

# Path Planning And Trajectory Control Of Collaborative Mobile Robots Using Hybrid Control Architecture

Trevor Davies, Amor Jnifene

Department of Mechanical Engineering, Royal Military College of Canada

PO Box 17000, Station Forces, Kingston, Ontario, Canada, K7K 7B4

Phone: (613) 541-6000 ext 6097, Fax: (613) 542-8612, Email: trevor.davies@rmc.ca, amor.jnifene@rmc.ca

## ABSTRACT

This paper presents the development and implementation a hybrid control architecture to direct a collective of three X80 mobile robots to multiple user-defined waypoints. The Genetic Algorithm Path Planner created an optimized, reduction in the time to complete the task, path plan for each robot in the collective such that each waypoint was visited once without colliding with *a priori* obstacles. The deliberative Genetic Algorithm Path Planner was then coupled with a reactive Potential Field Trajectory Planner and kinematic based controller to create a hybrid control architecture allowing the mobile robot to navigate between multiple user-defined waypoints, while avoiding *a priori* obstacles and obstacles detected using the robots' range sensors. The success of this hybrid control architecture was proven through simulation and experimentation using three of Dr. Robot's™ wireless X80 mobile robots.

**Keywords:** Path Planning, Genetic Algorithm, Collaborative Mobile Robots, Hybrid Control Architecture, and Trajectory Control

## 1. INTRODUCTION

Since the genesis of robotics in the early 1960's, robots have played an increasingly large role in today's society [1]. Their applications have been as diverse as the scientifically inspired NASA Mars Rovers to domesticated robot vacuum cleaners. In fact, there are currently over 800 000 industrial robots in operation and over 600 000 household robots, mainly composed of lawn mowing and vacuum cleaning robots [2]. Regardless of the application, path planning and its implementation are an integral part of the design of mobile robots. The goal of this research was to design and implement a hybrid control architecture to direct a collective of mobile robots to numerous user-defined waypoints while avoiding *a priori* obstacles and obstacles detected using the robots' range sensors. The main component of the deliberative layer of the hybrid control architecture was the Genetic Algorithm (GA) Path Planner and the main components of the reactive layer were the Potential Field Trajectory Planner and the kinematic based controller. Specific applications of this research include search and rescue missions and hazardous material surveillance, both examples of where robots have to move to numerous waypoints in order to complete their search. Recently, the Center of Robot Assisted Search and Rescue at the University of South Florida has designed single robots to search for survivors in collapsed buildings based on [3]. In fact, during the September 11<sup>th</sup> tragedy in the United States, robots were used to search for victims at the World Trade Centre site [2].

Hierarchical [1, 4] and reactive [1, 5] control architectures were also considered to meet the research goal. However, the hierarchical approach cannot effectively deal with unforeseen changes in to the robots' environment as it takes time to update the global world model and create a new plan. Furthermore, the computer processing time increases exponentially as the global world model expands due to an increase in the size of the area to be navigated. The major disadvantage of the reactive approach is its lack of planning which is not appropriate for complicated tasks where coordination between robots is required. Furthermore, the reactive approach does not allow for the retention of internal states, which again restricts the use of this form of control architecture. The selected hybrid control architecture combines the hierarchical and reactive approaches to mitigate their respective disadvantages. The hybrid control architecture was also chosen for its ability to plan using its deliberative layer and quickly respond to a changing environment using its reactive layer.

The potential fields [6], cell decomposition [1, 7, 8] and road map methods were considered for the deliberative layer of the hybrid control architecture. The potential field method has been successfully used in [9] and [10] for multiple mobile robot path planning for surveillance vehicles in a ground battlefield scenario and to ensure that members of a swarm, a large number of autonomous mobile robots working together, remain at a safe distance from each other while enforcing a level of cohesion. The major drawback of solely using the potential field method is the possibility that the algorithm will encounter a local minimum vice its final goal position. Another disadvantage for mobile robots using dead reckoning and the potential field approach is that the desired robot trajectories may be jerky.

The cell decomposition method has also been used to develop path plans for multiple robots as shown in [11]. The major drawback with this approach is the requirement to determine the resolution of the grid that creates the cells. If the resolution is too high, then the computational time to perform the search will be long. Conversely, if the resolution is too low, a free cell path may not exist.

Finally, the roadmap method analyses the connectivity of the freespace, the areas the robots can occupy, to create a network of free space curves known as the connectivity roadmap graph [2]. After this graph has been defined, it is searched for the optimal path. The roadmap approach was successfully applied to generate a path plan for six, six-degree-of-freedom welding robots as discussed in [12]. The selected deliberative layer of the hybrid control architecture used for this research was based on the concepts of the roadmap method. Once formed, the roadmap graph was then searched using a GA to find an optimized path. After considering other searches, such as the A\* search [1], it was determined that the GA search algorithm had the most potential to deal with multiple waypoint planning of mobile robots. The reason for this decision was that GAs can

find a complete solution to reach all the waypoints instead of navigating from waypoint to waypoint. This complete solution of waypoints can then be easily used to assign tasks to multiple mobile robots. Furthermore, it was foreseen that the GA search algorithm would be best suited for the optimal execution of other behaviours besides navigation, for future work. GA path planning has been used for path planning in [13] and in [14]. However both papers only give simulated results.

The reactive layer of the hybrid control architecture used the potential fields approach to generate the reference velocities for the robot to follow the path plan generated by the deliberative layer of the hybrid control architecture. The reference velocities were then achieved by the X80 mobile robots using a kinematic based controller using the non-linear controller developed at [15]. Other mobile robot controllers considered for this research included [16-18].

The discussion below applies the developed hybrid control architecture to investigate multiple waypoint planning for a collective of X80 mobile robots coloured red, green and blue. This investigation will discuss the design and theory involved in the creation of the deliberative and reactive layers of the hybrid control architecture. The investigation will then use the developed hybrid control architecture to generate simulated and experimental results of moving a collective of X80 mobile robots to the required waypoints along an optimized path in a two-dimensional environment populated with obstacles.

## 2. THEORY AND DESIGN

As illustrated in the hybrid control architecture shown in Figure 1, the main components of the reactive layer are the Potential Field Trajectory Planner and the kinematic based controller. The main components of the deliberative layer are the GA Path Planner and the Task Manager. Through the Task Manager, the user specifies the robots' task defined by the waypoint coordinates to be visited by the robots, the dimension of the workspace, and the coordinates of *a priori* obstacles. The Task Manager passes this information to the global world model and then initiates the GA Path Planner. The GA Path Planner uses the information in the global world model to plan the order in which the robots will visit the waypoints, so that all *a priori* obstacles will be avoided and the time to complete the task is minimized. Each robot's Potential Field Trajectory Planner module then generates reference trajectories using data from its robot's range sensors to move its robot along the planned path while avoiding obstacles, including other robots. In order to coordinate the robots in the colony and to ensure that there were no collisions between the robots, each robot's Potential Field Trajectory Planner had access to the current positions of all colony robots through the global world model. This information was used to stop the robot if it came within a set distance of a robot that had a higher priority. The priority of the robots was arbitrarily assigned for this investigation. Finally, the kinematic based controller associated with each robot implemented the trajectories developed by its Potential Field Trajectory Planner and sent the commands to drive the robots wheels at the required speeds. The kinematic based controller also updated its robot's position in the global world model. This section will introduce this hybrid control architecture's deliberative GA Path Planner, reactive Potential Field Trajectory Planner and kinematic based controller, and discuss how they were used to direct a collective of mobile robots to multiple user-defined waypoints in a 2-dimensional environment.

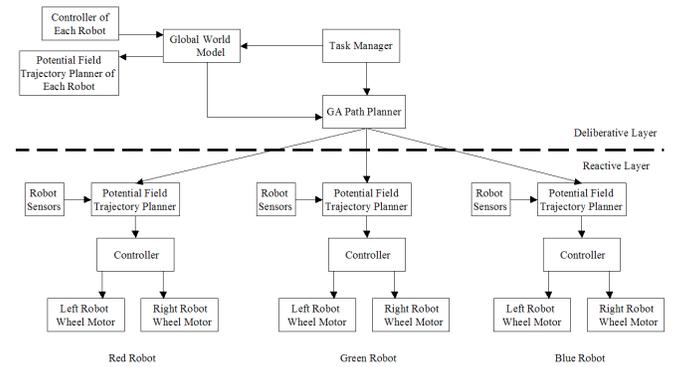


Fig. 1. Hybrid control architecture used to control a collective of three mobile robots.

The objective of the deliberative GA Path Planner was to generate an optimized path plan, by minimizing the time to complete the task, for the mobile robots to visit all of the user-defined waypoints without colliding with the *a priori* obstacles encountered in the robots' global world model. The main tool used to accomplish this objective was the GA search algorithm. Figure 2 contains a flow chart outlining the basic steps involved in the GA Path Planner.

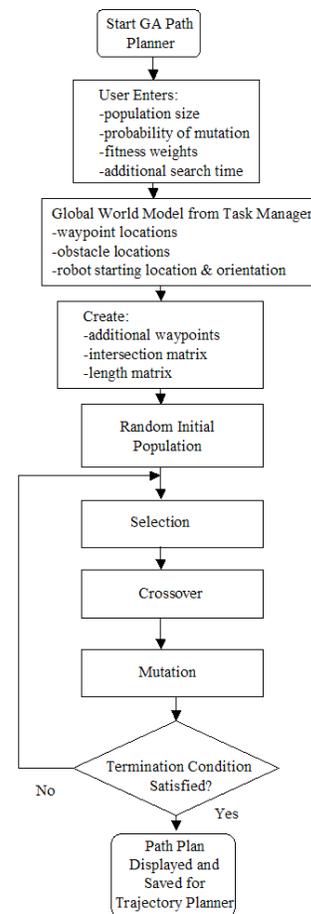


Fig 2: Flow chart for the GA Path Planner.

As illustrated in this figure, the user first enters the GA search parameters (population size, probability of mutation, fitness

weight, and additional search time) through a Graphical User Interface (GUI). The path planner then accepts the data contained in the global world model. The global world model contains the 2-dimensional coordinates of the user-defined waypoints for the robots to visit, the coordinates of the *a priori* obstacles, and the starting coordinates and orientation of the robots. The path planner then uses the global world model information to generate new waypoints to avoid collisions with *a priori* obstacles. Specifically, new waypoints are added to the end of the obstacle lines, defined by two obstacle points, at a distance to allow the X80 robots to safely pass. In most environments, these new waypoints are required to connect all the user-defined waypoints in a collision-free, straight-line path plan. At this stage, two matrices are also created: the Euclidean length between waypoints and whether an intersection with a *a priori* obstacle occurs between the waypoints. These two matrices will be used in the fitness function and are initially created in order to avoid needlessly repeating these calculations. The path planner then randomly creates an initial population of solutions equal to the population size specified by the user. The genome of these solutions is a sequential list of the order to visit all the user-defined waypoints. Also, associated with each additional waypoint is a flag to either activate or ignore the additional waypoint; a flag of 1 activates the additional waypoint and a flag of 0 ignores the additional waypoint. This flag allows a path plan to be generated that will permit the mobile robots to navigate around *a priori* obstacles while removing unnecessary additional waypoints. This feature only applies to additional waypoints, as the mobile robot must visit all the user-defined waypoints. For user-defined waypoints, the flag is assigned a value of 2. A flag of 3 signals the termination of the path of the red robot and the start of the path of another robot. The number preceding the flag was used to identify the new robot. Identical to the red robot, the path plans for the new robots were terminated when a waypoint flag of 3 was read. An illustration of the application of this GA genome is presented in Figure 3.

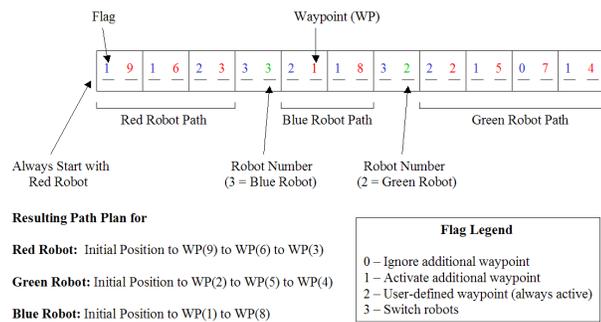


Fig. 3: Application of the multiple robot genome used by the deliberative GA Path Planner.

As part of the selection stage of the GA Path Planner, a fitness function is used to determine the probability of solutions within the existing population being chosen to create a new solution. The colony robot that had the longest path length was first identified for each solution in the population. From these lengths, the fitness function favoured the solution with the shorter single robot path length. The application of this method attempted to balance the distance traveled by each robot in the colony. This methodology was implemented to have the robots visit the user-defined waypoints in the shortest period of time. The factor that affected this desirable outcome was the longest path length of an individual robot in the colony; the colony

robots with shorter path lengths would be waiting for the colony robot with the longest path to finish. The fitness function uses the intersection and length matrices that were initially created to generate a probability that the solution will be selected for reproduction. These two components of the fitness function are weighted by the user to place emphasis on either quickly finding a collision-free path or finding an optimized path. Once two parent solutions have been selected for reproduction, based on their fitness function probability, the next step is to create a new child solution. The generation of the child solution uses a simple crossover point method to determine which parts of each parent's solution will be used in the child solution. If the crossover results in a child solution that has repeated and missing waypoints, the path planner corrects this problem by automatically replacing the repeated waypoints with the missing waypoints. Each child solution also has the possibility of being affected by a mutation; the probability of this mutation is defined by the user. The mutation function interchanges the values of two random positions within the child solution or it reverses the flag of a randomly selected additional waypoint. The resulting child solution will use parts of the parent solution, along with the possibility of random mutations to possibly create a superior solution when compared to the original two parent solutions. After enough child solutions have been created, they will replace the old parent solutions as the population with the exception of two cases. The solution from the old population that produced the best probability according to the fitness function and the solution from the old population that produced the least number of intersections will both be copied without mutation to the next population. This technique is known as elitism and it can increase the performance of the GA by ensuring that the best solutions will not be lost due to crossovers or mutations [19]. The GA Path Planner continues cycling through the above process until a solution with no intersections is produced. At this point, the planning will continue in order to optimize the path plan. The user determines the period of time allocated for this optimization.

After the optimization period has expired, the optimized paths for each robot were sent to their respective reactive Potential Field Trajectory Planner. Each robot's Potential Field Trajectory Planner module then generated reference trajectories using its robot's range sensors to move its robot along its planned path while avoiding obstacles, including other robots. The potential field method treats the mobile robot as a charged particle under the influence of a potential field. The mobile robot is attracted to its next waypoint and repulsed by the obstacles detected in the robot's workspace. The potentially jerky velocities of the Potential Field Trajectory Planner were smoothed using a velocity smoothing technique described in [17]. In addition, to ensure that there were no collisions between the robots, each robot's trajectory planner had access to the current position of all colony robots through the global world model. This information was used to stop the robot if it was in danger of colliding with a robot that had a higher priority. The priority of the robots was arbitrarily assigned for this investigation; the red robot had the highest priority and the blue robot had the lowest priority. Future work will examine using other factors to determine priority such as the robot with the longest path remaining being assigned the highest priority.

Finally, the kinematic based controller associated with each robot used the reference trajectories developed by its Potential Field Trajectory Planner to send commands to drive its robot's wheels at the required speeds to achieve the reference velocities and reach the waypoints. The kinematic based controller also determined its robot's position and orientation, using dead

reckoning, and updated its robot's position in the global world model.

The multiple robot hybrid control architecture used a combination of the de-centralised and centralised approaches for multiple robot planning as discussed in [12]. The deliberative GA Path Planner used aspects of centralised planning; the planner considered all robots to develop an optimized path plan. However, unlike true centralised planners, it did not plan reference trajectories for the robots to reach the user-defined waypoints. To create reference trajectories using the centralised approach would have required additional computational power to achieve reasonable deliberative GA Path Planner search times. This increase in computational power is probably best realized by linking computers as shown in [20]. However, using only one computer to control the robots was desirable for this research to improve the portability of the robot system and to decrease the overall cost. The reference trajectories generated by the Potential Field Trajectory Planner applied the de-centralised, prioritized approach; each robot had its own trajectory planner, which adjusted its reference velocities to avoid other robots. However, unlike true de-centralised planners, each robot did not generate its path plan separately; the path plan was generated by the deliberative GA Path Planner considering all the robots in the colony. The major disadvantage of the de-centralised approach is completeness; however, this disadvantage was deemed unimportant for this investigation as it was assumed that the robots would be operating in large areas where the requirement for highly coordinated robot movements would not be required.

All the modules of this investigation's hybrid control architecture were developed using MATLAB™ as it was correctly anticipated that the programming language's ability to easily manipulate matrices would be an asset. Visual Basic 6 on the host desktop computer was used to send commands generated from the hierarchical control architecture to the X80 mobile robots and receive their sensor data using a wireless router. The physical experimental set-up of the X80 robot collective is illustrated in Figure 4.

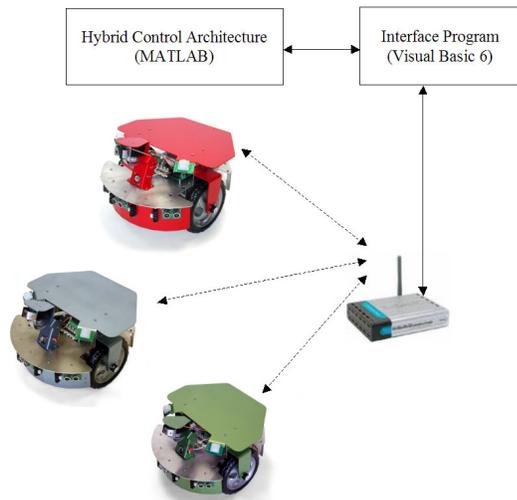


Fig. 4. Schematic of the flow of commands and feedback between the collective of X80 mobile robots and the MATLAB™ hybrid control architecture using the Visual Basic 6 interface program and wireless router.

A GUI, illustrated in Figure 5, was developed using Visual Basic 6 to allow the human operator to obtain sensor feedback such as the battery voltage levels, sonar distance ranges and IR

voltage values for the three X80 robots. Through the GUI, the user initiates simulated and experimental runs, and can stop the robots before the completion of their run. The robots' PID motor gains and the gains associated with the kinematic based controller can also be adjusted using the GUI. Finally, the cameras mounted on the X80 robots were used to display an image approximately every 10 seconds using the GUI. These images would be beneficial for surveillance applications.



Fig. 5: GUI used by the human operator to obtain sensor feedback, and generate simulated and experimental runs.

### 3. RESULTS

The deliberative GA Path Planner was first used to compare the results generated using a single robot and a colony of three robots. The global world model made for this comparison had three user-defined waypoints within an environment consisting of three rooms. The dashed line in Figure 6 illustrates a sample path plan generated for the colony of three robots. This trial was conducted 10-times for both the single robot and multiple robot scenarios. The user-defined GA search parameters for these trials were initially set as follows: 5% probability of mutation, population size of 50, an evenly weighted fitness function and an additional search time of 30 seconds after the first workable solution was found.

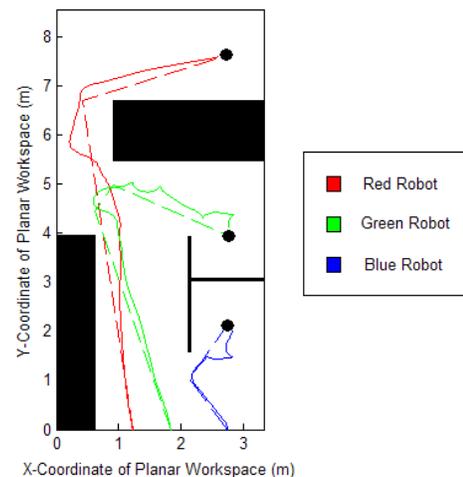


Fig. 6: Path plan (dashed line) and simulated reference path (solid line) for a colony of three mobile robots to visit the three rooms (defined by the three black circles) and avoid all obstacles.

After analyzing the results of these trials, it was found that the average overall path length of the single robot scenario was 17.74 m compared to an average overall colony path length of 18.65 m for the multiple robot scenario. However, if it is assumed that the robots travel at three-quarters of the X80 maximum speed, it will only take the colony of three robots 54 seconds to reach all their user-defined waypoints compared with a time of 118 seconds for the single robot. The time for the colony of robots to reach their user-defined waypoints was based on the longest average path length of the individual robots in the colony. Another advantage of using a colony of robots to complete a search is that it is easier for the deliberative GA Path Planner to generate a solution. The average time it took the deliberative GA Path Planner to find an optimized solution was 52 seconds for the single robot scenario compared with a shorter average time of 39 seconds for the multiple robot case. Initially, an evenly weighted fitness function was used to generate path plans for the single robot case. However, this weighting required unacceptable search times of over 2 minutes to generate a solution. The fitness function was therefore weighted solely on finding a collision-free solution vice finding an optimized solution for the single robot scenario.

The multiple robot path plan for the three mobile robots, illustrated by the dashed lines in Figure 6, was then sent to each robot's Potential Field Trajectory Planner. The solid lines in Figure 6 illustrate the simulated reference path generated by the Potential Field Trajectory Planner. As shown in this figure, the robots visited all three rooms, denoted by user-defined waypoints, and the robots avoided all obstacles. Also illustrated in this figure are the intersections of the paths of the red and green robots. To avoid collisions between these two robots, the lower priority green robot stopped if it came within 0.22 m of the higher priority red robot according to the theory detailed in Section 2. In simulation, the green robot did indeed stop for the red robot. During the first 15-seconds of the velocity profile for the green robot, illustrated in Figure 7, the green robot stopped five times to avoid a collision with the red robot.

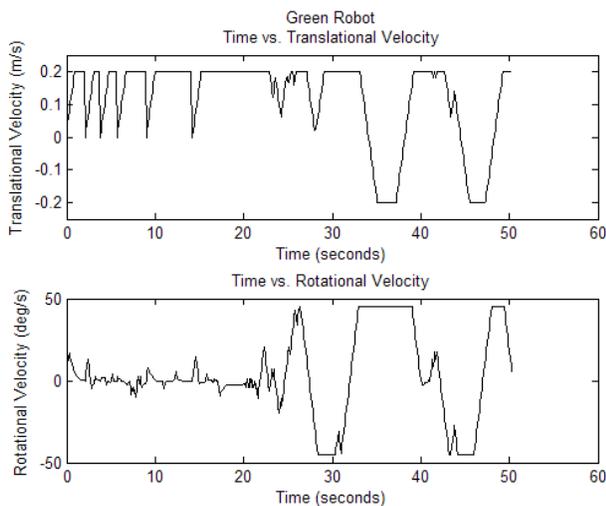


Fig. 7: Simulated reference translational (top graph) and rotational (bottom graph) velocity profiles for the simulated reference path of the green robot illustrated in Figure 6.

When the multiple robot hybrid control architecture kinematic based controllers were used to direct the three X80 robots along the reference paths, the colony robots followed the actual paths illustrated in Figure 8. The three X80 colony robots successfully visited the three rooms, denoted by the user-

defined waypoints, while avoiding all obstacles and other X80 robots. In doing so, the goal of this research, discussed in Section 1, was achieved.

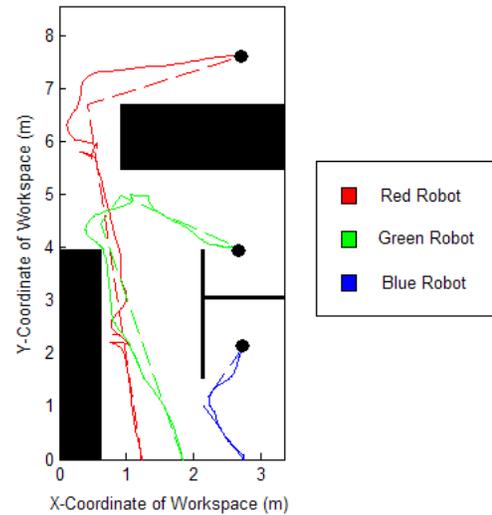


Fig. 8: Path plan (dashed line) and actual path (solid line) taken by the colony of three X80 robots to visit the three rooms (defined by the three black circles) and avoid all obstacles and other robots.

As illustrated in Figure 8 the path of the green X80 robot came close to the rectangular obstacle in the bottom left of the diagram. This obstacle was actually made up of computer carts. From the robot's level, the only obstacles detected by the sensors were the two thin, front legs of the computer cart. Since the robot sensors detected these thin legs infrequently, the robot perceived this area as an open space. To increase the distance between the computer carts and the green X80 robot, a solid and continuous barrier at the robot's level could be erected around the computer carts. A more complex solution might be to increase the number of range sensors on the X80 robot to improve its sensor coverage.

The red X80 robot's path was not smooth when the green X80 robot was close to it. This result was caused by the red X80 robot's Potential Field Trajectory Planner attempting to increase the distance from the green X80 robot and to avoid the obstacle to the left of the red X80 robot. Furthermore, the red X80 robot's path was not smooth when it was passing through the doorway to get to the top room in Figure 8. This result was probably caused the Potential Field Trajectory Planner attempting to find the best way between the close confines of the doorway.

Figure 9 illustrates the actual translational and rotational velocity profiles for the green X80 robot. In accordance with the theory discussed in Section 2 and simulated, the green robot did stop for the higher priority red X80 robot when it came within 0.22 m. In fact, the green X80 robot backed away from the red X80 robot to increase its distance from the red X80 robot. When the green X80 robot backed away from the red X80 robot, the green X80 robot's free-moving rear wheel lifted off the floor because of the robot's acceleration. This action is not desirable as it may negatively affect the accuracy of the robot's position and orientation calculated using the dead reckoning system. To remove this undesirable action, a weight could be added to the rear of the X80 robot. Another alternative to prevent this action might involve gradually reducing the speed of the robot when it comes close to a higher priority robot, instead of halting abruptly. The time period which the lower priority robot stops could have also been extended to

allow the higher priority robot more time to move away from the lower priority robot.

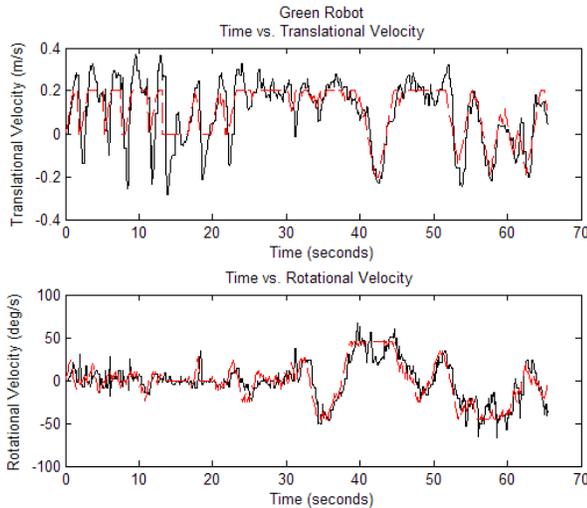


Fig. 9: Green X80 robot translational (top graph) and rotational (bottom graph) velocities for the actual path illustrated in Figure 8. The solid black lines denote the actual velocities of the green X80 robot and the dashed red lines denote the reference velocities.

#### 4. RECOMMENDATIONS

It would be beneficial to compare the deliberative GA Path Planner against other planners such as the probabilistic roadmap method or the A\* search algorithm. Furthermore, the deliberative GA Path Planner could be improved by culling solutions below a certain threshold, and by using different forms of crossovers such as two point or arithmetic [19].

As discussed in Section 2, the hybrid control architecture controller was a kinematic based controller. This controller may be improved to more accurately and quickly achieve the reference velocities by using a dynamic controller presented by [21]. This dynamic controller would account for such dynamic forces caused by the mass of the X80 robot. Due to the modularity of the hybrid control architecture, this exchange of controllers would be easy to implement.

Section 3 discussed the sensor difficulties experienced during this research. Specifically, there were locations to the rear of the robot that were not covered by the range sensors. This problem may have been remedied by the addition of more range sensors to remove any “dead areas” around the robot. Other difficulties included the inherent unreliability of sensor data. The incorporation of data fusion techniques to produce more reliable sensor readings might resolve this issue.

The research contained in this paper may be built upon by using a heterogeneous robot colony. This future work may improve the colony’s ability to handle different terrains within the environment. For example, a robot with tracks may be used to navigate through uneven terrain or a snake-like robot may be used to navigate between tight obstacles. These robots in the heterogeneous colony may also be assigned different roles. For example, one robot may be outfitted with an accurate laser range finder to build a map of an unknown. Once at the user-defined waypoint, robots with advanced sensors may be used to monitor the condition of human casualties, take readings of the environmental toxins, or use cameras to survey the area. Robots may also be outfitted with manipulators that can move obstacles or even evacuate human casualties to safety. The creation of a heterogeneous colony of robots would involve advanced

planning and methods to assign roles to the colony robots. The GA Planner presented in this research could be modified to meet these challenges.

#### 5. CONCLUSION

This paper has presented the design and implementation of a hybrid control architecture to direct a collective of mobile robots to numerous user-defined waypoints while avoiding obstacles and other colony robots. These user-defined waypoints could represent possible victims or points of interest in a hazardous material spill or a search and rescue mission. This hybrid control architecture used a deliberative planner based on GAs and centralized planning to quickly optimize a path plan while avoiding all *a priori* obstacles in the global world model and minimizing the time for the colony to complete the task. For the reactive layer of the hybrid control architecture, the potential field method was shown to produce reference velocities, in simulation and experimentally, to move the robot along the path generated by the GA path planner while maintaining a safe distance from all obstacles detected using the robots’ sensors. Each colony robot’s kinematic based controller was then used to send commands to the X80 robots wheel motors to move it to the desired waypoints at the reference velocities.

It was also shown that using a colony of robots to visit user defined waypoints resulted in: a shorter deliberative GA Path Planner search time; a reduction in the time to complete the overall task, and more emphasis could be placed on finding an optimized path. Finally, this paper offered a set of recommendations to improve further research in the area of collaborative mobile robots.

#### 6. REFERENCES

- [1] R. Murphy, *Introduction to AI Robotics*. London: MIT Press, 2000.
- [2] F. Lewis and G. Shuzhi, *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*. Boca Raton: CRC Press, 2006.
- [3] J. L. Casper, M. Micire, and R. R. Murphy, "Issues in Intelligent Robots for Search and Rescue," *Proceedings of SPIE*, vol. 4024, pp. 292-302, 2000.
- [4] R. Chatila and J. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 138- 145, 1985.
- [5] R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 14, 1986.
- [6] M. W. Spong, M. Vidyasagar, and S. Hutchinson, *Robot Modeling and Control*. New York: Wiley, 2006.
- [7] L. Yibin, L. Caihong, and Z. Zijian, "Search Strategy of Path for Mobile Robot," presented at IEEE International Conference on Vehicular Electronics and Safety, 2005.
- [8] O. Brock and L. E. Kavraki, "Decomposition-based Motion Planning: A Framework for Real-time Motion Planning in High-dimensional Configuration Spaces," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1469- 1474, 2001.
- [9] J. S. Baras, T. Xiaobo, and P. Hovareshti, "Decentralized Control of Autonomous Vehicles," *42nd IEEE Conference on Decision and Control*, vol. 2, pp. 1532-1537, 2003.
- [10] D. H. Kim, "Self-Organization for Multi-Agent Groups," *International Journal of Control, Automation, and Systems*, vol. 2, pp. 333-342, 2004.
- [11] T. Simeon, S. Leroy, and J.-P. Laumond, "Path Coordination for Multiple Mobile robots: a Resolution-complete Algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, 2002.

- [12] G. Sanchez and J. C. Latombe, "Using a PRM Planner to Compare Centralized and Decoupled Planning for Multi-robot Systems," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2112-2119, 2002.
- [13] L. Tian and C. Collins, "Motion Planning for Redundant Manipulators Using a Floating Point Genetic Algorithm," *Journal of Intelligent and Robotic Systems*, vol. 38, pp. 297, 2003.
- [14] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H.-M. Tail, "Autonomous Local Path Planning for a Mobile Robot using a Genetic Algorithm," *Evolutionary Computation*, vol. 2, pp. 1338-1345, 2004.
- [15] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," *1990 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384-389, 1990.
- [16] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive Tracking Control of a Nonholonomic Mobile Robot," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 609-615, 2000.
- [17] M. Egerstedt and X. Hu, "A Hybrid Control Approach to Action Coordination for Mobile Robots," *Automatica*, vol. 38, pp. 125, 2002.
- [18] C. Samson and K. Ait-Abderrahim, "Feedback Control of a Nonholonomic Wheeled Cart in Cartesian Space," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1136-1141, 1991.
- [19] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall/Pearson Education, 2003.
- [20] C. Clark, S. Rock, and J. Latombe, "Dynamic Networks for Motion Planning in Multi-Robot Space Systems," *International Symposium of Artificial Intelligence, Robotics and Automation in Space*, 2003.
- [21] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot using Neural Networks," *IEEE Transactions On Neural Networks*, vol. 9, pp. 589-600, 1998.