Proceedings of the 2020 IEEE/SICE
International Symposium on System Integration
Honolulu, Hawaii, USA, January 12-15, 2020

A Multi-Robotic System for Environmental Dirt Cleaning

Chuong P. Le[1], Anh Q. Pham[2], Hung M. La[3], David Feil-Seifer[3]

***Abstract*— An industrial environment usually has a lot of waste that could cause harmful effects to both the products and the workers resulting in product defects, itchy eyes or chronic obstructive pulmonary disease, etc. While automatic cleaning robots could be used, the environment is often too large for one robot to clean alone in addition to the fact that it does not have adequate stored dirt capacity. We present a multi-robotic dirt cleaning algorithm for coordinating multiple iRobot-Creates as a team to efficiently clean an environment. Often, since some spaces in the environment are clean while others are dirty, our multi-robotic system possesses a path planning algorithm to allow the robot team to clean efficiently by increasing vacuum motor power on the area with higher dirt level. Overall, our multi-robotic system outperforms the single robot system in time efficiency while having almost the same total battery usage and cleaning efficiency result. The project source codes is available on our ARA lab's github: `https://github.com/aralab-unr/multi-robot-cleaning`**

I. Introduction

In an industrial environment, such as a factory or a warehouse, the work conditions are harsher than normal with wastes - hazardous or non-hazardous lying around, which the workers are exposed to every day. Since hazardous wastes cause visible damages, they are often more carefully handled than non-hazardous ones like metal dusts or ashes, the damages of which are long-term and not always visible. One of the damages that non-hazardous wastes can cause is product defects, sabotaging the purpose of the product and making the producer look bad. In addition, workers working in such an environment may have chronic medical issues such as skin problems, sinusitis, and eye problems. In particular, if inhaled, the dust may cause such symptoms as coughing or breathing issues and, in more serious cases, could damage the worker's lungs or other organs [1]. If they try to clean the environment by sweeping the dust, it will go into the air and create a higher chance of the workers inhaling it.

In this situation, using a dirt cleaning robot, such as the iRobot Roomba, would be a logical solution. Commercial automatic cleaning robots have become very popular nowadays because of their ability to clean autonomously. However, one of the earliest problems with it is that their cleaning process is completely random, and there is no guarantee that they would be able to clean all the space. Recently, though, there are new models of iRobot that could map an entire environment and systematically clean it.

Even though mapping and cleaning every inch of the environment indiscriminately can be an effective method, it lacks the ability to plan the best path as some areas of the map might be dirtier while others remain clean. Having a dirt model to estimate the amount of dirt in each small area of the environment will help create a more efficient path of cleaning for the robot to follow and, thereby, will increase cleaning efficiency.

Fig. 1. Experiment setup of two iRobot Creates in ARA lab.

Although most automatic cleaning robots could systematically clean a whole environment, an industrial environment is much too large for one robot to clean by itself in addition to the fact that the amount of dirt will exceed the dirt stored capacity of one robot. In this project, we will be using multiple iRobot Create 2 because of its low cost and its programmable platform [2] to assemble a multi-robotic cleaning team. In addition, we used the hokuyo laser scanner, and the dirt sensor that is already equipped in the iRobot Create 2 to create a dirt map of the environment. Then we implement the multi-robotic system, where all robots share the same map, know their positions in it, and effectively communicate with each other, to clean an entire industrial environment efficiently.

This paper is presented in the following order: After the related work, we will be discussing the method of

This material is based upon work supported by the National Science Foundation (NSF) #IIS-1757929. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF.

[1]Chuong Le is an Undergrad at the School of Electrical Engineering and Computer Science, University of Oklahoma, Norman, OK 73019, USA.

[2]Anh Pham is with the Faculty of Electrical and Electronic Engineering, Duy Tan University, Da Nang, Vietnam. This work was conducted when Anh was a visiting scholar at the Advanced Robotics and Automation (ARA) Laboratory, University of Nevada, Reno, NV 89557, USA.

[3]Hung La and David Feil-Seifer are with the Department of Computer Science and Engineering, UNR, NV 89557, USA.

Corresponding author: Hung La (`e-mail: hla@unr.edu`)

mapping and localizing with the Simultaneous Localization and Mapping (SLAM) and the Monte Carlo Localization system. After that, we will do a brief overview of the cell-wise Poisson process by Hess et al. [3] that we will be using to build a dirt map. Then, we will present the algorithms for our multi-robotic system to create an efficient cleaning path for each robot in the team. Finally, we will present the result of our experiment, evaluate our work and discuss the future usage of our results in the conclusion.

II. Related Work

Multi-robotic system has many benefits, mainly because it can accomplish task that single robot could not or take much longer time to do. Prime examples of its benefits include [4]–[15]. [9] uses rapidly exploring random tree (RRT) for vacuum robot navigation and path planning for large indoor vacuum cleaning. [7] by Connell and La uses the RRT^x for path replanning in a non-static environment. The multi-robot algorithm here yields equivalent or better paths and planning time efficient. [11] by Pham et al. creates a wildfire distribution model using multiple unmanned aerial vehicles (UAVs) to track and predict wildfire spreading. In this case if they were using a single UAVs it would not be able to cover the entire fire and create the wildfire model. Another case of using multiple robots for scalar field mapping [16], [17] that clearly indicated the benefit of multi-robot teaming rather than a single one.

If the robot is cleaning the same environment again, it would be a waste of time to re-scan the environment. In a paper by Zaman et al. [18], it was shown that laser and odometry data from scanning of a whole environment using SLAM can be saved into Yet Another Markup Language (YAML) and portable gray-map format (PGM) files using map saver and map server. YAML files shows the image while PGM shows the descriptions of the map. When implementing the Adaptive Monte Carlos Localization (AMCL) with the YAML and PGM file, the robot can localize itself inside the saved map [19].

One of the work used to navigate efficient path planning and navigation is the cell-wise Poisson process that used equations from the homogeneous Poisson process and the maximum likelihood estimator to predict where the dirt will be after allowing the robot to clean through the environment several times. Then it would create a dirt map that predicts the intensity of dirt in each cell and, using the Traveling Salesman Problem (TSP), determine an efficient path for the iRobot to clean. The project's benefits include less noise and less energy consumption in the robot, because of the time and work efficiency [3]. The idea in this paper is similar to the Poison cell-wise method. In this paper, we will create a dirt map and apply the multi-robotic system and propose path planning algorithm for multiple robots to clean the environment in an efficient manner.

III. Method

As mentioned before, we used 2 iRobot Create. We used the iRobot Create due to the fact that it is an affordable platform and has a built-in dirt detection sensor at the bottom inside the suction unit. The iRobot Create uses a piezoelectric sensor, which generates electrical pulses when dirt hits it and gives a measurement of the dirt reading [20]. We also mounted a hokuyo laser scanner. In the experiment, we used cardboard boxes to create an environment for the iRobot Creates to map and clean and substituted play sand as dirt. The play sand were scattered randomly in the amount of 10 grams per cycle.

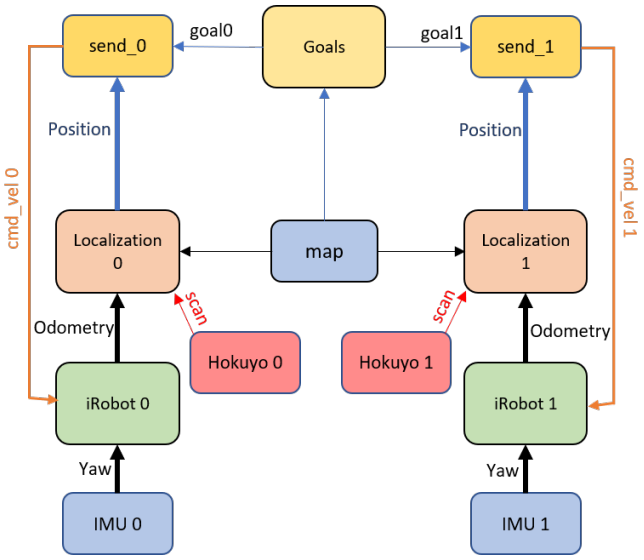

Fig. 2. The flowchart of our multi-robotic system for cleaning. For each robot, the IMU will improve its odometry. The localization system will take the odometry of the iRobot and the hokuyo laser data scan. The *Goals* node will publish a path for each robot, and each robot will have a *send* node that receive the information from *Goals* will tell the robot where to go.

A. Mapping the Environment

To build a map, we used the mounted hokuyo laser to scan the environment and implemented SLAM Gmapping [21] to convert the laser data into a 2D map. Once the environment map is completely built, we saved it and its data to send to the other two iRobots. After each robot received the map, they will localize themselves by using AMCL [18], [22], so they will know their positioning in the environment.

B. Creating the Dirt Map

For mapping the dirt level, we divided the environment into small square cells and have the iRobots clean the environment several times so that the dirt sensor can collect enough dirt data. Then to predict the dirt level of each cell, we use the homogeneous Poisson process used by Hess et al [3]. The expected dirt level or λ for each cell c in the $[s, t]$ interval is

$$E[N^c(t)] = \frac{(t-s)}{\sum(t_i - t_{i-1})} \sum_{i=1}^{n} k_i^c, \qquad (1)$$

where s is the latest cleaning. Every time the iRobot past through each cell c, it takes the dirt reading k_i^c, cleans the cell, and saves the time t_i. Therefore, $\sum_{i=1}^{n} k_i^c$ stands for the sum of all dirt reading of cell c.

C. Multi-robotic Cleaning System

Our multi-robotic cleaning system utilize 2 nodes function - *Goals* and *send* - as seen in Figure 2. The *Goals* node uses Algorithm 1 and 3 to divide the environment into separate cleaning space. Then the *send* node will uses Algorithm 2 and 4 to find the best cleaning path for each iRobot Create and send them there.

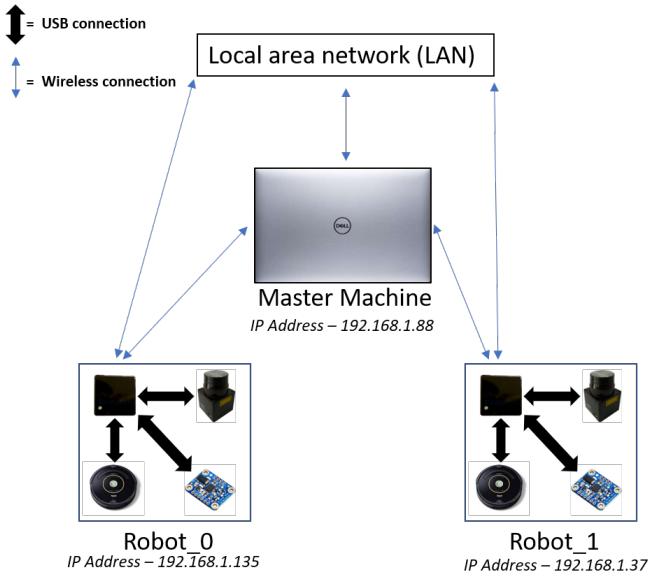

Fig. 3. The multiple machine connection.

1) Multi-Robot Networking: To connect the iRobots together, we connected all machines to the same router through local area network and implemented the MultipleMachines in Robotic Operating System (ROS) to configure multiple machines to use a single master using hostnames and IP addresses of the Intel NUCs on each iRobot. The master machine will be a laptop that we used to receive data from the iRobots and send directions to it. Figure 3 provides a visualization of our multi-robot networking.

2) Multi-robot Divide: After completing the dirt map, we need to divide it into separate cleaning space for each iRobot. To make the cleaning time of each iRobot relatively close, we tried to make the grids have even (or close to even) $lambda_{total}$. Our grid divided by Algorithm 1 uses a recursive method that divides the environment into many grids. The algorithm calculates the dirt level of each grid λ_g by λ_{total}/number of robot presented. It starts with $Grid1$ being the original grid map with all available cells of the dirt map while the other grids have all unavailable cells. The recursive function would, as it find the cells grids to divide, transfer the available cells from $grid1$ to $grid_i$ until λ_g is reach, then it will move on to the next grid to transfer until the number of grids divided $gridDivided$ is equal to the number of grids num. Since most of the time in real life, the grid's λ can not be evenly divided, we need to make sure that the grids we can make them close to even so we put p as the counter that increases if the map can not be divided evenly. For example, if a map can not be divided into 5 and 5, p would increment so that map will try 4 and 6. Even though we are using just two vacuum robots, any number of robots, num can be implemented in this algorithm because it will divide the environment into num cleaning space.

Algorithm 1: Grid Divide Algorithm

Input: number of robots num, total dirt level λ_{total}, grid V
$grid_1 = V$ set all cells c in other $grid_{2-num}$ to 0
p is a counter increase if map can't divide evenly
$gridDivided$ is the counter dividing grids
while $gridDivided < num$ **do**
  while p *less than* λ_{total}/num **do**
    find the starting cell c_{start} using row major order
    $l = \lambda_{total}/num$ - p
    recursive(c_{start}, l)
    if *solution is found* **then**
      break
    end
    else
      reset the 2 grids
      increment p
    end
  end
  increment $gridDivided$
end

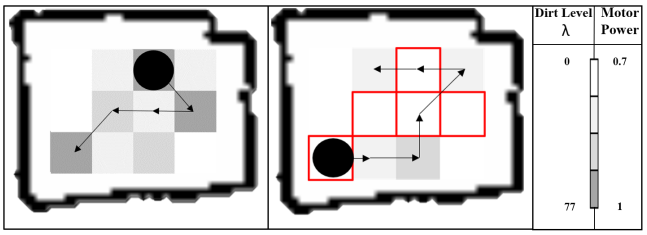

Fig. 4. An example of the path planning algorithm we create for each iRobot in their respective cleaning space. The cell colors represent c_λ. The deeper the color, the higher the dirt level and motor power. After the iRobot clean a cell, it will be labeled as clean/white as shown on the right picture with the red outlined cells.

3) Path Planning: After assigning each iRobot to their respective cleaning grid map, we created a path planning algorithm (Algorithm 4) that utilize the A* Search Algorithm as presented in Algorithm 2 for each for each robot. We used the A* because of its ability to find the shortest path possible as well as the fact that it also includes obstacle avoidance [23].

Our A* algorithm finding path by searching for the nearest cell c and using the heuristic function. It ignores the cells that is already cleaned c_λ and obstacles. In addition, if it finds a viable diagonal cell to go to, if there is obstacle on either side of the two cells, it ignores that cell as well.

In our path planning algorithm, we want to prioritize the cell with the highest λ before the clean cell because

Algorithm 2: A* Path planning Algorithm

```
1  Initialize closedList and openList
2  Note: closedList and openList is a list of cells c each has: coordinate (x,y), cell
   dirt level λ, distance from starting cell to the next cell g , distance from next
   cell to goal h, parameter of g + h f
3  Put the starting cell c in openList
4  while openList is not empty do
5      find the node with the least f in the openList q and pop q off openList
6      find q's 8 cell path p (east northeast north east-west west southwest south southeast) and set their parents to q
7      for each p do
8          if p = goal then
9              stop searching and break out
10         end
11         p_h = distance from p to goal, p_g = distance from q to p + q_g, p_f = p_h + pg
12         if a cell i has the same position as p in openList then
13             if i_f less than p_f then
14                 skip it
15             end
16         end
17         else if a cell i with the same position as p in closedlist then
18             if i_f less than p_f then
19                 skip it
20             end
21         end
22         else if p is cleaned then
23             skip it
24         end
25         else if path to p is diagonal then
26             if there is obstacle(s) on either side then
27                 skip it
28             end
29         end
30         else
31             add the node to the openList
32         end
33     end
34     push q in closeList
35 end
```

if the machine breaks down, we have already cleaned the dirtiest part of the map. In addition, we want to use more vacuuming force when cleaning dirtier cells. As can be seen in Algorithm 2, it finds and clean the cell with the highest λ before the counter decrements to the next dirtiest cell. However, the problem is that if we use this method to go to every single cell, it would take a very long time. Our solution to that is on the way it should also cleaned the cells that it passed and label it as cleaned so the A* algorithm can ignore that cell while finding a path. The iRobot should use different motor force depending on c_λ. In a real situations, there may be times when there will not be a path that ignores all of the cleaned cells. If that is the case, then just for that path, we use an alternate A* Search Algorithm that does not ignore cleaned cells. That algorithm will just be Algorithm 2 without line 26-28.

Our multi-robotic system is demonstrated in figure 6 where it divided the environment into two cleaning space for two iRobot to path plan and clean.

IV. Experiment

A. Testing multi-robotic system versus the built-in system

The experiment setup is shown in Figure 1.

In our experiment, we are testing our multi-robotic system against the iRobot Create's cleaning system and evaluate the performance of cleaning time, total battery usage. First, we timed the iRobot Create's cleaning process until it is finished and returned it its home-base. Then we check the battery usage percentage and recorded it. Then, we implemented our multi-robotic system and timed its cleaning process. When all iRobots stop, we also measure each of its battery usage percentage using the information display from [24], added

Algorithm 3: Recursive Function

Note: (for ALL Algorithms in this paper) even though the cell dirt level is [0,77], in the algorithm we made it so that λ: [0,3], where 0 = [0,19], 1 = (19,38], 2 = (38,57], and 3 = (57,77].
Function recursion(starting cell c, grid's required λ_{total})
if λ_{total} *is 0* **then**
    if *the grid1 is connected and grid2 is connected* **then**
        solution is found
    end
    return;
end
set starting cell's position in grid1 to unavailable
set starting cell's position in grid2 to available
λ_{total} -= c_λ
if λ_{total} ¡ *0* **then**
    return
end
find neighboring cells of east, west, south, north
for *each neighboring cell* **do**
    if *cell is available* **then**
        recursion(cell, λ_{total})
    end
end
return

Algorithm 4: Send

Initialize V an cell array Initialize $levelCounter$ the counter that keeps track of the current highest λ
Set $levelCounter$ to highest λ of a single cell
while $levelCounter$ *bigger than 0* **do**
    find all cell with λ = to $levelCounter$ and put it in V
    while *V is not empty* **do**
        find the cell c_{least} with the least distance away from starting point find the path to c_{least} using the A* Path Planning Algorithm
        if *path is not found* **then**
            use an alternative A* star path planning that does not ignore clean cells
        end
        go to c_{least} along the way, turn the dirt motor power on according to the dirt level of the cell on the way and change its status to cleaned/λ = 0 remove c_{least} from V
    end
    decrement $levelCounter$
end

all three together and recorded it. We repeat the process ten times and took the average time and total battery usage. Overall, there were no outliers in either categories - duration and total battery usage - and the spread was relatively small. For cleaning efficiency, each time it finished cleaning, we decided to observe the environment and decided that the result is very satisfactory. The demonstration of two robots cleaning an environment can be seen in this link: `https://youtu.be/sqrj5DN1SGQ`

B. Result

Figure 5 presents the overall result of our experiment. While the total battery usage of the multi-robot system is higher than the built-in cleaning system of a single iRobot, the total cleaning duration of our multi-robotic system is less than 1/2 the duration of a single robot. This experiment shows the time cleaning efficiency of our multi-robotic system while having almost the same battery usage and yielding little to no dirt remaining in the environment.

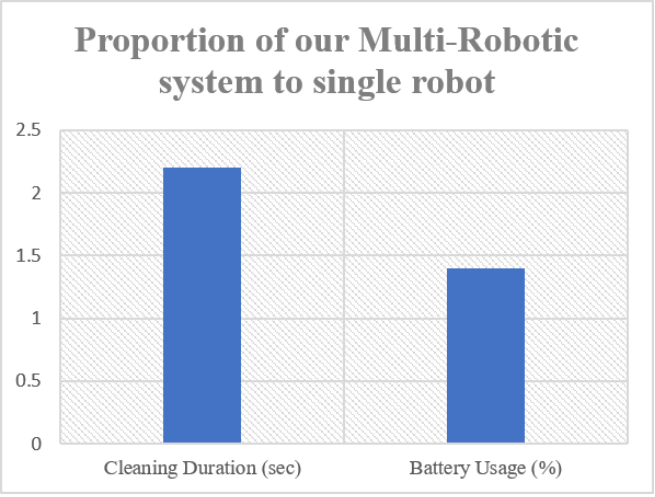

Fig. 5. The result of our experiment.

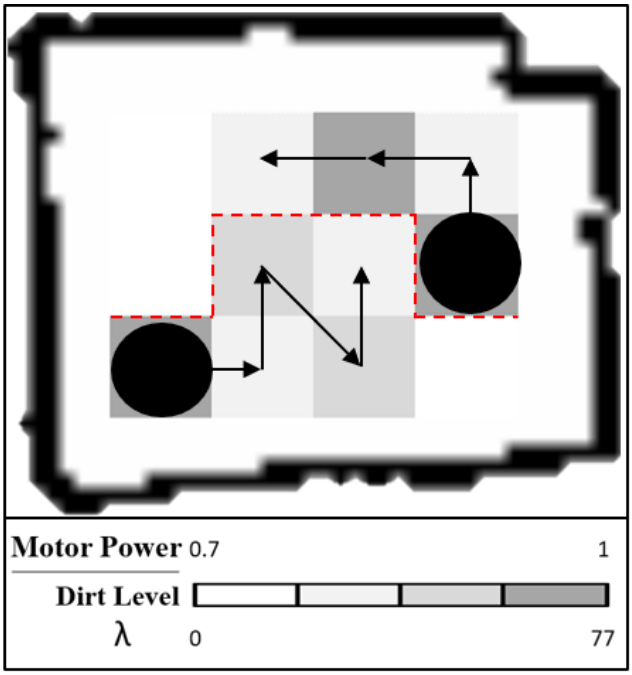

Fig. 6. The motion planning of two robots with respect to the dirt level a divided environment.

V. Conclusion

In this paper, we present a multi-robotic algorithm along with a dirt model for industrial environment cleaning. After collecting dirt reading data, the dirt model predicts the dirt level for each small area on the environment map. Our algorithm divides the model relatively even in terms of total dirt level among the robots and creates the best path for each robot as well as cleaning duration of each small area depending on its dirt level. The proposed algorithm was implemented and validated on two iRobot-Creates. The result was that while the battery usage of our multi-robotic system was close, its cleaning duration was approximately twice as fast as the single iRobot cleaning system. In the future, we plan to make our system viable for non-static environment such as when the map changes as well as collecting more data to update the new areas in the dirt model. In addition, we aim to have each of our iRobot be able to do replanning the path in case of a moving obstacle like [7], [25] and avoiding trapped by convex shaped obstacles by using a combination of rotational force field [26], [27] and repulsive artificial potential force field [28].

References

[1] L. Rushton, "Health hazards and waste management," *British Medical Bulletin*, vol. 68, no. 1, pp. 183–197, 2003. [Online]. Available: http://dx.doi.org/10.1093/bmb/ldg034

[2] B. Tribelhorn and Z. Dodds, "Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education," *IEEE International Conf. on Robotics and Automation*, pp. 1393–1399, 2007.

[3] J. Hess, M. Beinhofer, D. Kuhner, P. Ruchti, and W. Burgard, "Poisson-driven dirt maps for efficient robot cleaning," *IEEE International Conf. on Robotics and Automation*, pp. 2245–2250, 2013.

[4] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, "Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach," *Automatica*, vol. 100, pp. 75 – 81, 2019.

[5] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243–255, 2002.

[6] H. M. La and W. Sheng, "Dynamic target tracking and observing in a mobile sensor network," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996 – 1009, 2012.

[7] D. Connell and H. M. La, "Extended rapidly exploring random treebased dynamic path planning and replanning for mobile robots," *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, p. 1729881418773874, 2018.

[8] T. Nguyen, T. Han, and H. M. La, "Distributed flocking control of mobile robots by bounded feedback," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2016, pp. 563–568.

[9] A. Nikitenko, J. Grundspenkis, A. Liekna, M. Ekmanis, G. Kulikovskis, and I. Andersone, "Multi-robot system for vacuum cleaning domain," *PAAMS*, pp. 363–366, 2014.

[10] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, Jan 2015.

[11] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Transactions on Sys., Man, and Cyber.: Systems*, pp. 1–12, 2018.

[12] T. Han, H. M. La, and B. H. Dinh, "Flocking of mobile robots by bounded feedback," in *2016 IEEE International Conf. on Automation Science and Engineering (CASE)*, Aug 2016, pp. 689–694.

[13] M. T. Nguyen, H. M. La, and K. A. Teague, "Compressive and collaborative mobile sensing for scalar field mapping in robotic networks," *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 873–880, 2015.

[14] F. Muoz, E. S. EspinozaQuesada, H. M. La, S. Salazar, S. Commuri, and L. R. GarciaCarrillo, "Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 9, pp. 1566–1588.

[15] M. T. Nguyen, H. M. La, and K. A. Teague, "Collaborative and compressed mobile sensing for data collection in distributed robotic networks," *IEEE Trans. on Control of Network Systems*, pp. 1–1, 2018.

[16] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Trans. on Sys., Man, and Cyber.: Systems*, vol. 45, no. 1, pp. 1–12, Jan 2015.

[17] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 766–778, April 2013.

[18] S. Zaman, W. Slany, and G. Steinbauer, "Ros-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues," *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pp. 1–5, 2011.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[20] E. A. Gordon, "Vacuum cleaner with dirt detection," US Patent US 5 608 944, 03 11, 1997. [Online]. Available: https://patents.google.com/patent/US5608944A/

[21] "Slam gmapping," https://github.com/ros-perception/slam_gmapping.

[22] "Navigation," https://github.com/ros-planning/navigation.

[23] R. Belwariar, "A* Search Algorithm." [Online]. Available: https://www.geeksforgeeks.org/a-search-algorithm/

[24] P. Jacob, "create autonomy," https://github.com/AutonomyLab/create_autonomy.

[25] D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using rrt," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2017, pp. 1429–1434.

[26] A. D. Dang, H. M. La, and J. Horn, "Distributed formation control for autonomous robots following desired shapes in noisy environment," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sept 2016, pp. 285–290.

[27] A. D. Dang, H. M. La, T. Nguyen, and J. Horn, "Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force-based approach," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1729881419847897, 2019.

[28] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2018.