3rd International Conference on
Robotics Automation and Non-Destructive Evaluation,
Chennai, India, 23 April 2022.
https://doi.org/10.13180/RANE.2022.23.04.04

COMPOSITE ROBOT ALGORITHM AND MULTI-ROBOT FORMATION STRATEGY FOR COLLABORATIVE MATERIAL HANDLING SYSTEMS

M. N. ADITHYA and VIRAL PANCHAL
Mechatronics Engineering, SRMIST, Kattankulathur,
Chennai, Tamil Nadu, 603203, India
E-mail: adithyamn000@gmail.com, viralpanchal1999@gmail.com
www.srmist.edu.in

JOSEPH JOEL
Mechatronics Engineering, SRMIST, Kattankulathur,
Chennai, Tamil Nadu, 603203, India
E-mail: josephjoel3099@gmail.com

PILLAI R. RANJITH
Mechatronics Engineering, SRMIST, Kattankulathur,
Chennai, Tamil Nadu, 603203, India
E-mail: ranjithr1@srmist.edu.in

Collaborative Robotic Systems traditionally are systems that employ robot arm-manipulators that are intended to share a space with humans or other robots and collaborate with each other to complete a task specified. The use of Collaborative multi-robot systems for material handling is still being explored. These systems use multiple mobile robots to automate a task primarily, transportation-based. This work presents a multi-robot formation strategy and a composite robot algorithm for autonomous transportation of an object from one location to another within the industrial workspace. Particularly, this approach uses multiple-mobile robots that evaluate a suitable formation to adopt based on the dimensions of the object that is expected to be transported. A composite robot algorithm is employed which considers the robots in formation as one single entity and plans a trajectory to the goal location. The algorithm was tested and validated by measuring the velocity data obtained from the robots in formation and analysed whether the formation is maintained until the goal is reached.

1. Introduction

Advances in industry 4.0, particularly automation in the manufacturing industry is developing at a rapid rate. There are many new methodologies being proposed for automation in the material handling segment. Notably, the use of autonomous mobile robots to locate, track and move inventory in a warehouse is one that is very promising. Operating Collaborative Mobile Robots (CMRs) to efficiently handle material is one that is actively being researched. CMRs when compared to traditional Collaborative Robot-Arm manipulators (Cobots) [1] are more versatile in applications where robot mobility is required, especially material handling in a warehouse facility.

This work focuses particularly on transportation of a material that is packaged into a box from one location to another using multiple Autonomous Mobile Robots (AMRs) that can collaborate with each other. Since, any warehouse employs multiple robots to manage and automate the its operations, it is logical to make use of multiple robots for a given transportation task. This work provides a formation strategy and a composite robot algorithm for these AMRs, particularly locations for each mobile robot to arrive at and trajectory planning to transport an object of a given dimensions to any location in the global map by considering the robots in formation as a single entity, a composite robot.
Generally, the primary benefit of using multiple mobile robots for transporting an object is the design of these robots can be small since the weight required to be carried by each robot also decreases with an increase in the total number of robots assigned for that particular task,

therefore making it cost effective as well. Performing an operation otherwise requires a robot to be designed to withstand heavy loads leading to further complications such as the need for strong structural base, pricey actuators and accompanying sensors.

Current Research show various ways of tackling multi-robot formation. Few among them are multi-robot formation done by using a master-slave concept [3,7], mimicking crystal formation of molecules [4], collaboration of robots with different configurations [5]. Efforts have also gone into deriving a multi-robot model [6]. In addition, research and development in path planning algorithms like implementing A-Star, Dijkstra algorithm and D* lite algorithm [8,10] for multiple robots, optimized path algorithms [9] and collision avoidance [11,14]. Efforts to keep the robots in a tightly coupled formation is also necessary when traversing through environments with obstacles. This is very important when group of robots are carrying a load. Solution for this problem is explored in other papers [12] and a different approach is presented below. However, the proposed work aims to evaluate a formation that is dependent on the dimensions of the both the object and the robots that are transporting it. Furthermore, the robots in formation are a single entity or a composite robot while evaluating a path and trajectory for transportation.

2. Multi-Robot Formation Strategy

This work proposes a robot formation strategy for holonomic robots. As discussed earlier, there are multiple robots assembling at a location to maneuver an object. Evidently, the formation is influenced by the object dimensions and the robot design parameters. The optimum robot formation particularly in the case of material handling is aimed to distribute the weight of the package equally. The robot formation which is influenced by these parameters is either in the form of an equilateral triangle or a straight line such that there is always at least two lines (optimally three) of symmetry among the robots. This algorithm provides locations for the robots to assemble at, to perform a transportation task.

The first part of the algorithm is to find which formation the robots have to adopt. This formation depends on two important parameters, they are the robot dimensions and the object dimensions. It is assumed that the robot has an overall chassis radius of ’r’. Instinctively, it can be understood that the chassis radius and the object dimensions such as its length ’l’ and breadth ’b’ are directly related to the formation adopted.

The formation to be adopted is evaluated by considering the closest placement of the robots. Figure:1 illustrates the closest formation of the robots. From this, the formation that the robots choose can be derived. This can either be a 3-robot triangle formation or two robot straight line formation. As discussed earlier, this depends on the object’s length ’l’ and breadth ’b’. It can be inferred from the Figure 1. that, the robots must choose the three-robot formation when the length ’l’ and breadth ’b’ of the object is greater than $(r \times (2 + \sqrt{3}))$ and must choose the two-robot formation when the length ’l’ and breadth ’b’ of the object is less than $(r \times (2 + \sqrt{3}))$.

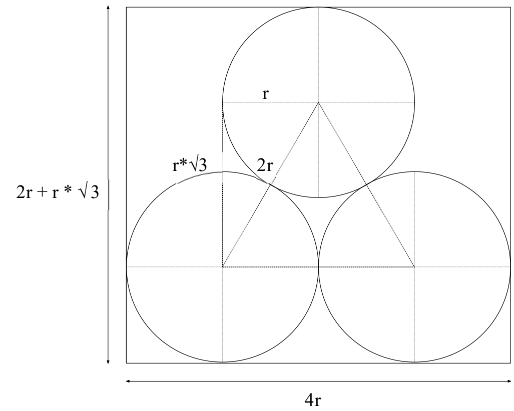


Figure 1 : Formation Parameters

2.1. *Three Robot Triangle Formation*

Either the length 'l' or the breadth 'b' of the object should be greater than (4r) with the other being greater than (r × (2 + √3)). The formation can be horizontal (height parallel to x axis) or vertical (height parallel to y axis) w.r.t. world frame

$$\text{Case1: Horizontal Formation} = \{l > (4r)\} \cap \{b > \left(r \times \left(2 + \sqrt{3}\right)\right)\} \tag{1}$$

$$\text{Case2: Vertical Formation} = \{b > (4r)\} \cap \{l > \left(r \times \left(2 + \sqrt{3}\right)\right)\} \tag{2}$$

The coordinates of the robots in the formation in both cases are derived with the following objectives in mind, to distribute the weight of the object equally between the robots, to maintain a symmetric triangular formation i.e., be 120° apart from each other w.r.t to center of the object. 'x' and 'y' are the starting coordinates of the rectangle (bottom left vertex).

2.1.1. *Case 1: Horizontal Formation*

$$\text{Robot 1 Coordinates} = \left(x + \frac{l}{4}\right), \left(y + \frac{b}{2}\right) \tag{3}$$

$$\text{Robot 2 Coordinates} = \left(x + \frac{3l}{4}\right), \left(y + \frac{b\sqrt{3} + l}{2\sqrt{3}}\right) \tag{4}$$

$$\text{Robot 3 Coordinates} = \left(x + \frac{3l}{4}\right), \left(y + \frac{b\sqrt{3} - l}{2\sqrt{3}}\right) \tag{5}$$

2.1.2. *Case 2: Vertical Formation*

$$\text{Robot 1 Coordinates} = \left(x + \frac{1}{2}\right), \left(y + \frac{3b}{4}\right) \tag{6}$$

$$\text{Robot 2 Coordinates} = \left(x + \frac{l\sqrt{3} + b}{2\sqrt{3}}\right), \left(y + \frac{b}{4}\right) \tag{7}$$

$$\text{Robot 3 Coordinates} = \left(x + \frac{l\sqrt{3} - b}{2\sqrt{3}}\right), \left(y + \frac{b}{4}\right) \tag{8}$$

Figure.2 illustrates the formations that the robots can adopt when the object is large enough to accommodate three robots. As discussed earlier in Eq.1 & Eq.2, there are two formations that the robots can adopt, figure on left depicts horizontal formation and figure on right depicts the vertical formation.

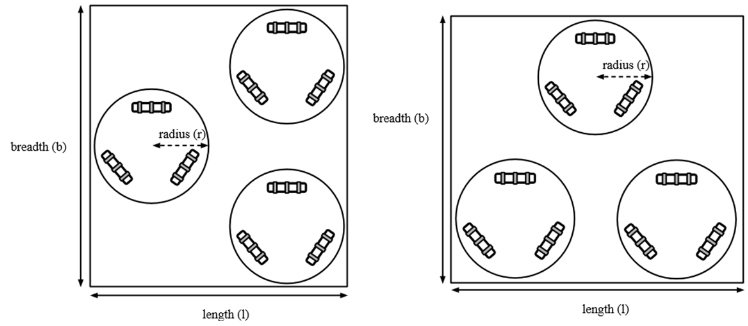


Figure 2 : Three Robot : Triangle Formation

2.2. *Two Robot Straight Line Formation*

Either the length ’l’ or the breadth ’b’ of the object should be greater than twice the robot radius ’r’ with the other being greater than the robot radius ’r’. The formation can be horizontal (parallel to x axis) or vertical (parallel to y axis) w.r.t. the world frame.

$$\text{Case1 : Horizontal Formation} = \{2l > b > 2r\} \tag{9}$$
$$\text{Case 2 : Vertical Formation} = \{2\mathrm{b} > \mathrm{l} > 2\mathrm{r}\} \tag{10}$$

The coordinates of the robots in the formation in both cases are derived with a primary objective to distribute the weight equally between the robots equally.

2.2.1. *Case 1 : Horizontal Formation*

$$\text{Robot 1 Coordinates} = \left(x + \frac{l}{4}\right), \left(y + \frac{b}{2}\right) \tag{11}$$

$$\text{Robot 2 Coordinates} = \left(x + \frac{3l}{4}\right), \left(y + \frac{b}{2}\right) \tag{12}$$

2.2.2. *Case 2: Vertical Formation*

$$\text{Robot 1 Coordinates} = \left(x + \frac{l}{2}\right), \left(y + \frac{3b}{4}\right) \tag{13}$$

$$\text{Robot 2 Coordinates} = \left(x + \frac{l}{2}\right), \left(y + \frac{b}{4}\right) \tag{14}$$

Fig.3 illustrates the formations that the robots can adopt when the object is not large enough to accommodate three robots. In such as case, only two robots are required. As discussed earlier in Eq.9 & Eq.10, there are two formations that the robots can adopt, figure on the left depicts the vertical formation and the figure on the right depicts the horizontal formation.

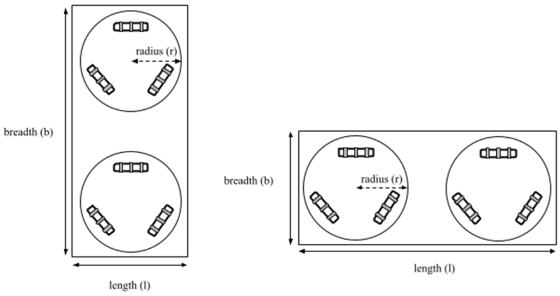


Figure 3: Two Robot: Straight Line Formations

Upon evaluating the robot's formation and its corresponding coordinates for a given object, traditional path planning algorithms can be used by the software stack to plan a path from its current location to the coordinates provided above.

3. Composite Robot Algorithm

The primary aim of this algorithm is to ensure that an optimum path is planned to the destination location such that, this formation is maintained until the object reaches this location. As discussed earlier, this can be achieved by considering the formation as a composite robot, meaning as a single entity. Since the robots in formation are of the same dimensions, design and kinematics, the wheel velocities required by the robots at different time steps to reach the destination are same, assuming that the orientation of the composite robot remains same at all times. In addition, the parameters of formation either triangle or straight line are known. Upon considering them as a composite robot, a single set of parameters can be provided to the composite robot and the robots will be able to reach the destination, autonomously

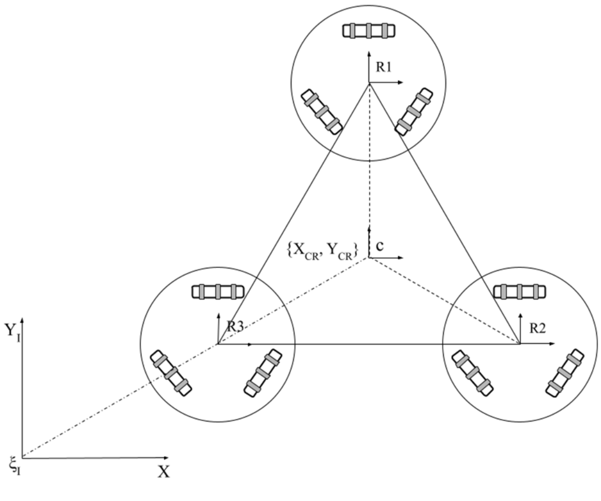
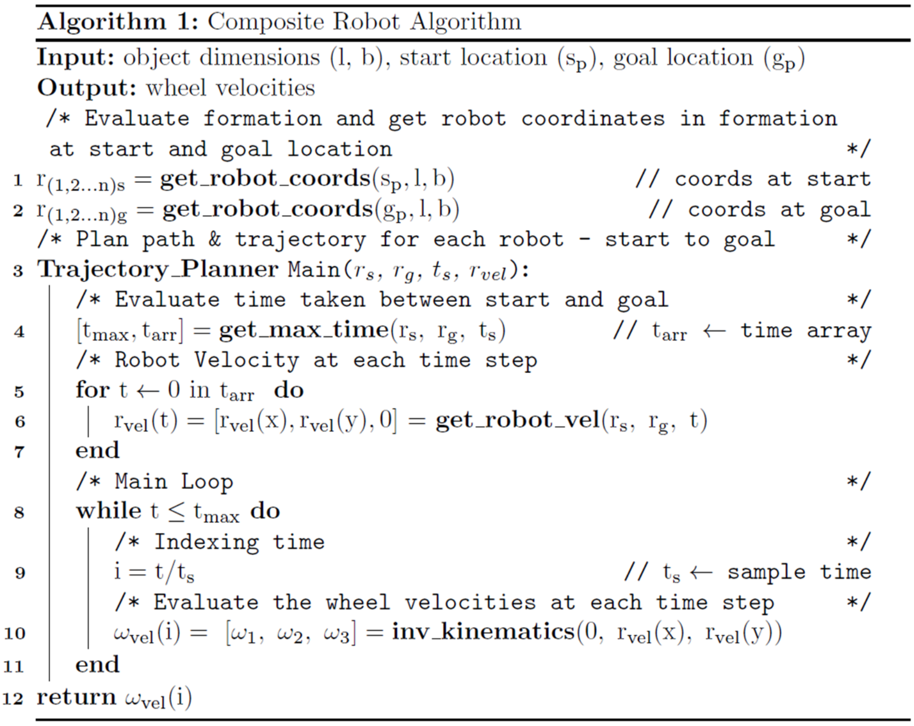


Figure 4 : Composite Robot Formation

Fig.4 is a representation of the composite robot in a three-robot formation. The frame-assignment convention is detailed as follows, frame C is the center of the composite robot, frames $R1, R2, R3$ are the frames of the robots in formation. As discussed earlier, since the coordinates of robots w.r.t the world frame ε_I are known, the formation strategy can be used to find the start (s_p) and goal locations(g_p) for each robot and use the composite robot algorithm to find the trajectory, such that the robots reach the destination in formation. Algorithm 1 elucidates the composite robot algorithm.



```
Algorithm 1: Composite Robot Algorithm
Input: object dimensions (l, b), start location ($s_p$), goal location ($g_p$)
Output: wheel velocities
/* Evaluate formation and get robot coordinates in formation
 at start and goal location                                      */
$r_{(1,2...n)s}$ = get_robot_coords($s_p$, l, b)        // coords at start
$r_{(1,2...n)g}$ = get_robot_coords($g_p$, l, b)        // coords at goal
/* Plan path & trajectory for each robot - start to goal         */
Trajectory_Planner Main($r_s$, $r_g$, $t_s$, $r_{vel}$):
    /* Evaluate time taken between start and goal                 */
    [$t_{max}$, $t_{arr}$] = get_max_time($r_s$, $r_g$, $t_s$)     // $t_{arr} \leftarrow$ time array
    /* Robot Velocity at each time step                           */
    for $t \leftarrow 0$ in $t_{arr}$ do
        $r_{vel}(t) = [r_{vel}(x), r_{vel}(y), 0]$ = get_robot_vel($r_s$, $r_g$, t)
    end
    /* Main Loop                                                  */
    while $t \leq t_{max}$ do
        /* Indexing time                                          */
        $i = t/t_s$                                  // $t_s \leftarrow$ sample time
        /* Evaluate the wheel velocities at each time step        */
        $\omega_{vel}(i) = [\omega_1, \omega_2, \omega_3]$ = inv_kinematics(0, $r_{vel}(x)$, $r_{vel}(y)$)
    end
return $\omega_{vel}(i)$
```

Figure 5: Composite Robot Algorithm

The **get _robot coords** function evaluates the coordinates for all the robots in the formation. As discussed earlier, this uses the multi-robot formation strategy to evaluate the best suitable formation based on the object dimensions (l, b). After deciding the formation, Eqs.3-8 and Eqs.11-14 are used to evaluate the robot's coordinates in the formation w.r.t world frame.

The **Trajectory Planner** plans the trajectory that the robot has to follow in order to reach the goal location. To be more precise, it evaluates a straight line path from the start to goal and provides the wheel velocities for a robot at each time step to ensure that the robot reaches the goal location. For the purpose of localization, the robots each have encoders on their wheels and these allow the robot to understand the approximate position with respect to the origin. Dead reckoning algorithm for three-wheel holonomic robot is used to convert wheel encoder counts to distance travelled in X and Y directions. The coordinate system used is shown in Fig.4.

The **get max _time** function computes the time it takes for a robot between the start and goal location. First it calculates the Euclidean distance from the start to goal. Eq.15 describes this, where (x_s, y_s) is the start and (x_g, y_g) is the goal coordinates.

$$\|d\| = \sqrt{\left(x_g^2 - x_s^2\right) + \left(y_g^2 - y_s^2\right)} \tag{15}$$

Then calculates the maximum time it takes for the robot to maneuver between these points using Eq.16, where”$r_{vel(max)}$” is the maximum robot velocity.

$$t_{max} = \frac{d}{r_{vel(max)}} \tag{16}$$

Then, splits the time in discrete time steps (t_{arr}) based on the sample time (t_s).

The **get robot _vel** function evaluates the robot velocity (r_{vel}) at each time step. It is assumed that the map doesn’t have any obstacles The function provides the a straight line path as follows, first computes the angle of inclination (’m’ - slope) of line through the start and goal coordinates w.r.t x and y axis. Then, computes the velocity components of the robot in x and y axis. Eqs.17-19 describes the angle of inclination (θ) and horizontal $\left(r_{vel}(x)\right)$ and vertical $\left(r_{vel}(y)\right)$ velocity components.

$$\theta \; = \; \tan^{-1}(m) = \tan^{-1}\frac{\left(y_g - y_s\right)}{\left(x_g - x_s\right)} \tag{17}$$

$$r_{vel}(x) = r_{vel(max)} \times \cos\theta \tag{18}$$

$$r_{vel}(y) = r_{vel(max)} \times \sin\theta \tag{19}$$

Then, uses the robot’s **inverse kinematics** function to find the corresponding wheel velocities (ω_{vel}(i)) for wheel 1 (ω_1), wheel 2 (ω_2), wheel 3 (ω_3) at each time step. The kinematics for the omni-directional robot is derived using the usual parameters [15].

$$\omega \; = \; \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = H(0)\epsilon_r = \frac{1}{r}\begin{bmatrix} -d & -\frac{1}{2} & 0 \\ -d & -\frac{1}{2} & -\sin\left(\frac{\pi}{3}\right) \\ -d & 1 & \sin\left(\frac{\pi}{3}\right) \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{20}$$

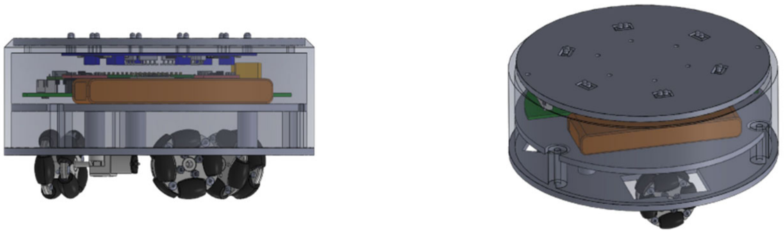
4. Experimentation and Results

4.1. *System Design*

A brief description of the system designed and fabricated to test and validated the proposed algorithms are detailed in this section. The aim of the design was to develop a collaborative multi-robot system that can handle material, i.e. use multiple mobile robots to transport an object from one location to another within a pre-defined arena, autonomously by collaborating with each other.

There are 3 important segments of the system design, they are a design of the robot, an arena and a user interface, the working of the complete system is as follows. Initially, the user will operate the user-interface and scan the bar code on the box. The bar-code contains the necessary data (Object Dimensions, Location, and Destination)for the operation to start. Upon scanning the bar-code, the mobile robots will start from their respective home positions and reach the location (job coordinates) where the object is expected to be placed. The job coordinates provided are evaluated from the multi-robot formation strategy discussed earlier. The formation and the job coordinates depend on the dimensions of the box. Upon arriving at their respective location, the robots will wait for the user to place the object on top of them. The robots will navigate from the job location to the destination, while maintaining formation with the object on top of them. Once they reach the goal position where the object is then stacked, the robots will navigate back to their respective home position, where they wait for the next job.

Design of the Robot : The configuration of the designed robot is an Triple Omni-wheel configuration. The chassis radius is 175 (mm) and height is 80 (mm), the total weight of the robot is 800 (gms). The radius of the omni-wheel is 19 (mm) and the distance from the wheel to robot center is 53 (mm). For optimum working of the composite robot algorithm, the robots should move instantaneously in any direction/path at any time instant, hence the said configuration was adopted.



(a) Robot : Front View

(b) Robot : Isometric View

Figure 6 : Assembled Robot

Fig.6 illustrates the front and isometric view of the robot designed. It can be seen that there are three distinct plates each housing different components that are considered integral to the functioning of the robots. The bottom plate accommodates the motors, magnetic encoder and wheels in an omni-directional configuration. the middle plate houses the key electronic components such as the PCB, the MCU/MPU (Raspberry Pi Zero W), the drivers and the battery. The top plate accommodates the object detection system, where 6 IR sensors are placed in a circular arrangement to detect the object placed on top of the robot.

Design of the Arena : As discussed earlier, the system is expected to replace typical operations in an industrial environment such as the transportation of object/ movement of inventory in a warehouse. The arena is designed to accommodate all the operations of the system in one place. Fig.7 is a rendered image of the arena where the robots (depicted in blue) are at the home position. It can be seen that there are designated racks for an object, the software framework has a set goal location for a corresponding rack. The robots will plan a trajectory to this location using the proposed algorithms.

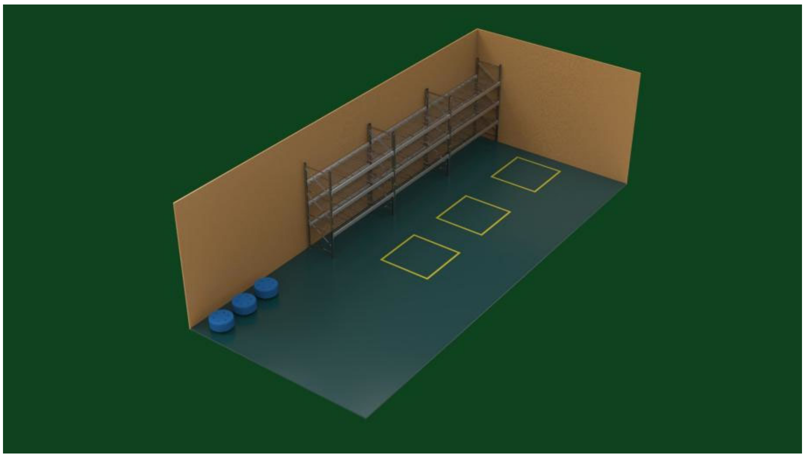
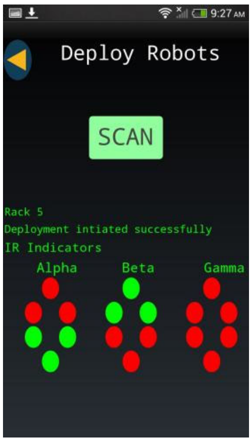


Figure 7 : Arena (Map)

User Interface/Application : The mobile application is the primary interface via which a user is expected to operate the system. For the collaborative system, a mobile application has been developed specifically to communicate with the user and also autonomously perform the task at hand. This application implements many IoT techniques to communicate with each or all three robots in any sequence to send and receive data. The software stack is interfaced with the application, such that the system is operational as soon as the user scans the QR/bar-code. Upon scanning the QR/Bar-code, the details such as the object location, dimensions and its destination are uploaded to the real-time database (firebase). The software stack then fetches this data to evaluate both the formation and the trajectory for the robots to follow to reach the destination.



Rack 5
Deployment intiated successfully
IR Indicators

Figure 8 : User Interface

Fig.8 is a representation of the user interface after the user has scanned the QR/Bar-code, the destination rack is displayed alongside the IR indicators which provide real-time feedback on whether the object is detected by the three robots or not. The robots will collaborate with each other and ensure that the object reaches the destination while maintaining formation. Since the trajectory is set for each robots at the start of the journey, the trajectory planner takes responsibility of keeping all the robot in their track through out the journey. In case the object falls mid-transport, the operation is stopped altogether.

4.2. *Results and Analysis*

Upon scanning the QR/Bar-code on the object, the system required the robots to transport the object to rack ”5”. The robots were tasked to plan a trajectory to this rack which is 1 (m) parallel to y axis w.r.t to world frame. In order to test and validate the proposed strategy and composite robot algorithm, for optimum results, the robot’s are already considered to be in formation. It is to be noted that the configuration of the robot is an omni-directional configuration, and it is assumed that the orientation of the robot are the same and doesn’t change throughout the trajectory. Therefore the robots are placed at the start or home point with zero degree change in their orientation with respect to origin and this does not change throughout. With this assumption we can provide the same set of wheel velocities to all the robots and can validate whether the formation is maintained or not based on the closeness of the robot’s actual velocity measured. The software stack is developed and configured in python 3.6.5. It was found that the total time it took by the composite robot to reach a distance of 1 m from the start location was 9.9 (s) at a maximum robot velocity of 0.1 (m/s). With a sample time of 50 (ms), there were a total of 198 measurement data obtained.

Figure 9 : Collaborative Material Handling

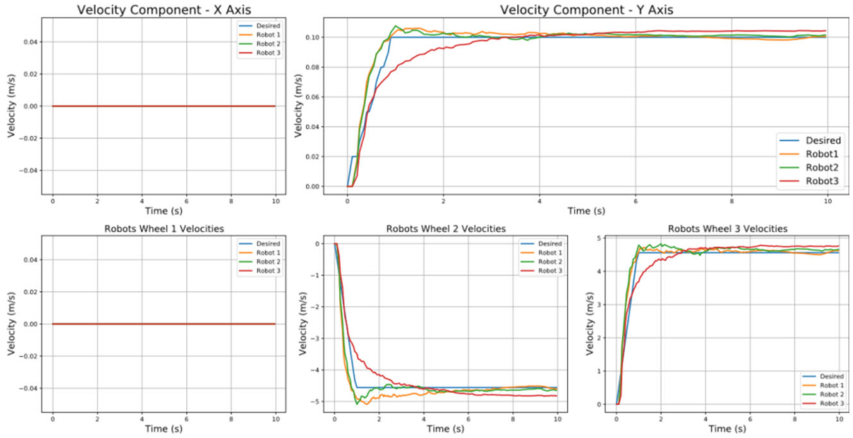


Figure 10 : Measurements of Composite Robot Velocity

The experiment was conducted to validate the composite robot algorithm and analyze the formation between the robots Fig.8 illustrates the performance of the composite robot. Table 1 supplements these results by providing the root mean squared error of the desired to the actual values obtained during the experiment.

As discussed earlier, the formation can be validated based on the individual robot's velocity. The composite robot's velocity component in the X axis for the path parallel to y axis is zero as expected and the velocity component in the Y axis for the robots ramp-up to the maximum velocity of 0.1 (m/s) and are similar compared to the desired velocity.

Table 1. RMSE Values of Measurements Obtained

Root Mean Squared Error (RMSE)			
	Robot 1	Robot 2	Robot 3
Robot Velocity (m/s)	0.0055	0.0050	0.0065
Wheel 1 Velocity (rad/s)	0.0	0.0	0.0
Wheel 2 Velocity (rad/s)	0.3785	0.2898	0.3195
Wheel 3 Velocity (rad/s)	0.2172	0.2638	0.2563

The wheel 1 velocities of the three robots in formation with the same orientation are expected to be zero and the wheel 2 and wheel 3 velocities are expected to equal in magnitude and the opposite sign. It can be inferred that the wheel 2 and 3 desired and actual wheel velocities for all the three robots are identical to each other.It was found that the formation was maintained until the destination is reached with only a small amount of drift, meaning the object was successfully transported to its destination. However, the system has to be further validated for various trajectories and feedback control to understand its robustness and reliability.

5. Conclusion

This work proposes a multi-robot formation strategy for holonomic robots and a composite robot algorithm for trajectory planning of the robots in formation. The formation of robots are dependent on the dimensions of the object that is expected to be transported. There are 4 different formations discussed, based on the object and robot dimensions, the algorithm evaluates a suitable formation for the robots and provide coordinates for the robots to arrive at. Also presented is a composite robot algorithm that considers the robots in the formation as a single entity meaning, providing the same set of wheel velocities from start to goal. Since the configuration of robots are holonomic and have the same design, kinematics and orientation, the formation is validated based on the closeness of robot's velocity. The robots planned a straight line trajectory between two points 1 (m) apart, parallel to y axis and reached the goal in 9.9 (s) with a maximum velocity of 0.1 (m/s). This was validated by measuring the wheel and robot velocities of the robots in formation.

References

1. Peshkin, Michael, and J. Edward Colgate. ”Cobots.” Industrial Robot: An International Journal (1999). https://doi.org/10.1108/01439919910283722
2. Vig, Lovekesh, and Julie A. Adams. ”Multi-robot coalition formation.” IEEE transactions on robotics 22.4 (2006): 637-649.
3. Dai, Yanyan, et al. ”A switching formation strategy for obstacle avoidance of amulti-robot system based on robot priority model.” ISA transactions 56 (2015): 123-134.
4. Balch, Tucker, and Maria Hybinette. ”Behavior-based coordination of large-scalerobot formations.” Proceedings Fourth International Conference on MultiAgent Systems. IEEE, 2000.
5. Yoo, Sung Jin, and Bong Seok Park. ”Connectivity preservation and collision avoidance in networked nonholonomic multi-robot formation systems: Unified error transformation strategy.” Automatica 103 (2019): 274-281.
6. De La Cruz, Celso, and Ricardo Carelli. ”Dynamic model based formation controland obstacle avoidance of multi-robot systems.” Robotica 26.3 (2008): 345-356.
7. Venkat Raaman M, Aravind S, Pavel R, Kuppan Chetty RM, Dhanraj JA. Design and Development of a General-Purpose Low-Cost Stewart Platform for Laboratory Teaching: A Mechatronics Approach. InAdvances in Mechanical Engineering 2021 (pp. 469-479). Springer, Singapore.
8. Zhang, Zhanying, and Ziping Zhao. ”A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm.” International Journal of Smart Home 8.3 (2014): 75-86.
9. Ramanathan KC, Ganapathy D, Dhanraj JA, Mohan M. Design And Development Of 6 Dof Bipedal Robot And Its Walking Gaits. International Journal of Robotics and Automation. 2021 Jan 1;36(1).
10. Peng, Jung-Hao, et al. ”Multi-robot path planning based on improved D* LiteAlgorithm.” 2015 IEEE 12th International Conference on Networking, Sensing and Control. IEEE, 2015.
11. Ryan, Malcolm. ”Constraint-based multi-robot path planning.” 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010.
12. Kuppan Chetty Ramanathan, Manju Mohan, Joshuva Arockia Dhanraj, Backward Motion Planning and Control of Multiple Mobile Robots Moving in Tightly Coupled Formations, Applied Computer Science, 17 (3), 60-72, 2021.
13. Hennes, Daniel, et al. ”Multi-robot collision avoidance with localization uncertainty.” AAMAS. 2012.
14. Alonso-Mora, Javier, Stuart Baker, and Daniela Rus. ”Multi-robot formation control and object transport in dynamic environments via constrained optimization.” The International Journal of Robotics Research 36.9 (2017): 1000-1021.
15. Lynch, Kevin M., and Frank C. Park. Modern Robotics. Cambridge UniversityPress, 2017.