \theta_{UKF}(k) \end{bmatrix} \tag{1}$$

In the rest of this paper, the state variable $POS_{UKF} = [X_{UKF}(k)\ Y_{UKF}(k)\ \theta_{UKF}(k)]^T$ will be represented by $POS = [x(k)\ y(k)\ \theta(k)]^T$ for simplicity reasons.

4.1. Position Based on Odom-NIRNA

4.1.1. Novel Infrared Navigational Algorithm (NIRNA)

The new docking strategy, named the novel infrared navigation algorithm (NIRNA), aims to navigate the autonomous robots to a specific place (here a docking station). Its controller's operating principle is summarized below in Algorithm 2. The novelty lies in the structure and the positioning of the IR sensors in both the charger and the charge controller on the robot. The final pose (x_R, y_R, θ_R) and connection status (CFlag) are the algorithm outputs, and a detailed description is given in [23]. In this paper, NIRNA is embedded into the odometer technology, and then the differential drive model [24] is adopted to estimate the kinematic model of the robot.

Algorithm 2. Controller of Novel Infrared Navigational Algorithm (NIRNA)

```
Input: 6 broadcast IR signals from the three IRTs of the charger
Output: $v$, $\omega$, direction//direction can be left, right, ahead and back
 1: while (1)
 2: looking for the 6 broadcast IR signals from charger
 3:     if Any signals Received then
 4:         Identify Signal sources and Signal level
 5:             if Obstacle in front of the Robot then
 6:                     Calculate Distance between Robot and the Obstacle
 7:                     Choose direction, $v$, $\omega$ to bypass the Obstacle
 8:                 Return direction = left or right or back, $\omega = 0.01, v = 0.01$
 9:         else//go ahead at more 1 m by following the received IR Signal
10:                 Return direction = ahead, $v = 0.03, \omega = 0$
11:         end if
12:     else goto line 5
13:     end if
14: end while
```

4.1.2. Odom-NIRNA Based Localization

For localization purposes, a mobile robot has to know its current location and orientation so that it can easily move from its current location to a destination. In the literature, there are many localization techniques with respect to mobile robots. Dead reckoning is one of them, which is more popularly used by scientific researchers. Thus, in this paper, we used the dead reckoning model, as applied in [23], as our research method for robot location. The selected velocities v_k, ω_k (linear and angular) and the orientation θ_k by NIRNA are used to calculate the output (position vector) of the Odom-NIRNA navigation system, denoted OdomN. Algorithm 1 above illustrates the working principle of Odom-NIRNA. The robot's Cartesian position is specified by the vector $P_{OdomN}(k) = [x_k,\ y_k]^T$, and its orientation is defined by $\theta_{OdomN}(k)$. Using the two coordinate frames (inertial reference frame $\{X_I, Y_I\}$ and robot body frame $\{X_R, Y_R\}$) during mobile robot motion, with respect to the inertial reference frame, the position vector $P_{OdomN}(k)$ is updated based on $P_{OdomN}(k-1)$, which is the position vector from the odometer $POS_{Odom}(k+1)$.

By adding the increments (travel distance (Δs) and change in angle ($\Delta\theta$)) of the robot from a known position (starting point), one can get its estimated pose $[x_{OdomN}(k),\ y_{OdomN}(k),\ \theta_{OdomN}(k)]^T$.

At time Ts, the estimated robot configuration is given by the following:

$$\begin{bmatrix} x_{OdomN}(k) \\ y_{OdomN}(k) \\ \theta_{OdomN}(k) \end{bmatrix} = \begin{bmatrix} x_{OdomN}(k-1) \\ y_{OdomN}(k-1) \\ \theta_{OdomN}(k-1) \end{bmatrix} + \begin{bmatrix} cos\left(\theta + \frac{\Delta\theta}{2}\right) & 0 \\ sin\left(\theta + \frac{\Delta\theta}{2}\right) & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix} \quad (2)$$

In the absence of wheel slippage and backlash, using the velocity data (linear and angular) of the robot, in addition to the odometric prediction above (commonly referred to as dead reckoning), good accuracy can be obtained on the location of the robot.

4.2. Position Based on INS (Inertial Measurement Units-Based Localization)

The INS (IMU) system can be used in two modes: standalone and cooperative. When used in standalone mode, it can be helped by a camera to navigate the robot to its charger. In cooperative mode, it will be used with the Odom-NIRNA system to determine robot state errors. The inertial measurement units (IMUs) play an important role in the INS to determine the position and orientation of the vehicles (robots here). This device is frequently used in robotics to help the robots in their navigation process. Usually, it contains many sensors such as accelerometers, gyroscopes and so on. The results (the linear acceleration and the velocity) produced by an accelerometer for mobile robots are affected by significant noise and accumulated drifts. The orientation obtained from a gyroscope contains some temporal drift and bias, which are the main source of gyroscope's errors. To overcome this unbounded error, several data fusion techniques have been developed [25–27]. Therefore, fusion (of both an accelerometer and gyroscope sensor) into a single device (IMU) is suitable to determine the pose of an object and to make up for the weakness of one over the other. In this paper, our used IMU is a 6-Degree of Freedom (DoF) accelerometer and gyroscope used to determine the pose estimation of our robot, as done in [23].

The measurements of the rotational rate from the gyroscope (in our used IMU) have to be integrated to yield the orientation. This is represented by

$$\dot{\theta}_{imu}(k) = \dot{\theta}^{a}_{imu}(k) + e_{imu}(k) + \eta_{imu}(k), \quad (3)$$

where $\dot{\theta}_{imu}(k)$ is the robot's heading rate based on the IMU's reading, $\dot{\theta}^{a}_{imu}(k)$ is the robot's actual heading rate, $e_{imu}(k)$ is the IMU bias drift error and $\eta_{imu}(k)$ is the associated white noise. The value of $\theta^{a}_{imu}(k)$ is obtained by integrating $\dot{\theta}^{a}_{imu}(k)$. Equation (4) below presents the robot's actual orientation based on the IMU reading:

$$\theta^{a}_{imu}(k+1) = \theta^{a}_{imu}(k) + \dot{\theta}^{a}_{imu}(k)Ts, \quad (4)$$

In the inertial reference frame, based on the readings of the accelerometer data from the IMU model used, the position of the robot $P_{imu}(k+1)$ is estimated by

$$P_{imu}(k+1) = P_{imu}(k) + \begin{bmatrix} cos(\theta^{a}_{imu}) & -sin(\theta^{a}_{imu}) & 0 \\ sin(\theta^{a}_{imu}) & cos(\theta^{a}_{imu}) & 0 \\ 0 & 0 & 1 \end{bmatrix} vr^{b}(k), \quad (5)$$

where the robot velocity in the robot body frame is $vr^{b}(k) = \left[v^{b}_{x}(k),\ \dot{\theta}_{imu}(k)\right]$, with $v^{b}_{x}(k)$ as the robot's linear velocity and $\dot{\theta}_{imu}(k)$ as its angular velocity.

In the Cartesian coordinate system, Equation (5) is rewritten to

$$\begin{bmatrix} x_{imu}(k+1) \\ y_{imu}(k+1) \\ \theta_{imu}(k+1) \end{bmatrix} = \begin{bmatrix} x_{imu}(k) \\ y_{imu}(k) \\ \theta_{imu}(k) \end{bmatrix} + \begin{bmatrix} cos(\theta^{a}_{imu}) & -sin(\theta^{a}_{imu}) & 0 \\ sin(\theta^{a}_{imu}) & cos(\theta^{a}_{imu}) & 0 \\ 0 & 0 & 1 \end{bmatrix} vr^{b}(k), \quad (6)$$

4.3. Position Based on RPLIDAR-A3 Scanner

The sensor measurements considered in this section were provided by an RPLIDAR-A3 scanner, similar to the approaches adopted in [28–30]. It was positioned on the robot at height h_L (46 cm) above the ground and with the tilt control fixated at $\theta_L = 9°$. Recall that, in the Cartesian coordinate system and in the inertial frame, our robot's position is expressed by $\text{POS}[x\ y\ \theta]^{\text{T}}$. N landmarks at known positions $\left(x^i_L,\ y^i_L\right)$, $i = 1, \ldots, N$ are considered in the robot's navigation space. We assumed to be zero, for simplicity, the uncertainties associated with landmark locations. During the robot's motion, at each time step, the distance d and the relative angle ϕ to one or more landmarks were observed. That means, with RPLIDAR-A3 scan readings given in the plane of the RPLIDAR-A3 scanner, each RPLIDAR-A3 scan $P^L(d_i,\ \phi_i)$ comprises a set of distance readings d_i, $i \in [1, N]$ and the angles $\phi_i \in [-90°, 90°]$ associated with these. The observation model provides a mechanism for computing the expected values of observations from the sensors, given the knowledge of the robot's navigation space (location of the charger and all the landmarks) and an estimate of the robot's location. In the frame of the RPLIDAR-A3 scanner, the scans P^L are represented in polar coordinates. Through the sensors mounted on the robot, at each time step $k + 1$, both the distance and the relative angle to the landmarks were observed, and the observation model is given by

$$\begin{aligned} d^i_{k+1} &= \sqrt{\left(\left(x^i_L - x_{k+1}\right)^2 + \left(y^i_L - y_{k+1}\right)^2\right)} + \omega_r, \\ \phi^i_{k+1} &= \phi^i_R = atan\left(\frac{y^i_L - y_{k+1}}{x^i_L - x_{k+1}}\right) - \theta_{k+1} + \omega_\phi, \end{aligned} \tag{7}$$

where ω_r and ω_ϕ are zero-mean Gaussian observation noises and $\theta_{k+1} = \theta_{OdomN}(k)$.

To get the Cartesian coordinates of a measurement in the robot fixed frame, the following mapping transformation $\mathcal{M}_{RL}(h_L,\ \theta_L) : P^L(d_i,\ \phi_i) \rightarrow P^i_R(x^i_R,\ y^i_R,\ z^i_R)$, as shown in Equation (8), can be applied, where P^i_R denotes the robot position related to the landmark i using the measurement model of an RPLIDAR-A3 scanner:

$$P^i_R = \begin{bmatrix} x^i_R \\ y^i_R \\ z^i_R \end{bmatrix} = \begin{bmatrix} d_i cos\theta_L cos\phi_i \\ d_i sin\phi_i \\ h_L - d_i sin\theta_L cos\phi_i \end{bmatrix}, \tag{8}$$

As our robot's navigation space is a flat environment and its body frame in consideration is a two-dimensional frame $\{X_R, Y_R\}$, the z_i coordinate was set to be zero and replaced by ϕ^i_R, the related angle between the robot and the landmark i for analysis simplicity. In the same body frame, the RPLIDAR-A3 scan measurement angle ϕ^i_R was zero forward and positive to the left. In this context, Equation (8) can be rewritten and represented below by Equation (9):

$$P^i_R = \begin{bmatrix} x^i_R \\ y^i_R \\ \phi^i_R \end{bmatrix} = \begin{bmatrix} d_i cos\theta_L cos\phi_i \\ d_i sin\phi_i \\ atan\left(\frac{y^i_L - y_{k+1}}{x^i_L - x_{k+1}}\right) - \theta_{k+1} + \omega_\phi \end{bmatrix}, \tag{9}$$

4.4. System Filtering Module

In order to estimate the errors of the robot state, the following standard discrete time equations can be used to represent the system, with the system model represented abstractly as f and the measurement model represented abstractly as h (be they linear or non-linear):

$$\begin{aligned} E_k &= f\left(E_{k-1},\ q_{k-1},\ u_{1,\ k-1}\right), \\ Z_k &= h\left(E_k,\ n_k,\ u_{2,\ k}\right), \end{aligned} \tag{10}$$

where E is the system state, q is the process noise, n is the observation noise, u_1 is the exogenous input to the state transition function, u_2 is the exogenous input to the state observation function and Z is the noisy observation of the system.

This SFM aims to smooth the location of the robot. To do so, the considered state variable of the robot is the error term (E) between Equations (2) and (6), which is found by subtracting P_{OdomN} from P_{imu}.

The adopted process model is presented below by Equation (11), which is a recursive equation deducted from Equations (2) and (6):

$$\begin{aligned} E(k+1) &= AE(k) + M(k)\begin{bmatrix} v_x^b(k) \\ v_y^b(k) \\ z(k) \end{bmatrix}, \\ &= f\Big\{E(k), vr^b(k), \theta^a_{imu}(k),\ \theta_{OdomN}(k)\Big\}, \end{aligned} \tag{11}$$

where A is a 3×3 identity matrix, $z(k) = \dot{\theta}^a_{imu}(k) - \dot{\theta}_{OdomN}(k)$ and $M(k)$ is a 3×3 rotational cosine matrix related to the control unit of the system and defined by

$$\begin{aligned} M(k) &= \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ a &= \ cos\theta^a_{imu}(k) - cos\theta_{OdomN}(k) \\ b &= -sin\theta^a_{imu}(k) + sin\theta_{OdomN}(k) \\ c &= \ sin\theta^a_{imu}(k) - sin\theta_{OdomN}(k) \\ d &= \ cos\theta^a_{imu}(k) - cos\theta_{OdomN}(k), \end{aligned} \tag{12}$$

The state variable of our system is defined by Equation (13):

$$[E_{UKF}] = \begin{bmatrix} x_{imu}(k) - x_{OdomN}(k) \\ y_{imu}(k) - y_{OdomN}(k) \\ \theta_{imu}(k) - \theta_{OdomN}(k) \end{bmatrix} = \begin{bmatrix} e_x(k) \\ e_y(k) \\ e_\theta(k) \end{bmatrix}, \tag{13}$$

The first step of the UKF implementation is the state vector augmentation. Thus, the n-dimension state vector E of the system needs to be restructured and augmented with q-term process noise. That is presented below in Equation (14):

$$E^a_{k-1} = \Big[E_{k-1},\ \ q_{k-1}\Big]^T, \tag{14}$$

where q_{k-1} is the augmented part and the dimension of the augmented state vector is $n^a = n + q$. The process model can be rewritten as a function of E^a_{k-1} to calculate the a priori state estimate:

$$\hat{E}_k = f\Big(E^a_{k-1}\Big) = E(k-1) + \eta(k-1) = \begin{bmatrix} e_x(k-1) \\ e_y(k-1) \\ e_\theta(k-1) \end{bmatrix} + \begin{bmatrix} \eta_x(k-1) \\ \eta_y(k-1) \\ \eta_\theta(k-1) \end{bmatrix}, \tag{15}$$

where the modeled part of the predefined data differences is represented by $E(k-1)$ while the augmented part is $\eta(k-1)$, a zero mean white noise.

Here, we only consider the system input error while neglecting the system model error.

Thus, the augmented a priori state estimate and its covariance matrix are restructured as

$$E^a_k = \begin{bmatrix} \hat{E}_k \\ 0_{3\times 1} \end{bmatrix},\ P^a_k = \begin{bmatrix} \hat{P}_k & 0 \\ 0 & Q_k \end{bmatrix}, \tag{16}$$

where $\hat{P}_k$, and Q_k are the covariance matrices of the state variable E and the process noise q, respectively.

The observation model used here, called the output function or errors estimated, is represented below by Equation (17):

$$\begin{aligned}\hat{z}(k+1) &= \hat{E}(k+1) = h\Big\{P_{OdomN}(k),\ P^i_R(k), E(k)\Big\}\\ &= \begin{bmatrix} x_{OdomN}(k) - x^i_R(k) \\ y_{OdomN}(k) - y^i_R(k) \\ \theta_{OdomN}(k) - \phi^i_R(k) \end{bmatrix} / \begin{bmatrix} e_x(k-1) \\ e_y(k-1) \\ e_\theta(k-1) \end{bmatrix} = \begin{bmatrix} e^x_{est}(k) \\ e^y_{est}(k) \\ e^\theta_{est}(k) \end{bmatrix},\end{aligned} \tag{17}$$

The procedure and implementation of the UKF algorithm adopted here is same as that given in [20]. By merging the INS's belief (P_{imu}) with the probability of making exactly that observation $\hat{E}_k$, the robot corrects its posture. That is $P_{HIRNAH}(k+1)$, represented in Equation (18) by $P_{Robot}(k+1)$.

$$P_{Robot}(k+1) = P_{imu}(k+1) - \hat{E}_k, \tag{18}$$

where $P_{Robot}(k)$ is the corrected pose using HIRNAH along the corresponding axis.

5. Conclusions

A new HIRNAH system for mobile robot state estimation and localization has been constructed in this research paper. Based on sensor fusion through a tight hybridization technique, the built system contains three hierarchical levels. Two navigation systems (Odometer and INS) and a sensor measurement module (an RPLIDAR-A3 scanner) cooperated to achieve this HIRNAH system. The information from the two navigation systems (INS(IMU) and Odom-NIRNA) are used to estimate the robot's state errors. These errors are entered into the SFM with the sensor measurement (RPLIDAR-A3 scanner) data to produce estimated errors and smooth the robot pose provided by the INS(IMU) system in order to produce the robot's final pose of the entire system. The Odom-NIRNA system is built based on integrating a new navigation algorithm NIRNA and odometry to improve the classical odometry navigation data.

In this research, simulations were conducted in order to validate the applicability of the proposed system. Based on the results from these simulations, a real system was built and used to experiment on a real robot in our laboratory. The experiment results show that HIRNAH outperforms all the performance measurements used in this research, such as HCTNav, RRT and INS(IMU). This means that the odometry integrated with NIRNA can be used to provide a more accurate estimation of the location information (position and orientation) for a 4-WMR.

In our future work, we plan to improve the proposed method by taking into account another scenario, including more landmarks and some dynamic objects. In addition, as we only tested the proposed method on a robot using a single IRR, there is further need to extend the number of IRRs to three (left IRR, central IRR and right IRR) and perform more evaluations of our built HIRNAH system. Another extension possibility will be to increase the number of runs in the experiments to at least a hundred times, and perhaps with other filtering techniques.

Author Contributions: Conceptualization, M.D.; data curation, M.D.; formal analysis, M.D.; funding acquisition, X.C.; investigation, M.D. and X.C.; methodology, M.D.; project administration, M.D.; resources, M.D. and X.C.; software, M.D.; supervision, X.C.; validation, M.D. and X.C.; visualization, X.C.; writing—original draft, M.D.; writing—review and editing, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Natural Science Foundation of China NSFC, grant number 61772185, and the APC was funded by the NSFC.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gupta, N.; Khosravy, M.; Patel, N.; Dey, N.; Gupta, S.; Darbari, H.; Crespo, R.G. Economic data analytic AI technique on IoT edge devices for health monitoring of agriculture machines. *Appl. Intell.* **2020**, *50*, 3990–4016. [CrossRef]
2. Gupta, N.; Khosravy, M.; Gupta, S.; Dey, N.; Crespo, R.G. Lightweight artificial intelligence technology for health diagnosis of agriculture vehicles. *Int. J. Parallel Program.* **2020**, 1–22. [CrossRef]
3. Gupta, N.; Gupta, S.; Khosravy, M.; Dey, N.; Joshi, N.; Crespo, R.G.; Patel, N. Economic IoT strategy: The future technology for health monitoring and diagnostic of agriculture vehicles. *J. Intell. Manuf.* **2020**, 1–12. [CrossRef]
4. Al–Jarrah, R.; Shahzad, A.; Roth, H. Path planning and motion coordination for multi-robots system using probabilistic neuro–fuzzy. *IFAC Pap. Online* **2015**, *48*, 6–51. [CrossRef]
5. Chi, W.; Wang, J.; Meng, M.Q. Risk-Informed-RRT*: A sampling-based human-friendly motion planning algorithm for mobile service robots in indoor environments. In Proceedings of the IEEE International Conference on Information and Automation (ICIA), Wuyishan, China, 11–13 August 2018; pp. 1101–1106.
6. Hossain, M.A.; Ferdousand, I. Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique. *Robot. Auton. Syst.* **2015**, *64*, 137–141. [CrossRef]
7. Soragna, A.; Baldini, M.; Joho, D.; Kümmerle, R.; Grisetti, G. Active SLAM using connectivity graphs as priors. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 340–346. [CrossRef]
8. Trimble, J.; Pack, D.; Ruble, Z. Connectivity tracking methods for a network of unmanned aerial vehicles. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 440–447. [CrossRef]
9. Zhang, C.; Chu, D.; Liu, S.; Deng, Z.; Wu, C.; Su, X. Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 29–40. [CrossRef]
10. Zeng, D.; Yu, Z.; Xiong, L.; Fu, Z.; Zhang, P.; Zhou, H. $ DBO $ trajectory planning and $ HAHP $ decision-making for autonomous vehicle driving on urban environment. *IEEE Access* **2019**, *7*, 165365–165386. [CrossRef]
11. Gao, K.; Xin, J.; Cheng, H.; Liu, D.; Li, J. Multi-mobile robot autonomous navigation system for intelligent logistics. In Proceedings of the Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 2603–2609. [CrossRef]
12. Almeida, H.P.; Júnior, C.L.N.; Santos, D.D.S.; Leles, M.C.R. Autonomous navigation of a small-scale ground vehicle using low-cost IMU/GPS integration for outdoor applications. In Proceedings of the IEEE International Systems Conference (SysCon), Orlando, FL, USA, 8–11 April 2019; pp. 1–8. [CrossRef]
13. Kanayama, H.; Ueda, T.; Ito, H.; Yamamoto, K. Two-mode mapless visual navigation of indoor autonomous mobile robot using deep convolutional neural network. In Proceedings of the IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 536–541. [CrossRef]
14. Li, Z.; Xiong, Y.; Zhou, L. ROS-based indoor autonomous exploration and navigation wheelchair. In Proceedings of the 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; pp. 132–135. [CrossRef]
15. Li, J.M.; Chen, C.W.; Cheng, T.H. Estimation and tracking of a moving target by unmanned aerial vehicles. In Proceedings of the American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 3944–3949. [CrossRef]
16. Magrin, C.E.; Todt, E. Multi-sensor fusion method based on artificial neural network for mobile robot self-localization. In Proceedings of the Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE), Rio Grande, Brazil, 23–25 October 2019; pp. 138–143. [CrossRef]
17. Ruan, X.; Liu, S.; Ren, D.; Zhu, X. Accurate 2D localization for mobile robot by multi-sensor fusion. In Proceedings of the IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; pp. 839–843. [CrossRef]
18. Erfani, S.; Jafari, A.; Hajiahmad, A. Comparison of two data fusion methods for localization of wheeled mobile robot in farm conditions. *Artif. Intell. Agric.* **2019**, *1*, 48–55. [CrossRef]

19. Al Khatib, E.I.; Jaradat, M.A.; Abdel-Hafez, M.; Roigari, M. Multiple sensor fusion for mobile robot localization and navigation using the extended Kalman filter. In Proceedings of the 10th International Symposium on Mechatronics and its Applications (ISMA), Sharjah, UAE, 8–10 December 2015; pp. 1–5. [CrossRef]
20. Ullah, I.; Su, X.; Zhang, X.; Choi, D. Simultaneous localization and mapping based on Kalman filter and extended Kalman filter. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 2138643. [CrossRef]
21. Debeunne, C.; Vivet, D. A review of visual-LiDAR fusion based simultaneous localization and mapping. *Sensors* **2020**, *20*, 2068. [CrossRef] [PubMed]
22. Varghese, A.M.; Jisha, V.R. Motion planning and control of an autonomous mobile robot. In Proceedings of the International CET Conference on Control, Communication, and Computing 0, Thiruvananthapuram, India, 5–7 July 2018.
23. Doumbia, M.; Cheng, X.; Chen, L. A novel infrared navigational algorithm for autonomous robots. In Proceedings of the IEEE International Conference on Artificial Intelligence and Information Systems, Dalian, China, 20–22 March 2020.
24. Parween, R.; Heredia, M.V.; Rayguru, M.M.; Abdulkader, R.E.; Elara, M.R. Autonomous self-reconfigurable floor cleaning robot. *IEEE Access* **2020**, *8*, 114433–114442. [CrossRef]
25. Shahian Jahromi, B.; Tulabandhula, T.; Cetin, S. Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors* **2019**, *19*, 4357. [CrossRef] [PubMed]
26. De Silva, V.; Roche, J.; Kondoz, A. Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots. *Sensors* **2018**, *18*, 2730. [CrossRef] [PubMed]
27. Nada, D.; Bousbia-Salah, M.; Bettayeb, M. Multi-sensor data fusion for wheelchair position estimation with unscented Kalman Filter. *Int. J. Autom. Comput.* **2018**, *15*, 207–217. [CrossRef]
28. Li, K.; Xu, Y.; Wang, J.; Meng, M.Q.H. SARL: Deep reinforcement learning based human-aware navigation for mobile robot in indoor environments. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 688–694. [CrossRef]
29. Surmann, H.; Jestel, C.; Marchel, R.; Musberg, F.; Elhadj, H.; Ardani, M. Deep Reinforcement Learning for Real Autonomous Mobile Robot Navigation in Indoor Environments. Available online: https://arxiv.org/abs/2005.13857 (accessed on 14 October 2020).
30. Amjad, H.; Sultan, M.; Khan, H.R. Low cost 2D RPLIDAR scanner based indoor mapping and classification system. In Proceedings of the 2019 International Conference on Robotics and Automation in Industry (ICRAI), Rawalpindi, Pakistan, 21–22 October 2019; pp. 1–6. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.