Multiple Internet of Robotic Things robots based on LiDAR and camera sensors

Yanyan Dai and Suk Gyu Lee

Abstract
A combination of Internet of Things and multiple robots with sensors has been an attractive research topic over the past years. This article proposes an Internet of Robotic Things system structure to monitor events, fuse sensor data, use local robots to determine a best action, and then act to control multiple mobile robots. The Internet of Robotic Things system includes two main layers: the host controller layer and the multiple robots layer. The controller layer communicates with the multiple robots layer by Wi-Fi module. The Internet of Robotic Things system helps finish five tasks: localizing robots, planning paths, avoiding obstacles, moving to waypoint stable, and creating a map. Based on depth data from depth camera and robot posture, a mapping algorithm is proposed to create map. Based on light detection and ranging sensor data and google cartographer, simultaneously localization and mapping (SLAM) is also processed in this article. The fuzzy sliding mode tracking control method is proposed for each robot to guarantee the robot stable moves. Simulation results show the effectiveness of the proposed algorithm and are used to compare with the experiment result. In the experiment, one host computer and two Kobuki mobile robots with light detection and ranging and depth camera sensors are integrated as an Internet of Robotic Things system. Two robots successfully localize themselves and avoid obstacles. The follower robot simultaneously builds a map.

Keywords
Multiple IoRT robots, intelligent sensing, data sharing, obstacle avoidance, SLAM, object detection, fuzzy sliding mode tracking control method

Introduction
Multiple robot systems have been the popular research topic since it can accomplish complex tasks more efficiently than a single robot.1 Multiple robots may also produce more accurate maps by using more information than a single robot as described in the literature.2–5 In the literature,3 Alonso-Mora et al. showed teams of robots avoiding obstacles but did not show robot motion control. In the literature,4 Ravankar et al. considered multi-robot obstacle avoidance, but it needs high computation time. In the literature,5 Papoutsidakis and Piromalis discussed multiple robots work using low-cost robots, but it does not include robot’s mapping topic. In the literature,5 Ravankar et al. showed multiple robots path planning, obstacle avoidance, and mapping tasks, but all these tasks rely on a leader.
robot. If the leader robot fails, the whole system will fail. In our article, we show multiple robots finish five tasks: localizing themselves, planning paths, avoiding obstacles, moving to waypoint stable, and creating a map. In the multi-robot systems research, multi-robot formation control system is recognized as a popular research topic, since the system not only carries on the advantages of multi-robot systems but also has its own advantages. For example, it ensures no team members get lost and enlarges communication bandwidths. The formation control approaches include leader–follower approach, virtual structure approach, and behavior-based approach. Due to easy implementation, the leader–follower formation approach is applied in this article, where the follower robot keeps tracking the leader robot by the desired distance and bearing angle.

Internet of things (IoT) is developing quickly and being used in many areas, such as agriculture, healthcare, smart homes, and transportation. To optimize the system, adding dynamic actuators into the IoT applications is needed. Therefore, a combination of IoT and robotics as Internet of Robotic Things (IoRT) is a well-matching solution. IoRT was first introduced by ABI Research. ABI Research defines IoRT as that: intelligent devices can monitor events, fuse sensor data, use local and distributed intelligence to determine a best action, and then act to control or manipulate objects, and sometimes while moving through that world. IoRT focuses not only on supporting information services for pervasive sensing, tracking, and monitoring but also on producing action, interaction, and autonomous behavior. Based on the literature, sensor configuration is the foundation for driving intelligent vehicles. A decision-making program is necessary while configuring sensors. To obtain information on environment, fusion of vision and light detection and ranging (LiDAR) data is widely used. The IoRT system is applied by some companies. For example, FANUC joined forces with Rockwell Automation, Preferred Networks, and Cisco for the development of a system called “FIELD” (FANUC Intelligent Edge Link and Drive). In the Amazon Robotics warehouse automation fulfillment center, mobile robots move bins and pallets and can coordinate their movements to avoid accidents. Google’s vehicle applied robotics to a modified vehicle that has an array of additional sensors allowing the car to make its own decision based on the information as it drives. In the literature, a RoboEarth Cloud Engine, which is an open-source Platform-as-a-Service framework, is designed and implemented for robotics applications. However, the differences between Web and robotics applications make it hard to use existing Web solutions for robotics. Such differences include programming languages, the number of processes, and the communication protocols. Based on the literature, a variety of skills to design and implement an IoRT system are necessary, such as:

1. Instrument skill: the IoRT includes a sensor fusion concept. Sensor data are used to control and manipulate objects in the real world.
2. Design control system skill: it includes creating modeling and designing a stable control system.
3. Choose suitable controller skill: there are some microcontrollers, such as Raspberry Pi, Arduino, NI myRIO, and so on. Based on the system requirement and budget, choose a suitable controller.
4. Program skill: it includes choosing a rapid control software and program to realize the application.
5. Build network skill: it includes knowing about the different network topologies and how to set them up, such as transmission control protocol/internet protocol (TCP/IP) and user datagram protocol (UDP).
6. Know IoT application protocol skill: there are hyper text transfer protocol (HTTP), Web Socket, Message Queuing Telemetry Transport (MQTT), Advanced Message Queueing Protocol, and Data Distribution Service (DDS). DDS and MQTT provide real-time behavior to guarantee the IoRT application a real-time performance.

Although several pieces of article discussed IoRT topics, fewer of them do real experiments and show how to analyze sensor data to finish tasks. In this article, we show how to use the above-mentioned skills to build an IoRT system to monitor events, fuse sensor data, use local robots to determine a best action, and then act to control multiple mobile robots. IoRT systems help to finish five tasks: localizing robots, planning paths, avoiding obstacles, moving to waypoint stable, and creating a map, using Kobuki robots. The encoder and Inertial Measurement Unity (IMU) are used for robot localization. The leader robot’s posture is shared with the follower robots to realize the leader–follower formation control approach. LiDAR sensors data are shared with each other to avoid obstacles or create a map. The ORBBEC Astra depth camera is used to detect an object and create a map. The NI myRIO controller with Wi-Fi module or NVIDIA Jetson TX2 controller is used to control robots and communicate with human machine interface (HMI). TCP/IP-based network is setup, and DDS IoT application protocol is applied, due to its advantages like reliability and speed while comparing with MQTT.

For robot robust motion control, there are some methods, such as extended kalman filter (EKF), optimal feedback control, robust control, intelligent control, decentralized control, and combining physics- and maneuver-based prediction method. For the combining physics- and maneuver-based prediction method, the uncertainty is considered in the nonlinear dynamic prediction model using the unscented transform. EKF requires a long time to update the desired states. Intelligent control requires high computation power. In robust control, a priori information about the bounds of the uncertain or time-varying parameters is necessary. About the drawback of decentralized control it’s difficult to coordinate the
overall system. The sliding mode control method is recognized as one of the most efficient tools, with advantages such as fast response and robustness with respect to system uncertainties and external disturbances. In this article, the sliding mode tracking control method is used to stable control each robot with less computation time. To weaken the charting, the fuzzy sliding mode controller is designed.

The contributions of this article are: (a) we proposed an IoT system structure, to monitor events, fuse sensor data, use local robots to determine a best action, and then act to control multiple mobile robots. IoT system helps to finish five tasks: localizing robots, planning paths, avoiding obstacles, moving to waypoint stable, and creating a map, using low-cost robots; (b) a mapping method is discussed to create a map using depth camera’s data and robot posture; (c) a fuzzy sliding mode controller is proposed for robust tracking control for the nonholonomic mobile robots; and (d) the experiment results show the effectiveness of such IoT system in real environment.

The article is organized as follows. The second section gives two modelings. The third section describes the structure of IoT system and shows how to build a map, based on the depth camera data. The fourth section outlines the fuzzy sliding mode tracking control method. The fifth section shows the simulation result. Finally, the sixth section presents the experimental results.

**Modeling**

**Robot modeling**

The mobile robot $R_i$ is under consideration, the motion of each robot is described in terms of $P_i = [x_i, y_i, \theta_i]^T$, where $x_i, y_i$ and $\theta_i$ are $x$ coordinate, $y$ coordinate, and bearing angle, respectively. The trajectory of each robot has the form of $q_i = [x_i, y_i]$, where the input of each robot is linear velocity $v_i$ and angular velocity $\omega_i$. The model of robot has the form of

$$
\dot{P}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}.
$$

**Leader–follower formation modeling**

As the leader–follower formation modeling in the literature, one robot is designed as the leader robot and the others are designed as the follower robots. The leader robot moves along a trajectory, avoiding obstacles, while the follower robots keep track of the leader robot and maintain the desired distance and desired bearing angle with the leader robot.

For $n$ robots, robot $R_1$ is assigned to be the leader robot, which determines each follower robot’s motion. Let $R_1$ and $R_i$ ($i = 2, 3, \ldots, n$) be the leader robot and the follower robot, respectively. $d_{iw}$ is the waypoint posture of the follower robot. $l_{i1}$ is defined as the desired distance between $R_1$ and $R_{iw}$. $\beta_{i1}$ is the desired bearing angle from the orientation of the follower robot to the axis connecting $R_1$ and $R_{iw}$. The waypoint posture $d_{iw} = [x_{iw}, y_{iw}, \theta_{iw}]^T$ of follower robot $R_i$ is denoted as (2) and (3).

$$
\begin{align*}
\dot{x}_{iw} &= x_1(t) - l_{i1} \cos(\beta_{i1} + \theta_1(t)) \\
\dot{y}_{iw} &= y_1(t) - l_{i1} \sin(\beta_{i1} + \theta_1(t)) \\
\theta_{iw}(t) &= \theta_1(t)
\end{align*}
$$

**IoRT system structure**

The IoRT system structure is shown in Figure 1. It includes two main layers: the host controller layer and the multiple robots layer. Human controls and monitors the host controller. The host controller collects each robot’s sensor data and posture, analyze sensor data, display sensor data and each robot’s posture, and share necessary data to each robot. The main responsibility of host controller is to monitor and display all data for human. After analyzing LiDAR sensor data, if there are obstacles, the switching formation strategy for obstacle avoidance is applied. The desired distance and bearing angle are assigned to the follower robot. In Figure 1, the dash line is an emergency line for the system. If the host controller fails to work, the leader robot will communicate with the follower robot directly to share its posture and LiDAR sensors data. About multiple robots’ layer, multiple robots move based on leader–follower formation approach. Initially, all robots are given start points.

The leader robot’s control scenario is as follows:

1. detects environment by the equipped LiDAR sensor;
2. avoids obstacle or moves forward, depending on the LiDAR sensor’s data; if the leader robot uses the NVIDIA Jetson TX2 controller, it will process cartographer SLAM, based on LiDAR sensor data;
3. sends a command to its motors to move;
4. localizes itself using encoder and IMU; and
5. shares its posture and LiDAR sensor data to the host controller by wireless module.

The follower robot’s control scenario is as follows:

1. obtains shared LiDAR sensor data from the leader robot;
2. switches formation to avoid an obstacle or maintains predefined formation;
3. sends a command to its motors to move;
4. localizes itself using encoder and IMU;
5. shares follower robot posture by wireless module; and
6. creates a map by a depth camera and shares camera sensor data to host by wireless module.

Although the follower robot is equipped with a depth camera, the horizontal field of view of the depth camera is

It is difficult to get 360° environment information simultaneously. To plan safe path, the follower robot needs 360-degree environment information simultaneously. Therefore, the follower robot uses shared LiDAR sensor data from the leader robot to avoid obstacles.

The robot uses an ORBBEC Astra depth camera to build a 2D environment map. The horizontal field of view of the Astra camera is 60 degree. Choosing depth mode for Astra camera, it will show 640 × 480 depth image. The value of each pixel presents depth data, which is the distance from the camera to the object in millimeter. We use the central row (row 320) of the depth image data to generate a 2D map. As shown in Figure 2, the point $H(x_h, y_h)$ in the map is calculated, based on (4) and (5). $x_h$ is equal to the depth data $d$. $H$ is presented by the $j$ pixel ($j = 1, 2, \ldots, 640$) of the central row. The parameter $\beta$ is calculated as

$$\beta = \frac{(j - 320) \times 60}{640} \quad (4)$$

Therefore, $y_h$ is calculated as

$$y_h = d \tan \beta \quad (5)$$

In Figure 2, $\{X_g, Y_g\}$ is present as the global reference frame. $\{X_l, Y_l\}$ is present as a local reference frame on the robot chassis. Rotation matrix $R$ expresses the orientation of the local reference frame with respect to the global reference frame, where $\theta_l$ is the robot orientation in the global reference frame.

$$R(\theta_l) = \begin{bmatrix} \cos \theta_l & \sin \theta_l & 0 \\ -\sin \theta_l & \cos \theta_l & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Based on robot posture $q_i = [x_i, y_i]^T$, transfer point $H[x_h, y_h]^T$ in the global reference frame as $H_g[x_h, y_h]^T$

$$H_g = \begin{bmatrix} \cos \theta_l & \sin \theta_l \\ -\sin \theta_l & \cos \theta_l \end{bmatrix} H + q_i \quad (7)$$

Figure 1. The IoRT system structure. IoRT: Internet of Robotic Things.

Figure 2. Map point H using ORBBEC Astra depth camera.
Fuzzy sliding mode tracking control method

For robot motion control, two points should be considered such as reducing the calculation time and keeping the robot motion stable. To achieve these goals, we design a control input \((v, \omega)^T\), that can bound the error posture \(E_i = (x_{ie}, y_{ie}, \theta_{ie})^T\) for any initial error, and \(\lim_{t \to \infty} ||(x_{ie}, y_{ie}, \theta_{ie})^T|| = 0\).

Define the waypoint and the current state error as 
\[ e_i = [x_{ie}, y_{ie}, \theta_{ie}]^T = P\text{in} - P_i \]. Transfer state error \(e_i\) to state error posture \(E_i\).

\[ E_i = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = T_i \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} \]

(8)

If \(P\text{in} = P\text{ic}\) and the error \(e = 0\), then \(E = 0\). Based on the literature,\(^{31}\) using (1) and an equal equation \(x_{ie}\sin \omega_{ie} - y_{ie}\cos \omega_{ie} = 0\), the derivative matrix \(E_i\) is

\[ \dot{E}_i = \begin{bmatrix} \dot{x}_{ie} \\ \dot{y}_{ie} \\ \dot{\theta}_{ie} \end{bmatrix} = \begin{bmatrix} \cos \theta_{ie} \\ \sin \theta_{ie} \\ 0 \end{bmatrix} v_{iw} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \omega_{iw} \]

(9)

where \(v_{iw}\) and \(\omega_{iw}\) are the desired linear and angular velocities.

The switching function \(s_i\) for robot \(R_i\) is as

\[ s_i = \theta_{ie} \]

(10)

The exponent reaching law is designed as

\[ \dot{s}_i = -k_2 \text{sgn} s_i - k_3 s_i \]

(11)

where \(k_1 > 0\), \(k_2 > 0\), \(\text{sgn} s_i = \frac{s_i}{\sqrt{s_i^2 + \sigma^2}}\), and \(\sigma\) is a positive constant.

The control law is defined as

\[ \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} v_{iw} \cos \theta_{ie} + y_{ie} \omega_{ie} + k_1 x_{ie} \\ \omega_{iw} + k_2 \text{sgn} s_i + k_3 s_i \end{bmatrix} \]

(12)

To prove the stability of the control law, the Lyapunov function is designed as

\[ V_i = \frac{1}{2} v_{ie}^2 + \frac{1}{2} \theta_{ie}^2 \]

(13)

The derivative of the Lyapunov function is

\[ \dot{V}_i = x_{ie} \dot{x}_{ie} + \theta_{ie} \dot{\theta}_{ie} \]

\[ = x_{ie} (\cos \theta_{ie} v_{iw} - v_i + y_{ie} \omega_{ie} + s \dot{s}) \]

\[ = x_{ie} (\cos \theta_{ie} v_{iw} - (v_{iw} \cos \theta_{ie} + y_{ie} \omega_{ie} + k_1 x_{ie}) + y_{ie} (\omega_{iw} + k_2 \text{sgn} s_i + k_3 s_i)) - k_2 |s_i| - k_3 s_i^2 \]

\[ = -k_1 x_{ie}^2 - k_2 |s_i| - k_3 s_i^2 \leq 0 \]

Table 1. Fuzzy rule.

| \(|s_i|\) | PB | PM | PS | ZO | NS | NM | NB |
|---|---|---|---|---|---|---|---|
| \(k_1\) | PB | PM | PS | ZO | NS | NM | NB |
| \(k_2\) | PB | PM | PS | ZO | NS | NM | NB |

PB: positive big; PM: positive medium; PS: positive small; ZO: zero; NS: negative small; NM: negative medium; NB: negative big.

If a moving point is far from the sliding surface, \(k_2\) should be large to have fast trending motion from system status to sliding surface, if the moving point is close to the sliding surface, \(k_2\) should be small to have slow trending motion from system status to sliding surface, to weaken the charting. Therefore, the fuzzy sliding mode controller is designed. The fuzzy rule is described in Table 1. Seven fuzzy sets are used to describe the fuzzy partitions, as positive big, positive medium, positive small, zero, negative small, negative medium, and negative big.

The triangular function is defined as

\[ f(x) = \begin{cases} \max(0, 1 - \frac{2(a-x)}{w}), & x \leq a \\ \max(0, 1 - \frac{2(x-a)}{w}), & x > a \end{cases} \]

(15)

Using the centroid defuzzification method, the output is as

\[ \tilde{k}_2 = \frac{\sum_{i=1}^{7} \tilde{k}_2 \mu(k_2)}{\sum_{i=1}^{7} \mu(k_2)}, \quad \tilde{k}_3 = \frac{\sum_{i=1}^{7} \tilde{k}_3 \mu(k_3)}{\sum_{i=1}^{7} \mu(k_3)} \]

(16)

The fuzzy reaching law is defined as

\[ \dot{s}_i = -\tilde{k}_2 \text{sgn} s_i - \tilde{k}_3 s_i \]

(17)

Based on (8), the fuzzy sliding mode control law is as

\[ \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} v_{iw} \cos \theta_{ie} + y_{ie} \omega_{ie} + k_1 x_{ie} \\ \omega_{iw} + \tilde{k}_2 \text{sgn} s_i + \tilde{k}_3 s_i \end{bmatrix} \]

(18)

Simulations

Figure 3 shows sinusoid trajectories of three robots, controlled by fuzzy sliding mode control algorithm. Three robots form triangle formation. The black line presents leader robot’s trajectory. Red line and blue line are the trajectory of follower robot 1 and follower robot 2,
respectively. The parameter for the fuzzy sliding mode control method is $\sigma = 0.06$, $k_1 = 0.005$, $k_2 = 0.005$, and $k_3 = 0.001$. The time step is set as $dt = 0.1$ s. The predefined trajectory for the leader robot $R_1$ is as

$$x_{1w}(t + 1) = x_{1w}(t) + 0.07dt \quad (19)$$

$$y_{1w}(t + 1) = 0.5 \times \sin(2\pi \times dt/100) \quad (20)$$

$$\theta_{1w}(t + 1) = a \tan 2((y_{1w}(t + 1) - y_{1w}(t)), (x_{1w}(t + 1) - x_{1w}(t))) \quad (21)$$

The state errors of leader robot, follower robot 1 and follower robot 2 are shown in Figures 4 to 6. At first time, follower robots track leader robot from their start points, error occurs. Three robots should form and maintain a triangle formation. The ideal paths for three robots are that three robots form and maintain triangle formation correctly. Initially, three robots are not located in the right triangle formation. Two follower robots should track leader robot and form triangle formation with leader robot. Therefore, error occurs. At $t = 8$ s, the errors converge to zero, it means follower robots correctly and stably maintain formation with leader robot.

The simulation result in Figure 7 is used to compare with the experiment result. In Figure 7, the black lines are presented as obstacles, and the blue line and red line are shown as the trajectories of leader robot and follower robot, respectively. Initially, the desired distance and desired bearing angle between the leader and the follower robots is defined as 1 m and 90 degree, respectively. When the distance between the leader robot and the obstacle is small than 1.5 m, the leader robot avoids the obstacle. The follower robot switches safe formation by changing the desired distance and desired bearing angle with the leader robot.

**Experiment**

The experiment is used to show the effectiveness of the proposed IoRT system, where leader and follower robots successfully finish five tasks: localizing themselves, planning paths, avoiding obstacles, stable moving to waypoint, and creating a map. Based on the IoRT system structure...
shown in Figure 1, we created the multiple robots IoRT system. As shown in Figure 8, the robot used in the experiment is the Kobuki mobile robot. The main processor is the NI myRIO controller with a wireless module. Two Kobuki robots localize themselves using encoder and IMU. The robot heading angle is measured by a Three-Axis Digital Gyroscope (Part Name: L3G4200D). Based on the literature, figure in-place rotation test, the average error is in the range of –2.5 to 2 degree/rev. The errors of encoder and IMU result location error during the whole process. Therefore, we use fuzzy sliding mode tracking control method to minimize error. The leader robot is equipped with RPLiDAR A1 sensor to detect obstacles in the environment. The core of RPLiDAR A1 runs clockwise to perform a 360° omnidirectional laser range scanning for its surrounding environment. The detection range is 12 m. The sample rate is 8000 Sa/s. The angular resolution is 1°. The distance resolution is 0.2 cm. If the distance between the leader robot and an object is smaller than 1.5 m, the leader robot will avoid the obstacle by changing its orientation. The follower robot is equipped with ORBBEC Astra depth camera to create a depth map of the environment. Both leader and follower robots communicate with the host PC by wireless module. The algorithms are implemented in LabVIEW 2017 with Mathscript Module. The experiment platform is described in Figure 9. The DDS IoT application protocol is used to process the leader–follower formation control. The controller is master, and two Kobuki robots with myRIO controllers are clients. Master and clients communicate with each other by TCP/IP. The dash line between myRIO 1 and myRIO 2 presents that they will communicate with each other if the host controller fails.

Two robots’ trajectories and LiDAR sensor’s data is shown in Figure 10, which is compared with Figure 7.
Initially, two robots are located at $(0, 0, 0)^T$ and $(0, -1, 0)^T$. Follower robot tracks leader robot in 1 m desired distance and $\pi/2$ desired bearing angle. At $t = 40$ s, the leader robot detects an obstacle and avoid it. The follower robot switches formation to successfully avoid obstacles, by changing the desired distance to 0.95 m and desired bearing angle to 0 with the leader robot. At $t = 60$ s, the follower robot switches formation with the leader robot by changing the desired bearing angle to $-5\pi/6$ to avoid obstacles. The robots localize themselves based on global frame reference. The LiDAR sensor’s data is drawing based on leader robot’s local frame reference. To minimize calculation time and process time, the 270-degree LiDAR data is drawn, ignoring the 90-degree data behind the Kobuki robot. Therefore, the LiDAR sensor’s data is rotated. When two robots avoid obstacles, they cannot turn 90 degree in one sample time (0.01 s), due to the limitation of turning angle. Therefore, there is an error comparing the experiment result in Figure 10, and the simulation result in Figure 7. The experiment video is uploaded to https://www.dropbox.com/s/0dftkorqychz1qs/IoRT.mp4?dl=0.

The depth map from the camera is shown in Figure 11, based on the mapping algorithm in section “IoRT system structure.” Initially, we define a $600 \times 600$ grayscale image. The image pixel is equal to 70. The distance between two conjoint pixels is 1 cm in real world. After

**Figure 10.** Two robots’ trajectories and LiDAR sensor’s data: (a) $t = 17$ s, (b) $t = 44$ s, (c) $t = 60$ s, and (d) $t = 80$ s. LiDAR: light detection and ranging.
detecting by depth camera, the white space is presented as a free space. The black dot is present as the object detecting point. The red circle is presented as the robot.

If the leader robot is equipped with an NVIDIA Jetson TX2 controller, it will process cartographer SLAM, avoiding obstacles. Figure 12 shows the experiment result when testing in the corridor. The leader robot avoids obstacles, simultaneously localizes itself and builds a map.

Conclusions

This article proposed how to build an IoRT system for multiple mobile robots, to monitor events, fuse sensor data, use local robots to determine a best action, and then act to control multiple mobile robots. Multiple robots IoRT system finishes five tasks simultaneously, such as localizing themselves, planning paths, avoiding obstacles, moving to waypoint stable, and creating a map. A mapping method is discussed to create a map using depth camera’s data and robot’s posture. Based on LiDAR sensor data and Google cartographer, SLAM is also processed in this article. The fuzzy sliding mode tracking control method is used to minimize the robot’s tracking error. The simulation results prove the effectiveness of the proposed motion control algorithm. The experiment results show the effectiveness of proposed IoRT system structure. One host controller and two Kobuki mobile robots are integrated, and the robots successfully localize themselves, avoid obstacles, and build a map. In the future, we will consider to add more robots in the IoRT system and control them outdoor.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science [Grant Number 2017R1D1A3B04031864].

ORCID iD

Yanyan Dai https://orcid.org/0000-0003-2484-4585

References

1. Song Y, Kim DY, Ahn HS, et al. Simultaneous pedestrian and multiple mobile robots localization using distributed extended Kalman filter. In: IEEE international conference on robotics and biomimetics, Bangkok, Thailand, 22–25 February 2009, pp. 1065–1069.
2. Alonso-Mora J, Baker S, and Rus D. Multi-robot navigation in formation via sequential convex programming. In: International conference on intelligent robots and systems (IROS), Hamburg, Germany, 28 September–2 October 2015, pp. 4634–4641.
3. Ravankar A, Kobayashi Y, and Emaru T. Symbiotic navigation in multi-robot systems with remote obstacle knowledge sharing. Sensors 2017; 17(7): E1581.
4. Papoutsidakis M and Piromalis D. Intelligent algorithms based on data processing for modular robotic vehicles control. WSEAS Trans Syst 2014; 13: 242–251.
5. Ravankar A, Ravankar AA, Kobayashi Y, et al. Hitchhiking robots: a collaborative approach for efficient multi-robot navigation in indoor environments. Sensors 2017; 17(8): 1878.
6. Dai YY. Leader-follower formation control of multi-robot systems in cluttered environment. PhD Thesis, 2015.
7. Dai YY, Choi KS, and Lee SG. Adaptive formation control and collision avoidance using a priority strategy for nonholonomic mobile robots. Int J Adv Rob Syst 2013; 10: 1–14.
8. Egerstedt M, Hu X, and Stotsky A. Control of mobile platforms using a virtual vehicle approach. IEEE Trans Rob Auto 2001; 46(11): 1777–1782.
9. Lawton JRT, Beard RW, and Young BJ. A decentralized approach to formation maneuvers. *IEEE Trans Rob Auto* 2003; 19(6): 933–941.

10. Razafimandimby C, Loscri V, and Vegni AM. Towards Efficient Deployment in Internet of Robotic Things. In: Gravina R, Palau C, Manso M, Liotta A, and Fortino G (eds) *Integration, Interconnection, and Interoperability of IoT Systems. Internet of Things (Technology, Communications and Computing).* 2017, Cham: Springer.

11. ABI Research. Internet of robotic things, https://www.abiresearch.com/market-research/product/1019712-the-internet-of-robotic-things. 2014.

12. Simoens P, Dragone M, and Saffiotti A. The Internet of Robotic Things: a review of concept, added value and applications. *Int J Adv Rob Syst* 2018; 15. DOI: 10.1177/1729881418759424.

13. Kara D and Carlaw S. *The internet of robotic things.* Technical report, ABI Research, 2014.

14. Vermesan O, Bröring A, Tragos E, et al. Internet of robotic things: converging sensing/actuating, hypoconectivity, artificial intelligence and IoT platforms. In: Vermesan O, and Bacquet J (eds) *Cognitive hyperconnected digital transformation: internet of things intelligence evolution.* Norway: River Publishers Series, 2017, pp. 1–35.

15. Ray PP. Internet of robotic things: concept, technologies and challenges. *IEEE Access* 2016; 4: 9489–9500.

16. Xie GT, Gao H, Qian LJ, et al. Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE T Ind Electron* 2018; 65(7): 5999–6008.

17. Gao HB, Cheng B, Wang JQ, et al. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE T Ind Inform* 2018; 14(9): 4224–4231.

18. Hunziker D, Gajamohan M, Waibel M, et al. Rapyuta: the robocarth cloud engine. In: *IEEE international conference on robotics and automation,* Karlsruhe, Germany, 6–10 May 2013. DOI: 10.1109/ICRA.2013.6630612.

19. Lévis M. Quanser’s perspectives on internet of things for academic applications. *Quanser* 2018.

20. *A practical guide for connecting LabVIEW to the industrial IoT.* Austin: National Instruments, 2017.

21. Faisal M, Hedyar R, Alsulaiman M, et al. Robot localization using extended Kalman filter with infrared sensor. In: *International conference on computer systems and applications,* Doha, Qatar, 10–14 November 2014. DOI: 10.1109/AICCSA.2014.7073220.

22. Miah MS and Gueaieb W. Mobile robot trajectory tracking using noisy RSS measurements: an RFID approach. *ISA Trans* 2014; 53(2): 433–443.

23. Ghommem J, Mehrjerdi H, and Saad M. Robust formation control without velocity measurement of the leader robot. *Control Eng Pract* 2013; 21(8): 1143–1156.

24. Chang YH, Chen CL, Chan WS, et al. Type-2 fuzzy formation control for collision-free multi-robot systems. *Int J Fuzzy Syst* 2013; 15(4): 435–451.

25. Yang AL, Naem W, Irwin GW, et al. Stability analysis and implementation of a decentralized formation control strategy for unmanned vehicles. *IEEE Trans Control Syst Technol* 2014; 22(2): 706–720.

26. Li DY and Gao HB. A hardware platform framework for an intelligent vehicle based on a driving brain. *Engineering* 2018; 4: 464–470.

27. Qian DW, Tong SW, Guo JR, et al. Leader-follower-based formation control of nonholonomic mobile robots with mismatched uncertainties via integral sliding mode. *Proc Inst MechEng I J Syst Control Eng* 2015; 229(6): 559–569.

28. Solea R, Filipescu A, and Nunes U. Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties. In: *Asian control conference,* Hong Kong, China, 27–29 August 2009, pp. 1701–1706.

29. Li YD, Zhu L, Wang ZY, et al. Trajectory tracking for nonholonomic wheeled mobile robots based on an improved sliding mode control method. In: *ISECS international colloquium on computing, communication, control and management,* Sanya, China, 8–9 August 2009, pp. 55–58.

30. Dai YY, Kim YG, Wee SG, et al. A switching formation strategy for obstacle avoidance of a multi-robot system based on robot priority model. *ISA Trans* 2015; 56: 123–134.

31. Kanayama Y, Kimura Y, Miyazaki F, et al. A stable tracking control method for an autonomous mobile robot. In: *IEEE international conference on robotics and automation,* Cincinnati, OH, USA, 13–18 May 1990, pp. 384–389.

32. Kobuki user guide. https://docs.google.com/document/d/15k7UBnYY_GPmKzQCjzRGCW-4dIP7zl_R_7tWPLM0zKI/edit (accessed 15 March 2017).