Multi-Mobile Robot Motion-planning Methods in Smart Warehouse Environment

Prajwal Prabhu and Abhra Roy Chowdhury*, SMIEEE

Abstract— **This paper studies the feasibility of Autonomous mobile robot's motion planning solutions for smart warehouses in the Industry 4.0 environment. Motion planning algorithms/techniques for a single autonomous mobile robot (AMR) and its shortcomings in the warehouse environment are discussed. Further, potential of multi-agent AMR's motion planning solutions i,e. Leader follower, bio-inspired modelling and multi-robot collision avoidance solutions in enhancing the warehouse's productivity are discussed. Using evaluation matrix, comparative performance study of motion planning algorithms based on industrial standard obstacle avoidance and detection is presented with RRT algorithm (93.3%) and behavior based multi-robot collision avoidance system (96.7%) having highest performance efficiency in single and multi-robot AMR systems respectively.**

Keywords— **Autonomy, Mobile robots, Motion planning, Multi agent systems, Warehouse, Smart manufacturing, Industry 4.0**

I. INTRODUCTION

In business operations, the warehouse industry plays a vital role in ensuring the smoothness of the overall business operation thus determining the overall setup efficiency. A high amount of flexibility is seen by incorporating the human and machine in the warehousing industry that lead to overcoming of the mentioned limitations. All possible limitations affecting the system performance have to be considered for a successful operation of the AMR [1] before incorporating the robotic solution into the warehouse. Human beings alone in such an environment would be a huge positive factor because of the cognitive and fine motor skills that can accommodate any of the arising operational changes. Autonomous mobile robots (AMR) are known for their unique ability [2] to navigate in an uncontrolled environment with a higher-level understanding via sensors, decision-making via artificial intelligence, and more [3, 4].Due to the versatile advantages, AMR's finds itself as a critical component in the industry 4.0 for smart factories as it will provide autonomy in terms of the motion planning without depending on pre-planned routes and can suitably adapt itself to the warehousing environment without employing sensor for robotic system guidance [5].Recent advancements in robotics have led to highly efficient methods for safe navigation and accurate perception of the surrounding environment at high velocities. Using a single autonomous robot can be used in small warehouses and can have advantages like reducing the warehouse traffic, but however is limited for a warehouse which is very small. Further, maneuvering in complex and highly unstructured environments will be very difficult and may burden the overall system in terms of the weight and the overall performance [6,7]. The use of a multi robot autonomous [8] solution can be further looked at, which will optimize the performance of the warehouse by using a suitable SLAM approach. The use of multirobot systems has led to better communication among the systems as the local minima of the system is always allocated for the task thereby improving the overall system efficiency [9]. While dealing with multiagent autonomous robotics some of the most evident problems that are seen are the reduced run time, sophisticated motion planning for an unstructured environment expecting continuous change in the floor plan and training of the robot for maximum real world scenario to ensure the system is aware of all the possible constraints. This paper looks at some possible motion planning solutions [10, 11] in terms of the single agent autonomous mobile robot and potential of multi-agent robots [9, 13]. Further, it looks at some motion planning solutions in terms of multiagent robotics systems and speaks about the possible limitations seen in the study. The section I gives a brief introduction about the current warehousing trends, need for autonomous mobile robotics in the warehousing industry and factors responsible for emerging multiagent AMR as the best solution to maximize the warehouse productivity. The section II looks at possible single agent AMR motion planning solutions that can be incorporated into the warehouse. The section III explores possible multiagent solutions that can be incorporated into the warehouse to improve the warehouse productivity. The section IV gives the conclusions and the future implementations for AMR's to improve the warehouse productivity.

Previous work: Robot motion planning has always been one of the most fundamental problems in warehouse navigation. For single AMR system A* algorithm and RRT algorithms are popular. In motion planning problems, the A* algorithm is a well-known approach for choosing the optimum path between two points in a finite period of time. The RRT family algorithm, on the other hand, converges to a collision-free path using random sampling from the environment. The RRT* algorithm has significant advantage over RRT as it can reroute and store cost value of each node.

This paper was first submitted on April 3, 2021.

* Corresponding Author is the Assistant Professor, Division of Mechanical Sciences Centre for Product Development and Manufacturing, Indian Institute of Science, Bangalore e-mail: abhra@ iisc.ac.in

Prajwal Prabhu and Abhra Roy Chowdhury are with the Division of Mechanical Sciences Centre for Product Development and Manufacturing, Indian Institute of Science, Indian.

Contribution:

- This work focuses on implementation of algorithms for both single and multi-robot AMR systems for warehouse-based path planning.
- The authors have presented comparative study and advantages of both single and multi-robot AMR systems along with their efficiency respectively allowing better decision while choosing path planning algorithm for warehouse-based path-planning.

II. MOTION PLANNING FOR SINGLE AGENT AMR

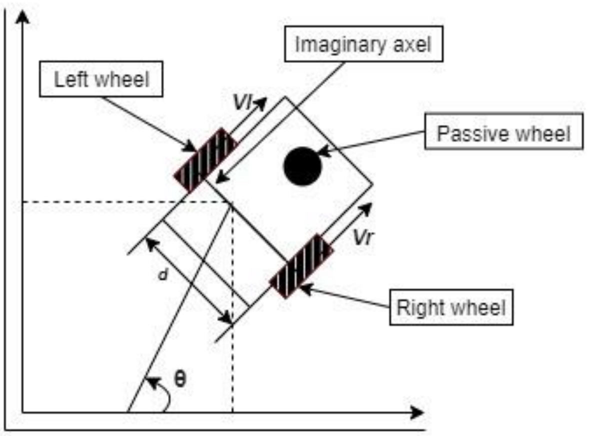

Fig. 1. Generalized Differential Drive robot model.

The generalized model for the Differential drive robot is shown in figure 1. The differential drive robot model is used to look at the different motion planning scenarios for the warehouse. By varying the velocities of each of the wheels, the robot trajectories can be varied. The wheel velocity for the robot is given by equation 1,2.

$$Vr = \omega\left(R + \frac{l}{2}\right) \quad (1)$$

$$Vl = \omega\left(R - \frac{l}{2}\right) \quad (2)$$

Further to find the trajectory for the robot motion the following relations were used as per the equation 3,4,5.

$$\dot{x} = \frac{R}{2} * (Vr + Vl) * \cos(\theta) \quad (3)$$

$$\dot{y} = \frac{R}{2} * (Vr - Vl) * \sin(\theta) \quad (4)$$

$$\dot{\theta} = \frac{R}{l} * (Vr - Vl) \quad (5)$$

where,
Vr = Right wheel velocity
Vl = Left wheel Velocity
R = Radius of the wheel
l = length of the robot base
θ *= Orientation of the robot.*
ω *= Angular acceleration of the robot*

We define the position and the orientation of the system in terms of the velocity of the robot as per the unicycle model as per the equations 6,7,8.

$$\dot{x} = V\cos(\theta) \quad (6)$$

$$\dot{y} = V\sin(\theta) \quad (7)$$

$$\dot{\theta} = \omega \quad (8)$$

Since equations (6), (7), (8) represent the position and the orientation for the differential drive robot, the equations (3,(4),(5) can be equated and a relation for the velocity and the angular acceleration can be found as equation

$$v = \frac{R}{2} * (Vr + Vl) \quad (9)$$

$$\omega = \frac{R}{l} * (Vr - Vl) \quad (10)$$

The dynamics for the robot was derived using the Lagrangian formulation and the torques for both the wheels are derived as per the equation 11,12.

$$\left(m + 2I\omega / R^2\right) * v^2 - m * d * \omega^2 = \frac{1}{R} * (\tau r + \tau l) \quad (11)$$

$$\left(I + 2 * \left(\frac{l^2}{R^2}\right) * I\omega\right)\omega + m * d * \omega * v = \frac{l}{R} * (\tau r - \tau l) \quad (12)$$

A. *Motion Planning using RRT (Rapidly exploring Random Tree) Algorithm*

Traditionally, in a warehousing scenario, we can expect the map of the warehouse to continuously change due to either the needs of the warehouse or the constant technological upgrade. So, there will be a need for using algorithms that are generally suitable for dynamic models, ensure the correctness of the data thrown by the algorithm and feasibility for a real time implementation of the motion planning. Generally, the RRT algorithm is used in such a scenario because of its properties mentioned above. Further, by the use of the RRT algorithm, we can check the feasibility of achieving the target for all the possible scenarios. The traditional RRT algorithm samples the input to the robot, however an alternate solution can be to sample the input to the controller. Figure 2 shown above gives a brief description for the motion planning for the robot done using RRT algorithm based on the vehicle dynamics where the dotted blue lines.

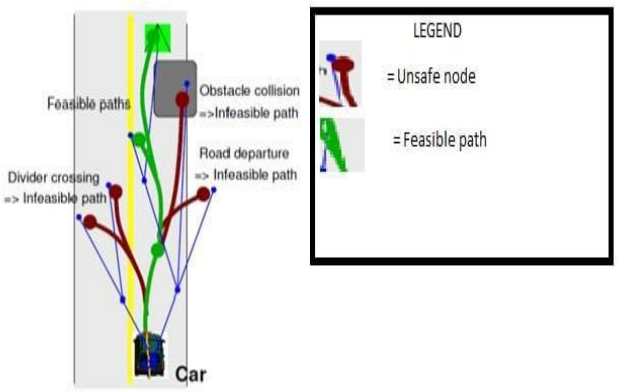

Fig. 2. The motion planning using the robot dynamics model, where paths are propagated and then evaluated for feasibility.

curves indicate the tree generated for the environment, considering all the possible routes. The robot before starting off the motion will look at the possible paths to achieve the target and will select the best feasible path based on the target available. The nodes marked in the red indicate the algorithm saying that it would not be a feasible path as there would be collision / moving out of path / collision with an obstacle. The green lines in the figure indicate the path taken by the robot to traverse from the start to its goal. The RRT algorithm used for the motion planning has been characterized with following steps as per the table 1. This algorithm has much more features incorporated into it than the traditional algorithm, such as lines (5,8,12,24). The line 5 looks at the aspect of biased based sampling, which is mainly used to increase the probability of a feasible trajectory. The bias generated in such a scenario will be purely dependent on the situation. The samples are taken in 2D, and they are used to form the input to the steering controller. The sample (x-sample, y-sample) is taken randomly but has some parameters to bias its location/shape, i.e.,

$$\begin{bmatrix} Xsample \\ Ysample \end{bmatrix} = \begin{bmatrix} Xo \\ Yo \end{bmatrix} + r * \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

$$with\ r = \sigma r * \left| n_r \right| + ro\ and\ \theta = \sigma_\theta * n_\theta + \theta_0$$

where nr and nθ are random variables that have Gaussian distributions, σr and σθ give the 1-σ values of the radial and circumferential direction, r0 and θ0 are the offsets, and (x0, y0) is the center of the Gaussian cloud. The uniqueness of this approach is that by varying the bias values based on the situational information, the planner can generate various maneuvers including lane following, The line 8 is provided as a safety constraint where the safety parameter for the robot can be successfully defined if the robot is able to maneuver itself and avoid the obstacle successfully or stop if it's close to colliding with an unknown obstacle. Before a feasible trajectory is generated, the tree would grow based on the exploration heuristic and would connect the nearest node to the tree, thereby generating all the possible trajectory for a given start and finish. During the travel from a given start and the finish points, the path generated would be such that it is obstacle dependent and safety would be given the utmost priority. Upon the movement from the start node to reach its destination, the robot would go to its nearest waypoint which would be deemed safe and there would be no possibility of a collision. The waypoints for the successive movement to reach the required goal would be generated through the sensors, looking at the safety constraints and the possibility of the robot colliding with the obstacles to reach its final required goal. The line 12 of the algorithm looks at the risk assessment for the plan where it looks at, whether the trajectory generated would collide with any obstacles or not. Suitably, a penalty would be incurred for the robot steering of the desired trajectory. Further, there will be strict ensuring that the robot stops a few meters before the obstacle. The line 12 looks at the concept of an unsafe node. The RRT algorithm will generate a trajectory based on the start and the finish points, considering all the possible cases. However, if the generated trajectory may cause a collision, it will not reach such a trajectory and such a thing will be termed as unsafe and will be discarded. The line 24 looks at the concept of lazy check. In a warehouse like scenario there will be continuous technological upgrade which will require a dynamic monitoring of the environment. In this algorithm, whenever the perceived environment is updated, feasibility of each edge is checked based on the current situation. In a dynamic environment like a warehouse, there would be a large tree that would check the feasibility of each edge.

TABLE 1

RRT ALGORITHM USED FOR MOTION PLANNING

	Algorithm 1 RRT – based planning algorithm
1.	Receive current vehicle state and environment
2.	Propagate states by computation time limit
3:	Repeat
4:	Take a sample for input to the controller
5:	Select the node in tree using heuristics based on the feasible condition
6:	Propagate from selected node to sample until the vehicle stop
7:	Add branch node on the path
8:	**If** propagated path is feasible with the drivability map **then**
9:	Add sample node and branch node to the tree
10:	else
11:	**If** all the nodes are feasible **then**
12:	Add branch nodes to the tree and mark them as unsafe
13:	**end if**
14:	**end if**
15:	**For** each newly added node v **do**
16:	Propagate to the target
17:	**If** propagated path is feasible with the drivability map **then**
18:	Add path to the tree
19:	Set cost of the propagated path as upper bound of the cost to go at v
20:	**end if**
	End for
21:	**Until** the time limit is reached
22:	Choose the best feasible trajectory in the tree, and check feasibility with the latest drivability map
23:	**If** the best trajectory is infeasible **then**
24:	Remove the infeasible portion from the tree, go to line 24
25:	**end if**
26:	Send the best trajectory to the vehicle controller **Until vehicle reaches goal**

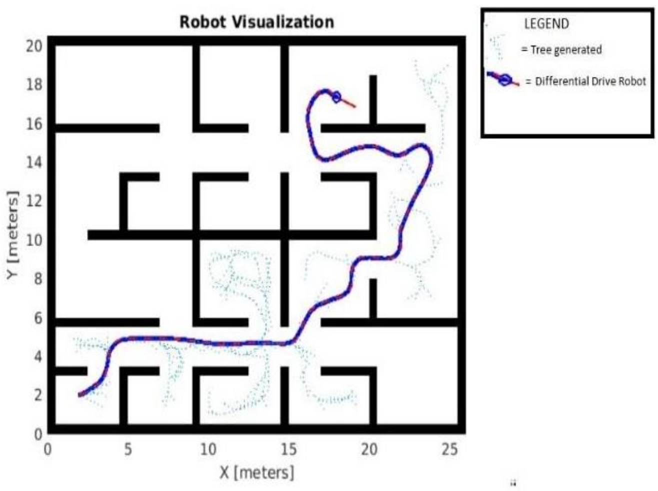

Fig. 3. The RRT algorithm result achieved.

The concept of a lazy check would ensure the algorithm would focus on the growing tree, while ensuring the feasibility of the executed trajectory in the latest drivable map. The scenario for the parking in a complex environment was generated using MATLAB and the result was validated for the proposed algorithm and the results were as per the figure 3. The robot was able to successfully navigate itself through a complex environment and reach its goal of [19,17] from the starting position of [1,2] and the robot was able to generate tree for all the possible and was able to maneuver itself through all the obstacles and generate an optimal path for the trajectory based on the proposed algorithm. Further, the differential drive model was used for the modelling of the robot.

The dotted blue lines shown in the figure shows the tree that was grown for the environment to find the optimal path for the system. The dark blue line shown in the figure is the optimal path taken by the robot [shown as blue dot] to reach the goal. The robot algorithm, however robust, will not be suitable when there are dynamic obstacles because of the fixed step size. So, there may be a limitation for the usage of the robot for such a scenario. There may be a case of using two planners to achieve a robust algorithm for achieving a more robust system for trajectory planning in a complex environment.

B. Real Time Obstacle Detection and Motion Planning using VFH (Vector Field Histogram) Technique

Vector field histogram technique depends on the system measurements, modelling errors. This method will have a statistical representation of the robot environment as a histogram plot and will consider the dynamics of the robot and the shape / orientation of the robot to give steering commands suitably. This technique has more reliability and is more robust as it will give the real time accurate description of the robot environment and all the obstacles present in it. This can be used in cases where there are densely populated obstacles in the environment.

The figure 4 shown gives a pictorial representation of the environment from the robot's point of view where the obstacles surrounding the robot are earmarked as histogram / polar plots as shown in the figure 5 where the blue sticks indicate the lidar sensor scanning the surrounding environment. The histogram plots in its bare essence gives details about the statistical representation of the obstacles depending on the position of the obstacle with respect to the robot frame.

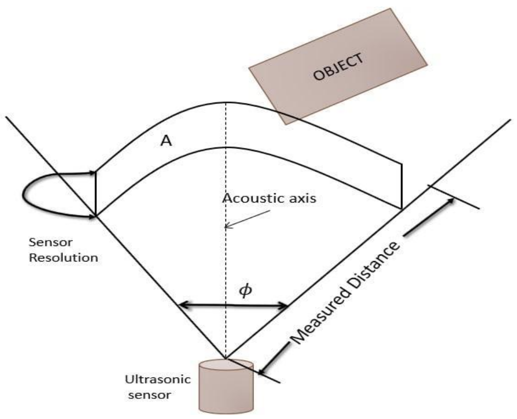

Fig. 4. The representation of the obstacle detection for the environment using VFH technique.

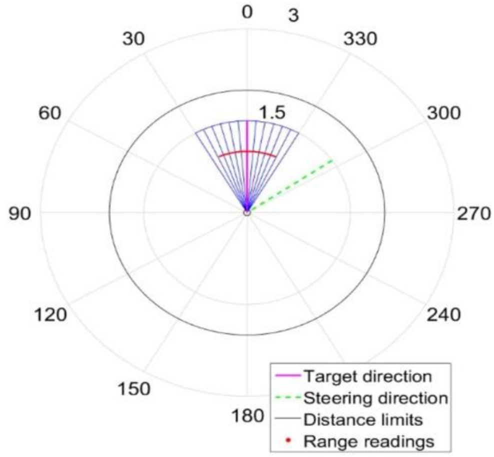

Fig. 5. The polar histogram plot for the systems giving the details of the obstacle occupancy of the system.

Such systems are typically suited in scenarios where data from the sensor is inaccurate and will allow multiple sensor fusion.

1. The 1D histogram will be continuously updated through laser rangefinders / ultrasonic sensors.

2. The polar histogram gives the momentary location of the robot around the environment.

3.The target selection is done based on which point will take the robot closer to the goal.

Once the point is selected that leads to the target, the robot will be steered accordingly to the goal. Since the objective of this method is for obstacle avoidance there may be a scenario where the robot may not reach the required goal, if there is an obstacle close by. The results for the simulation are shown MATLAB simulation results as per the figure 6. The nodes shown as the 'x' indicate the waypoints that the robot will follow in order to reach the required goal. The robot is marked as a circle and the blue sticks shown in the figure is the lidar sensor attached to the robot sensing the environment to ensure the obstacle avoidance for the robot.

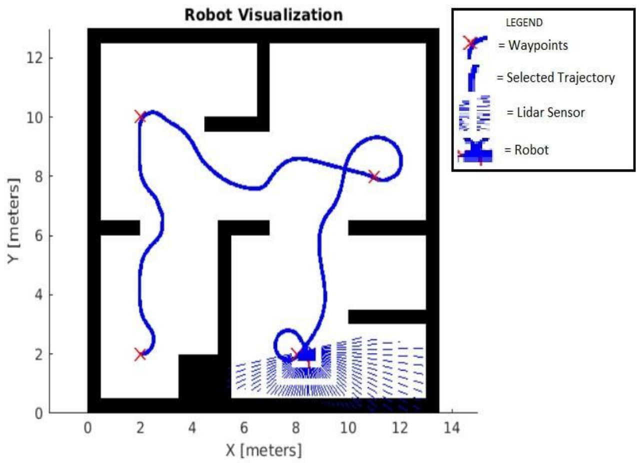

Fig. 6. The path traced by the robot in a complex environment using the VFH technique.

C. Types of Multi Agent Robotic Systems

Table 2 **[5]** shows the typical classification for the multiagent system based on the number of agents and the scope of each classifications. This decides on how the system is modelled, choice of connectivity, number of inputs to the system. Typically, teams are selected keeping in mind the objective to maximize the local goal with the help of these individual agents. One of the most successful ways to improve the team's performance would be to make the individual agents compete against each other, alternatively there may be a case of looking at optimizing the local behavior keeping the global reward in mind. In formation there is always a cooperative interaction amongst the agents with well-defined objectives. A swarm generally refers to a group of similar agents that displays emergent behavior arising from local interactions among the agents.

TABLE 2
CLASSIFICATION OF MULTIAGENT SYSTEMS

Type	Scope	Size
Team	Typically, a small group, with each agent optimization In cooperative manner	≤ 10
Formation	Each agent is assigned a specific task	Typically, < 10
Swarm	Typically, large group of dispensable agents: global capability arises from emergent behavior	large

D. Synchronization with Leader Following Model for Multi Agent Autonomous Systems

Typically, a warehouse will be a large area, which dictates a huge number of multiagent systems is required in the warehouse to ensure the smooth operation of the warehouse. To decide the control strategy, it may be essential to define a leader who all the multi agent systems follow. The motion of the leader is known prior or is defined by a separate dynamic. Alternatively, the trajectory for the leader can be found through optimal control / motion planning algorithms. The remaining agents are controlled indirectly through the interaction of the neighbors or through the interaction with the leader.

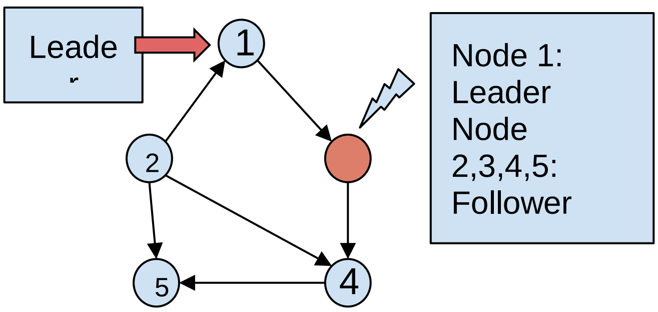

Fig. 7. Generic representation for a leader follower system.

The figure 7 shows the generic representation for the leader follower system. The leader and the follower will communicate with each other and suitably it will manipulate itself through the obstacle filled environment. The formation and orientation of the leader-follower pair will be purely dependent on the user. The problem of tracking the trajectory of the virtual leader or the overall group with nonlinear dynamics can be addressed simultaneously by synchronizing the neighboring agents. By doing this there will be a smaller synchronization error. The errors seen in such a scenario will be purely due to the modelling errors or the presence of bounded dependencies. The simulation results for the proposed system is shown in figure 8. The figure 8 shows how the leader follower system successfully manipulates through a system of obstacles and reaches the final goal where the blue circle are the robots, where robot 3 is selected as the leader and the black polygons are the obstacles for the system. The system would use a behavior-based model to maneuver and some of the generalized behavior rules are described in the behavior based multi robot collision avoidance topic.

E. Bio Inspired Non cooperative Multi-robot Herding

When modelling the autonomous multi agent robot systems it is highly difficult to achieve an optimal control strategy, so as an alternative the concept of bio inspired robotics can be explored. The bioinspired robotics use the concept artificial intelligence, use the examples of swarms, groups where use both mind and the body and the concept is used to developed suitable algorithm used from nature in the robotics and the

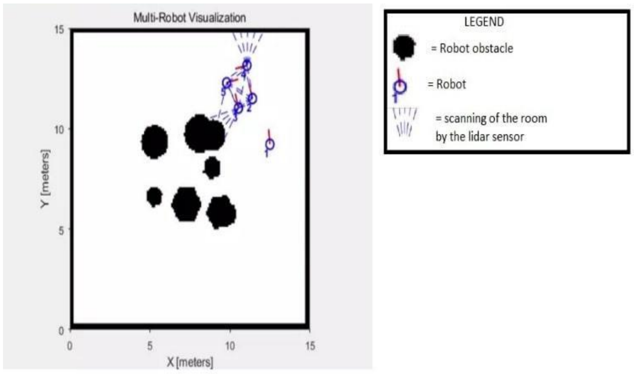

Fig. 8. The leader follower system manipulating through the obstacles and final orientation after manipulation.

concept is used to develop suitable algorithms used from nature in robotics. The bio-inspired concept of non-cooperative sheep being herded by dogs through repulsive potential can be looked at as a suitable concept to model a multi-robot system. The sheep-like agents, which may be biological or robotic, respond to the presence of the dog-like robots with a repelling potential field common in biological models of the behavior of herding animals. The unicycle robot model is used to map the dynamics of the dog-like robot and the sheep to model the positional relationship between the two agents as shown in figure 9.

The unicycle model for the robot is given in figure 9, for a nonholonomic vehicle, with respect to local reference frame Q relative to the global base frame B. Its forward velocity v defines the local qx direction, the orientation relates the heading qx to the global bx.

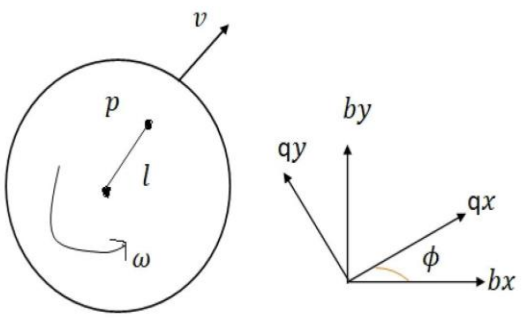

Fig. 9. Ideal Unicycle Model for a Non-Holonomic Vehicle.

The modelling of the systems is done as Single Sheep with Two Dogs model, or Single Sheep with m Dogs model. The dynamics of the single sheep and m dog model are shown in figure 10. The model uses the position and the angular orientation of the dogs with respect to the sheep to describe the dynamics of the system to reach the final orientation as required. The fundamental inference of this model can be said as the dogs are fixed on some circle of radius r relative to the herd, which limits the initial configurations of the dogs relative to the sheep. A tracking controller for the dogs that allows them to start anywhere in the environment and converge upon this configuration.

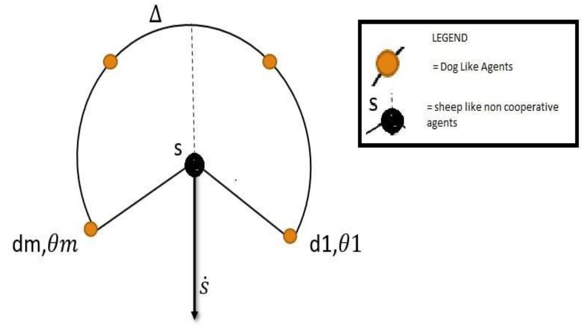

Fig. 10. Ideal unicycle model for a single sheep and m dog model.

The unicycle model is used to describe the dynamics for the system. The angular orientation for the dogs can be given by the equation 13.

$$\theta_j = \phi + \pi + \Delta_j \tag{13}$$

$$where,\ \Delta_j = \Delta \frac{(2j - m - 1)}{(2m - 2)} \tag{14}$$

m= number of dog agents
J=1,2,3,4,5……….
Δ = relative position of each dogs with respect to other

The dynamics for the dogs is given by equation 15. By defining the orientation of the dogs in terms of ϕ and Δ along some radius as per equation 13 and restricting the dog kinematics as per equation 15, the mapping is done with respect to the linear and angular velocity of the unicycle model. The fundamental inference of this model can be said as the dogs are fixed on some circle of radius r relative to the herd, which limits the initial configurations of the dogs relative to the sheep.

$$dj = \dot{s} + r\left(\dot{\phi} + \dot{\Delta}j\right) * \begin{vmatrix} \sin(\phi + \Delta j) \\ -\cos(\phi + \Delta j) \end{vmatrix} \tag{15}$$

The results for the bio-inspired modelling for two dogs and one sheep model is simulated using MATLAB and the results achieved were as per the figure 11. The original paper results were about the model in a structured environment. As an addition to the proposed work the results speak about the same model in an unstructured environment. The model starts at points [12,2] and it was able to successfully maneuver itself out of the maze. The path taken by the maze is shown in triangles and the robots are marked as a circle in blue color. The triangle path taken by the robot indicates the 2 dogs guiding the non-cooperative sheep robot along the complex environment. There is scope of increasing the number of dogs like agents in the same unstructured environment which may be typically seen in a warehousing scenario that can be looked at in future. Further upon increasing the number of sheep and the dog like agent the controller complexity would increase as there should be proper assigning of the control between multiple dogs and the sheep like agents. Further the possibility of looking dogs as a static communication posts to maneuver the robot can be looked in the future implementation

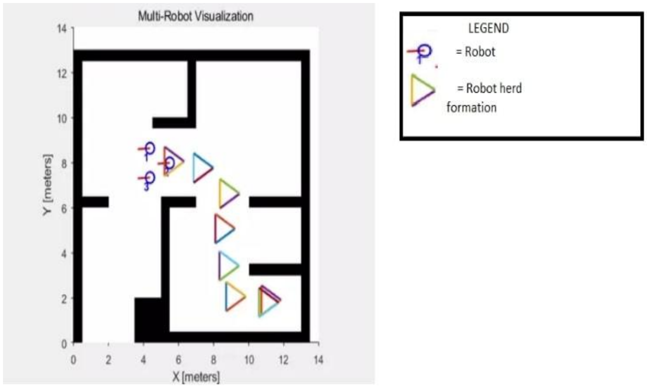

Fig. 11. The bio-inspired modelling result in a complex environment.

F. Multi-robot Collision Avoidance Based on the Behavior Modelling

The critical thing that makes autonomous mobile robots successful is that there will be flexibility provided in terms of motion planning. However, there is a need to ensure that the robots used are self-aware of their environment so that there is a safety factor included as one of the key features. In such a scenario it is necessary to ensure that while using multiple robots there is collision avoidance incorporated into it so that it is warehouse ready. Most of the traditional robot collision avoidance methods assume perfection in the robot sensing and localization and use overhead cameras. To do a proper collision avoidance the robot has to be able to estimate the pose in the environment correctly and also be able to correctly estimate the pose of other robots and be able to deal with uncertainty and possible actions of other robots. There are different approaches used to solve the problems of collision avoidance. Alonso-Mora et al. (2015a), speaks about collision avoidance algorithms for multiple unmanned aerial vehicles (UAVs) where a centralized and decentralized convex optimization approach is explained, and the system is integrated with two UAVs flying in close proximity to a human. Further there are approaches defined by Bruce and Veloso (2006) where a centralized controller is used for collision avoidance. A suitable method that can be used is the behavior-based control for the collision avoidance. One of the primitive advantages of a behavior-based collision system would be that with an exhaustive behavior set the robot would be able to manipulate through the environment of different complexities and would be successful in a warehousing scenario. In a behavior-based system a generic approach would be to define how the robot has to manipulate through the different waypoints and define the behavior for the robot when another agent is approaching it to avoid the collision. The generalized behavior used to describe the collision avoidance for the system is given as

1. Avoid: - The robot should drive around other robots or the obstacles. When there is no robot around, then this

behavior would be complete, and the robot would follow its desired path.

2. Exchange: - If there is an impending collision between 2 robots head on then the robots should pass each other, and they are each other's partners. When there are no partners in front of the robots then this behavior would be completed

3. Wait to go through: - If there is any impending collision which can't be avoided then one of the robots has to stop and wait till the robot collision is averted and then move further to its desired goal.

4. Go through: - If there are any collisions that may happen through the sides i.e the robot going through the intersection, the other robot has to wait till the robot has passed and then move along.

The generalized algorithm used for multi-robot collision avoidance systems that is based on behavior of the system is given by table 3.

TABLE 3
GENERALIZED ALGORITHM FOR MULTIROBOT COLLISION AVOIDANCE

	Algorithm: Behavior based Multi-Robot Collision Avoidance
1.	Start from home position.
2.	Scan the surrounding complex environment for future steps.
3.	If obstacle is found through the scanning, go through the generalized behavior defined **Case1:** If possible, try to drive around the obstacle. **Case2:** Else go back to the previous position and find the new route to travel.
4.	In case of robot-robot collision: **Step1:** Try to avoid the robot. **Step2:** Wait for the robot to move and then move.
5.	In case unavoidable robot collision at the interaction: **Step1:** Wait till the other robot has passed. **Step2:** Move along the defined path.
6.	In the case of provision for the robot to exchange: **Step1:** The robot has to wait and ensure that the other robot has to be close enough for successfully transfer. **Step2:** Exchange part of the behavior as defined.
7.	In case the robot reached the goal. **Step1:** Stop at a defined location. **Step2:** Wait for further inputs.

The results for the multi robot collision avoidance simulation shown in the figure 12 are based on the generalized behavior rules defined where the robot is marked as the blue circle and the blue lines indicated in the figure is the lidar sensor attached to the robot sensing the environment. The below figure shows the multi robots in a complex environment. As per the behavior-based model, if there is a possible collision the robot will try to stop or move away from the robot [or the wall] to avoid the collision. Further the rules defined to obtain the result can be updated as per the requirements to match the complexity of the environment.

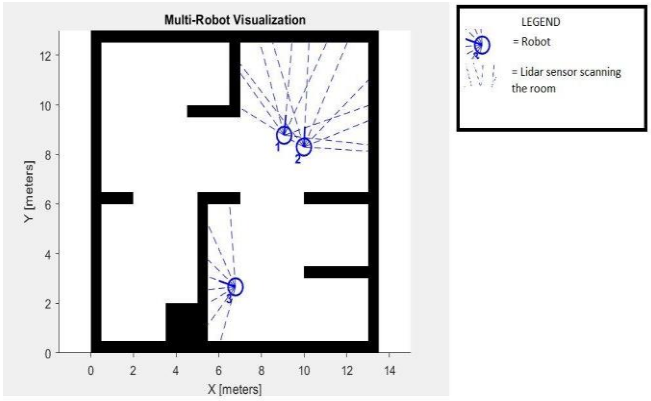

Fig. 12. Multi robot collision avoidance in complex unstructured environment

Further there can be a case of including the speed, momentum, distance of deceleration in the modelling of the behavior to ensure the safe operation of the robot.
The evaluation metric for the motion planning algorithm was checked based on how effectively the robot was able to detect the obstacles and avoid the same.

Let **A** = no. of times the robot was able to detect and avoid the obstacles
B = number of iterations run for the algorithm
Performance efficiency = $\frac{\boldsymbol{A}}{\boldsymbol{B}}$

TABLE 4
COMPARATIVE STUDY OF MOTION PLANNING ALGORITHM

Motion planning algorithm	**No. of times the robot was able to detect and avoid the obstacles**	**No. of iterations Run**	**Performance Efficiency (%)**
RRT algorithm	28	30	93.3
VFH algorithm	25	30	83.3
Leader follower algorithm	27	30	90
Bio-inspired modelling	28	30	93.3
Behavior based multi robot collision avoidance	29	30	96.7

From the iterations run it was seen that the RRT algorithm had the best performance when compared to the VFH algorithm and had an efficiency of 93.3%. In the multi robot system it was seen that the behavior based multi-robot collision avoidance system was the best suitable motion planning algorithm because of the exhaustive behavior set. Based on the problems specified in the previous sections and particular applications the performance parameter can be suitably normalized.

III. Conclusion & Future work

The use of autonomous mobile robots is an upcoming trend in the warehousing industry. This paper reviews some of the possible motion planning scenarios that can be incorporated for the warehousing scenario. A single autonomous robot would be sufficient in a small warehouse. However, as the overall size of the warehouse increases, using a single autonomous robot would not be feasible and there exists a need to use multiple autonomous mobile robots. This paper looks at some of the motion planning solutions for autonomous mobile robots that can be incorporated into warehouses. The use of the RRT algorithm is mainly useful for a complex scenario like a warehouse. However, there are certain limitations like fixed step size for the algorithm which makes it unsuitable for a scenario like dynamic obstacles. Further VFH algorithms can be used for obstacle avoidance which is more robust as there will be statistical representation of the obstacles and any possible errors would be only due to sensor or through modelling errors. The concepts of using bio inspired modelling will help in creating a more complex team of robots, the model derived will have a biological complexity that is evident from nature. The concept of multi agent robots being herded is an interesting scenario where multiple robots are used to guide the target robot to the goal using simple models. However, there are future scope in the model where the concept of multiple agents guiding the multiple targets to the goal can be looked at. There will be communication issues when using such models that may limit the convergence of the models. The herd of robot agents can be controlled depending on the relative position of the robot followers with respect to leaders. The use of sensors can be looked at as a protective guide in case of the failure of the model and can also be looked at for future cases. This paper has looked at the different motion planning scenarios for the autonomous robots for single and the multi-robot for a warehousing scenario that can be used to improve the warehouse productivity. Further the performance analysis for the all of the proposed algorithm was checked keeping 2 important factors, namely 1) how the robot was able to detect the obstacles and 2) how effectively it was able to avoid the detected obstacles. In each of the cases the robot was successfully able to detect and avoid the obstacles for the complex scenarios of the warehouse. For future scenarios there may be a case to look at the concept of least action to look at for the motion planning systems which looks at using the minimum energy case to travel between the start and the end point by minimizing the Lagrangian. This method can further be discretized time instant for making it applicable for dynamic obstacles thereby further increasing the effectiveness of the motion planning.

Acknowledgment

This project is funded by Samarth Udyog, Common Engineering Facility Centre (CEFC) CPDM, IISc, Bangalore; Department of Heavy Industries (DHI), Govt. of India.

References

[1] Liaqat, A., W. Hutabarat, D. Tiwari, L. Tinkler, D. Harra, B. Morgan, A. Taylor, T. Lu, and A. Tiwari. "Autonomous mobile robots in manufacturing: Highway Code development, simulation, and testing." *The International Journal of Advanced Manufacturing Technology* 104, no. 9 (2019): 4617-4628.

[2] Agarwal, Divya, and Pushpendra S. Bharti. "Nature inspired evolutionary approaches for robot navigation: Survey." *Journal of Information and Optimization Sciences* 41, no. 2 (2020): 421-436.

[3] Chung, Soon-Jo, and Jean-Jacques E. Slotine. "Cooperative robot control and concurrent synchronization of Lagrangian systems." *IEEE transactions on Robotics* 25, no. 3 (2009): 686-700.

[4] Rashid, Razif, N. Perumal, I. Elamvazuthi, Momen Kamal Tageldeen, MKA Ahamed Khan, and S. Parasuraman. "Mobile robot path planning using Ant Colony Optimization." In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pp. 1-6. IEEE, 2016.

[5] Chung, Soon-Jo, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. "A survey on aerial swarm robotics." *IEEE Transactions on Robotics* 34, no. 4 (2018): 837-855.

[6] Pierson, Alyssa, and Mac Schwager. "Bio-inspired non-cooperative multi-robot herding." In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1843-1849. IEEE, 2015.

[7] Claes, Daniel, and Karl Tuyls. "Multi robot collision avoidance in a shared workspace." *Autonomous Robots* 42, no. 8 (2018): 1749-1770.

[8] Kuwata, Yoshiaki, Gaston A. Fiore, Justin Teo, Emilio Frazzoli, and Jonathan P. How. "Motion planning for urban driving using RRT." In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1681-1686. IEEE, 2008.

[9] Mehdi, Syed Bilal, Venanzio Cichella, Thiago Marinho, and Naira Hovakimyan. "Collision avoidance in multi-vehicle cooperative missions using speed adjustment." In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2152-2157. IEEE, 2017.

[10] Mulgaonkar, Yash, and Vijay Kumar. "Autonomous charging to enable long-endurance missions for small aerial robots." In *Micro-and Nanotechnology Sensors, Systems, and Applications VI*, vol. 9083, p. 90831S. International Society for Optics and Photonics, 2014.

[11] Su, Housheng, Xiaofan Wang, and Zongli Lin. "Flocking of multi-agents with a virtual leader." *IEEE transactions on automatic control* 54, no. 2 (2009): 293-307.

[12] Grosse, Eric H., Christoph H. Glock, and W. Patrick Neumann. "Human factors in order picking system design: a content analysis." *IFAC-PapersOnLine* 48, no. 3 (2015): 320-325.

[13] Kousi, Niki, Spyridon Koukas, George Michalos, and Sotiris Makris. "Scheduling of smart intra–factory material supply operations using mobile robots." *International Journal of Production Research* 57, no. 3 (2019): 801-814.

[14] Beul, Marius, David Droeschel, Matthias Nieuwenhuisen, Jan Quenzel, Sebastian Houben, and Sven Behnke. "Fast autonomous flight in warehouses for inventory applications." *IEEE Robotics and Automation Letters* 3, no. 4 (2018): 3121-3128.

[15] Salzman, Oren, and Dan Halperin. "Asymptotically near-optimal RRT for fast, high-quality motion planning." *IEEE Transactions on Robotics* 32, no. 3 (2016): 473-483.

[16] Borenstein, Johann, and Yoram Koren. "The vector field histogram-fast obstacle avoidance for mobile robots." *IEEE transactions on robotics and automation* 7, no. 3 (1991): 278-288.

[17] Babinec, Andrej, Martin Dekan, František Duchoň, and Anton Vitko. "Modifications of VFH navigation methods for mobile robots." *Procedia Engineering* 48 (2012): 10-14.

[18] Murray, Richard M., Zexiang Li, and S. Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[19] Chris Batten , Control for Mobile Robots , Maslab IAP Robotics Course,2005.