Hybrid deliberative reactive navigation system for mobile robots using ROS and Fuzzy Logic Control

Elvis Ruiz, Raul Acuña, Pedro Vélez, Gerardo Fernández-López
Mechatronics Research Group
Simón Bolívar University
Miranda, Venezuela
Email: elvisruiz@usb.ve

***Abstract*—Autonomous navigation involves solving problems of perception, localization, planning and control. Based on how these problems can be solved it is possible to have a deliberative, reactive or deliberative-reactive (hybrid) navigation. Additionally, navigation must meet security requirements for both the robot and humans operating in the same workspace. Thus, a navigation system for differential hybrid mobile robots in structured environments was designed, which integrates the advantages of a high level deliberative planner with a reactive low-level control and easing some of its weaknesses. For this, a fuzzy logic controller was implemented. The proposed system was tested with a simulated robot Amigobot with a Kinect sensor using V-REP and ROS.**

***Keywords:* Navigation, path-planning, mobile robots, control, ROS, Kinect.**

I. INTRODUCTION

In the past three decades serious efforts have been made to design a strategy for efficient and reliable navigation which can be able to solve problems of obstacle avoidance, reaching objectives, transport, surveillance and exploration [1]. The current navigation algorithms can be categorized into two types: those targeted for cases with full knowledge of the environments and those with incomplete knowledge [2]. In this regard, the first efforts to solve a plan free of obstacles in order to reach a goal from a map built by the knowledge of the environment belongs to the first type of navigation algorithms.

However, fully known environments, which are generally solved using deliberative navigation systems, have been widely solved, so the challenge lies in working in dynamic environments [3]. In this sense, reactive navigation can cope better with these scenarios. However, in reactive navigation, the prior knowledge or partial knowledge of the world its not used and this becomes a disadvantage for certain cases in real applications.

The concept of hybrid navigation presents the possibility of using both a global vision of the environment as well as a reactive behaviour. However, the main problem that arises is how to build an adequate cooperation between the different navigation strategies (deliberative and reactive) in order to combine them into a new one.

II. RELATED WORKS

A hybrid navigation that deals with a partial and imprecise knowledge of the environment can be found in the work of L. Wang 2002 [4], where a global planner based on the distance transformation algorithm and a local control based on potential fields are proposed. For this type of problem other alternatives had been proposed, such as the use of the path planning algorithm A* with fuzzy control, H. Maaref 2002 [5], or using fuzzy neural networks to establish reactive laws, Y. Jiang 2005 [6]. At the reactive level can be found methods of: potential fields [7], histogram of velocity vectors, elastic bands and dynamic windows approaches with the use of a localization based on odometry and the extended Kalman filter, R. Vazquez-Martin 2006 [8].

The global planner can be established through algorithms based on sampling, graph search, cell decomposition or the implementation of algorithms based on biological inspirations like pulse coupled neural networks, H. Qu 2009 [9], or meta-heuristics like genetic algorithms, H. Huang 2011 [10]. An advantage of sample-based planners is that they do not have to explored every possible action sequence. Additionally, the traditional planners have no direct means of verifying that no solution exists for a particular domain [11].

However, the combination of this paradigms brings to the surface three major problems: cooperation mode (synchronous or asynchronous), use of references, and resolving conflicts between deliberative planning and reactive control. An independent proposal for asynchronous integration, the management of lists to place and extract references and give priority to the reactive layer in the presence of a conflict, can be found in Y. Zhu 2012 [2].

III. METHODOLOGY

The proposed system is conformed by the following principal parts:

1) Deliberative navigation layer: Given a discrete map, in which each part of the workspace could be unknown or possess a value of occupation probability, and an initial and final position for the robot, it is possible to plan a path using sampling based algorithms.

2) Reactive navigation layer: Given the robot positions, the objective and the obstacles (known and uncertain), potential

978-1-4673-7129-2/15 $31.00 © 2015 IEEE
DOI 10.1109/LARS-SBR.2015.24

fields are generated which will be converted into velocity references for the robot using the negative gradient of the resulting potential field using the techniques described in previous works of our group [7].

3) Hybrid navigation system: Based on the robot's defined safety variables relative to the environment, for example feasibility of the deliberative planned route or high proximity to obstacle, the results from both methods previously mentioned are integrated into a new reference for the robot.

The proposed solution is organized as shown in Figure 1, and it consists of five main blocks. First, the robot has on-board sensors and actuators that allow it to navigate the environment in which it is located. In this sense, the sensing system of the robot is used for construction and renovation of a map. The representation of the environment for the robot, and for purposes of this investigation, a two-dimensional plane with a arbitrary and finite number of static obstacles are used. Additionally, it is possible to have initial pose knowledge or not, and this knowledge may not be correct at all. In other words, a certain location on the map can be known or unknown with an uncertain value.

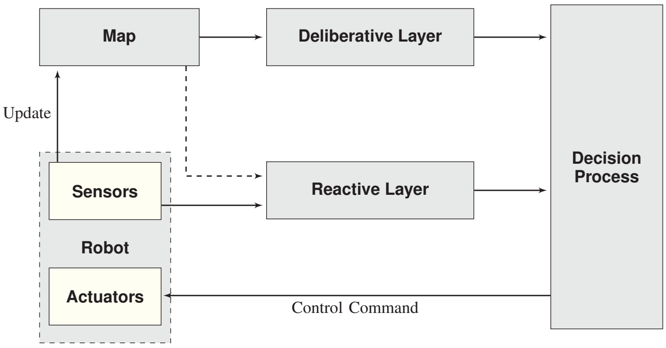


Fig. 1. Hybrid navigation architecture.

For the implementation of the proposed solution, a 3D model of the robot AmigoBot® with the design tool SolidWorks was designed, this model was used for the simulations. The dynamic model of the robot was implemented in the simulation framework V-Rep [12], where it is possible to observe the robot's behaviour and monitor all the variables of interest. The code responsible for carrying out all the necessary computations and send the appropriate control signals to the actuated elements of the robot were implemented in ROS [13]. And a pre-defined Kinect sensor of V-REP was used, with their respective constraints for indoor navigation, implemented as in our previous work [14].

Additionally, the platform Move-it [15] was used, it represents the state of the art software for mobile robotics. It incorporates the latest advances in motion planning, 3D perception, kinematics, control and navigation. Finally, the library fuzzylite [16] was used as a fuzzy control framework.

The project consists of nine major nodes using the ROS parallel processing capabilities. The following is a detailed explanation of each node and the interconnection between them.

- **V-REP:** Node that simulates the robot with its sensors and actuators, and the environment. It is also responsible for passing other nodes the following information: data generated by the Kinect sensor on the robot, the position of the robot, and the position of the target. It receives from other nodes the reference speeds for each of the motors on the robot (left and right).
- **Move-it:** It is the node responsible for creating a map from the mobile robot sensor data (Kinect) and the path generation by request, based on sampling algorithms, from an initial state to a end state [15]. These states correspond to the actual robot and target positions.
- **Plan Status:** It monitors the validity of the plan. Since the environment is changing during execution, the generated plan may no longer be feasible so it may require replanning.
- **Path following:** This node is responsible for following the path produced by the previously mentioned, using a weighted sum of an approach field and a tangential field to the route as explained in the work of Medina-Melendez [17].
- **Potential field:** This node calculates the potential fields associated to the obstacles, according to what is perceived by the sensors and the knowledge of the surrounding map, and sets the attractor field towards the target. The equation to generate a repulsive field should produce a soft landing, with high repulsion in the middle of the obstacle and a limited area of effect in the space around the obstacle defined by some variable. Gaussian repulsive fields have been previously used for obstacles as they fulfil the conditions, and are generated by the following expression [18] [7]:

$$Z(x,y) = Ae^{-((\frac{(x-x_o)^2}{2\sigma_x^2})+(\frac{(y-y_o)^2}{2\sigma_y^2}))} \tag{1}$$

  Where A indicates the maximum weight or height of the potential field, σ_x^2, σ_y^2 represent the maximum dispersion of the field in the corresponding axis, which allows the creation of elliptical fields, and x_o, y_o denote the center of the obstacle.
- **Gradient:** This node calculates the gradient of the previously obtained potential field. This will provide a velocity field that will be used to possibly guide the robot safely to a target.
- **Fuzzy logic:** According to two input variables: Validity of the plan and security; the first one given by the deliberative layer and the second by the reactive layer the following fuzzy rules are proposed:
    - If **Plan** is **Valid** and **Safety** is **High**. Then **Navigation** is **Deliberative**.

- ○ If **Plan** is **Valid** and **Safety** is **Low**. Then **Navigation** is **Reactive**.
  - ○ If **Plan** is **Invalid**. Then **Navigation** is **Reactive**.

  From this set of rules a value between 0 and 1 will result, which will indicate the level of deliberativeness or reactivity of the resulting command.

- **Fuzzy control:** From a decision f and two given speed commands (one deliberative - V_D and the other reactive - V_R), a resulting command agreement is established according to the following expression:

$$V = fV_d + (1 - f)V_r \tag{2}$$

- **Control:** A PI controler (proportional integrative) for the magnitude and phase of a given velocity vector was programmed. Additionally, the velocity vector module is scaled with the cosine of the error angle, this will allow the platform to move with the projection of the velocity vector over its current moving direction and avoid movements in inappropriate directions [18].

A. Reactive layer

This layer is responsible for attracting the robot to a target point and ensure the safety of the mobile by keeping it away from the perceived and previously known obstacles using potential fields.

It is composed of the nodes: Move-it, Potential Field and Gradient, it receives as input the sensor information and returns a velocity field, see Figure 2. In addition, an area of safe operation is set for the robot and with the gradient in this zone the value of the security status variable is determined. As shown in the figure 5, for each position of the robot a velocity vector is defined, according to the obstacles that the mobile has sensed and the knowledge that it has in that moment.

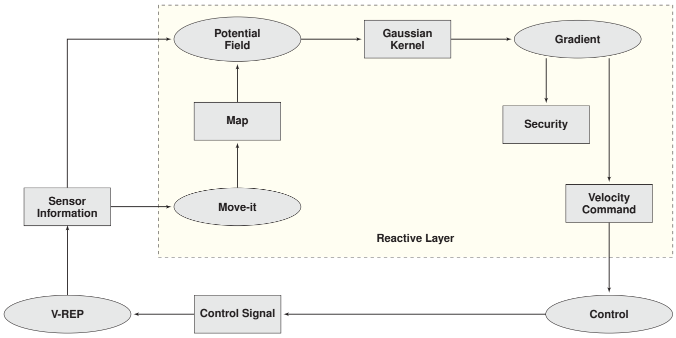


Fig. 2. Reactive navigation architecture.

B. Deliberative layer

It is responsible for generating a path from the current position of the robot to a target desired point and ensures the validity of the planning generated, if the route is no longer viable a new solution is sought, if any.

It is composed of the nodes: Move-it, Plan Status and Path Following. It receives the sensor information as the input, calculates a plan and returns a valid velocity command for the current position of the robot, see Figure 3. Due to the fact that the environment may change during execution, it is possible to make online re-planning, leaving the former Plan invalid, as shown in Figure 6. In case of an update in the map, and if the deliberative layer detects an obstacle in the planned route, a new plan will be made.

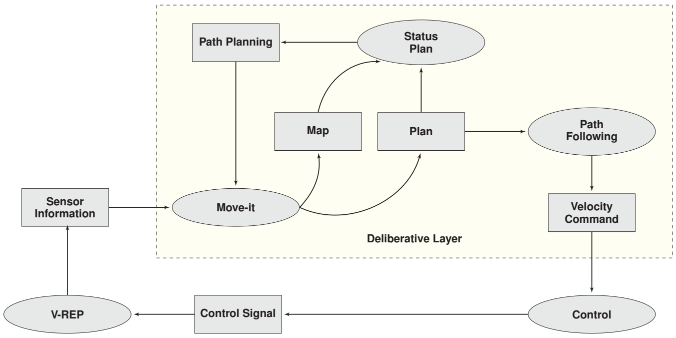


Fig. 3. Deliberative navigation architecture.

C. Decision processes

The decision process block is responsible for observing the validity of the plan and the safety of the robot in order to establish which control command will be sent to the actuators of the robot.

It is composed of the following nodes: Fuzzy Logic and Fuzzy Control, it receives as input two statuses (security and planning) and two velocity vectors associated with a purely deliberative and a purely reactive navigation. It is also responsible for returning a hybrid velocity vector by processing the supplied inputs, see Figure 4. In this way, the robot's behaviour can be more reactive or deliberative depending on the surrounding conditions, as shown in Figure 7. Along the trajectory the robot will have a dominant behaviour. With this and according to a given reference vector given by the reactive layer and one from the deliberative layer plus the decision of the Fuzzy Logic, a hybrid vector in module and angle is created, equation 2, in module-angle figure 8 and 9.

D. Test Cases

The proposed solution aims to solve general problems of navigation in environments with uncertainty. The cases shown in the table I were selected. Where each position of the workspace can be Known Accurately (AK), Unknown (U) or Inaccurately Known (IK), ie. an object that was in one place and it is not. In this regard, environments with variations in these parameters are created to set the initial conditions of navigation. In Figure 10, a representation of the case 2 is shown, in Figure 10a the complete V-REP simulation environment is shown and in Figure 10b the knowledge of the environment held by the robot.

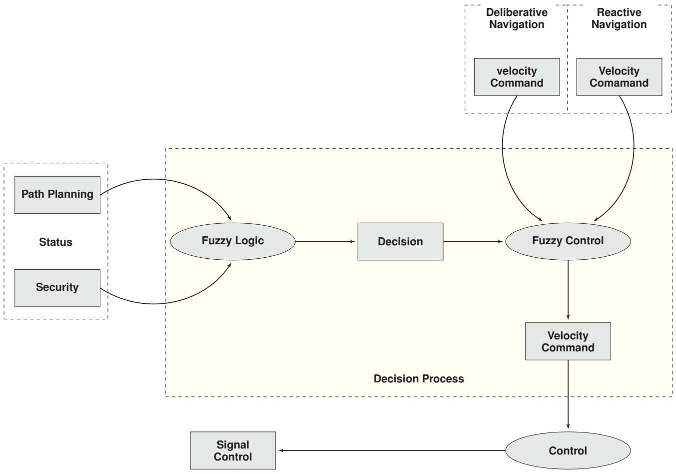


Fig. 4. Decision process architecture.

TABLE I. KNOWLEDGE LEVELS

Case	AK[%]	U[%]	IK[%]
1	80	15	5
2	30	35	35
3	10	80	10
4	5	95	0

IV. RESULTS AND DISCUSSION

The nodes were programmed using ROS Groovy, this version is stable in the operating system Ubuntu 12.04.5 LTS (*Precise Pangolin* Long Term Support).

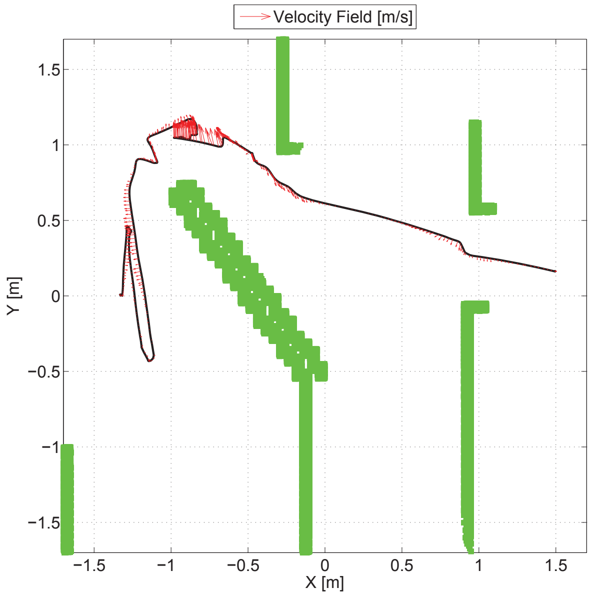


Fig. 5. Reactive field for each point in the trajectory executed. Green dots represent obstacles or walls.

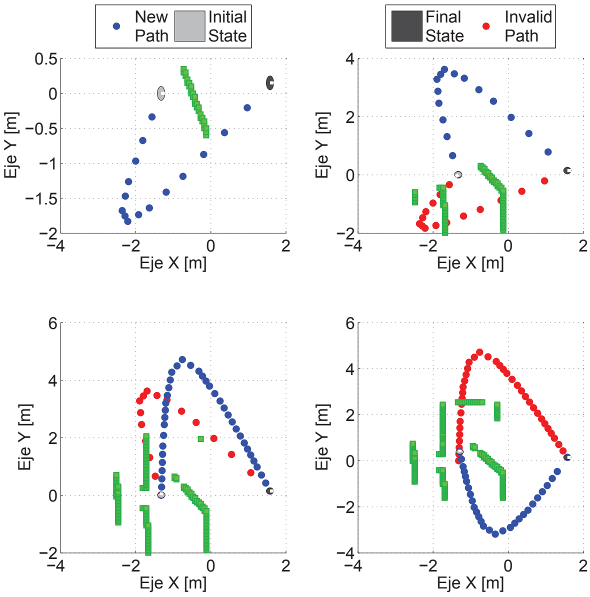


Fig. 6. Initial path and replanning in cases of an invalid plan.

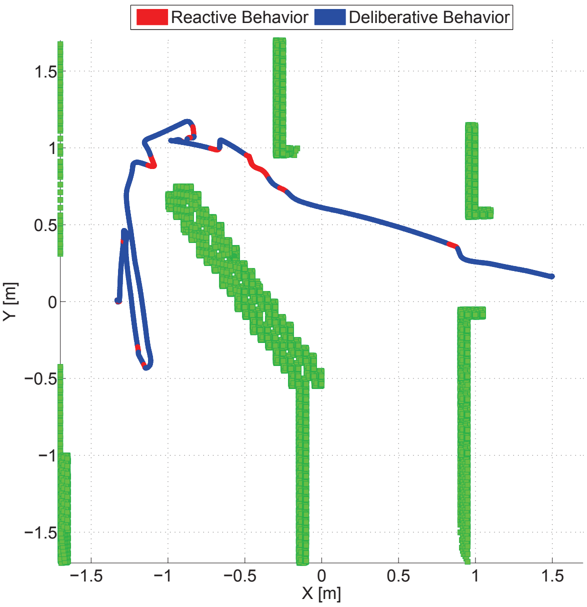


Fig. 7. Dominat behavior of the mobile robot along the trajectory.

From the proposed cases in terms of percentage of success (robot reach the goal safely), the following results were obtained, see table II. For a purely reactive navigation the robot is unable to reach the target due to the presence of local minima in the potential fields associated to the environment.

The deliberative navigation with replanning presented high levels of success, however, in terms of safety fails in such cases where the robot strikes or rubs the walls, this was caused by a lack of available time for replanning or because the planned route was too adjusted to the dimensions of the mobile robot and the imprecisions in the movement control caused a collision. It is worth noting that the behavior of the deliberative path-planner may be improved by increasing the safety radius around the obstacles for the planning (expanding the borders) or a greater restriction in the maximum movement velocity of

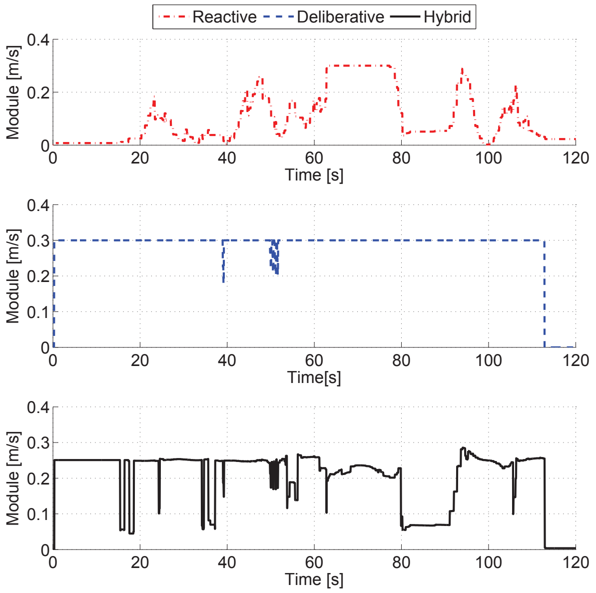


Fig. 8. Module of the hybrid velocity vector according to a deliberative vector, a reactive vector and a decision.

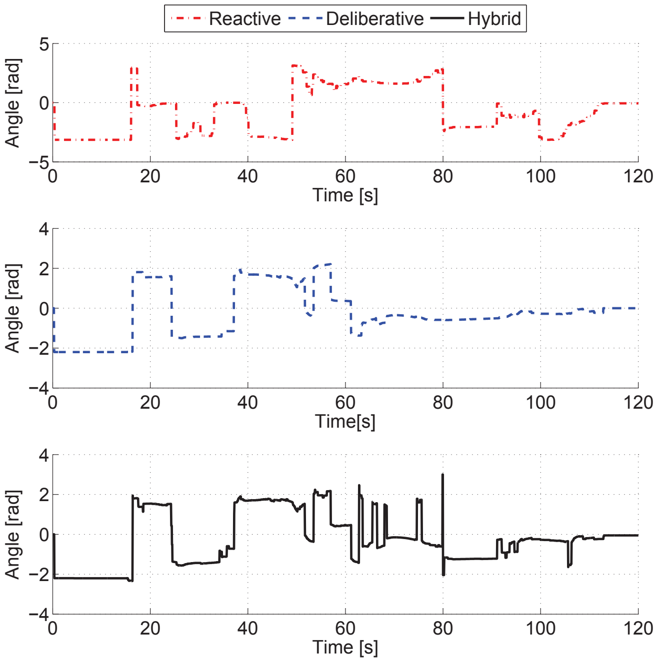
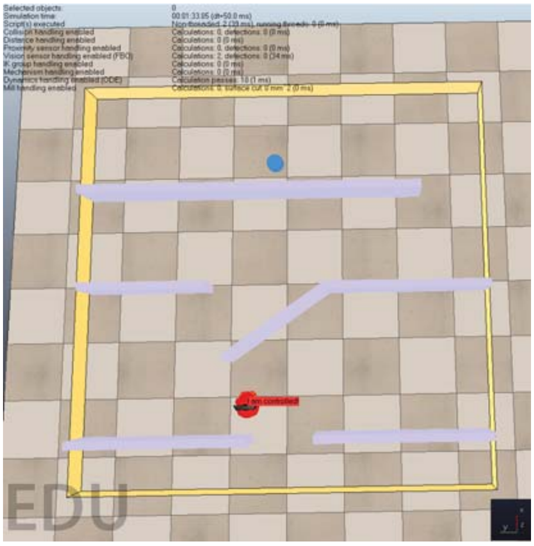


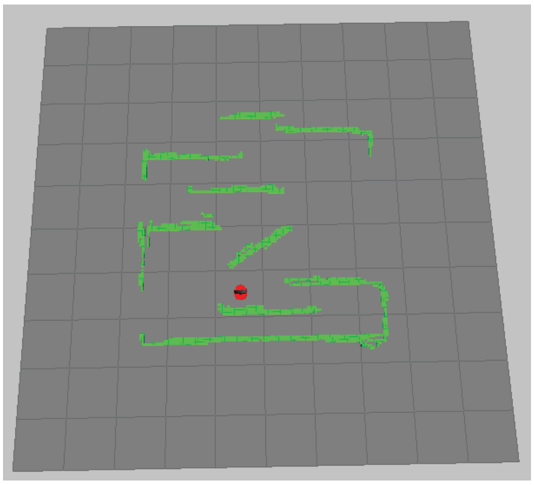
Fig. 9. Angle of the hybrid velocity vector according to a deliberative vector, a reactive vector and a decision.

the robot, but that would require the system to specialize in a unique set of cases.

Finally, the proposed hybrid navigation achieved a hundred percent success in the test cases. In times when deliberative planning was very risky, the reactive layer took control and it assured the safety of the mobile robot. In the case when there was no valid plan, the reactive layer allowed to reduce the risk as it can be seen in figure 11.



(a) V-REP



(b) Move-It

Fig. 10. Environment of work and initial conditions of knowledge.

TABLE II. RESULT OF THE PROPOSED CASES

Case	Reactive[%]	Deliberative[%]	Hybrid[%]
1	0	80	100
2	0	90	100
3	0	60	100
4	0	40	100

V. CONCLUSION

A hybrid navigation architecture that could benefit from the advantages of reactive and deliberative navigation, mitigating their individual disadvantages like local minima and planning time, was presented. Also a strategy to merge the commands computed by each layer and prioritize in case of conflicts based on Fuzzy Logic was defined.

In contrast to the works of Y. Zhu [2] [1] and H. Maaref [5], the reactive layer has used the map information in an asyn-

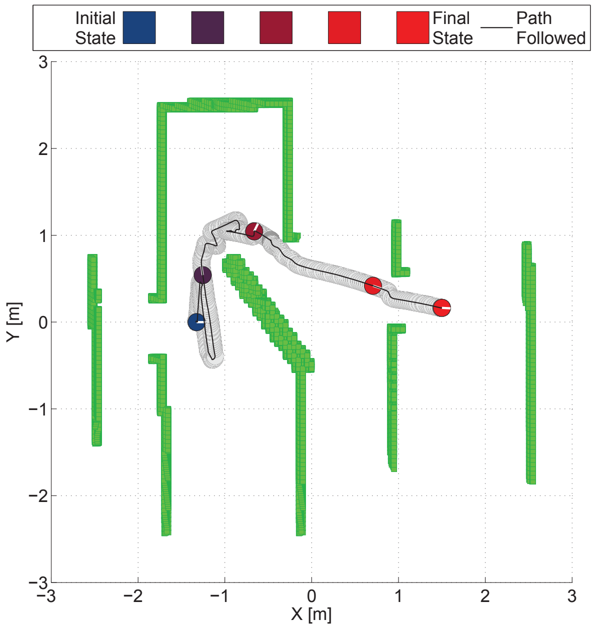


Fig. 11. A resulting hybrid navigation to the case 4 of knowledge.

chronous way and in a nearby area around the mobile, to take advantage of the newly known, and the deliberative navigation has used a sampling-based algorithm, with which the concept of probabilistic completeness [11], in dynamics environments, is taken into account, to take advantage of the planning time.

Finally, the proposed architecture was tested successfully in environments with different levels of uncertainty using ROS, Move-It and V-REP as framework. Furthermore, thanks to the modularity with which the proposed navigation system was developed, an implementation in a physical robot will be straight forward and it will be the next step to execute in our research group, the only thing that will change is the robot module (V-REP) which will be replaced by the actual robot, all else remains the same. Consequently, a low cost approach could be made using the Kinect Sensor and the AmigoBot, both widely used in universities and research laboratories. Nevertheless, the architecture could be applied to other robots and sensors (3D, cameras, laser).

REFERENCES

[1] Y. Zhu, T. Zhang, J. Song, and X. Li, "A hybrid navigation strategy for multiple mobile robots," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 4, pp. 129–141, Aug. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0736584512001408

[2] ——, "A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge," *Know.-Based Syst.*, vol. 27, pp. 302–313, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2011.11.009

[3] C. Laugier, D. Vasquez, M. Yguel, T. Fraichard, and O. Aycard, "Geometric and bayesian models for safe navigation in dynamic environments," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 51–72, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11370-007-0004-1

[4] M. Ang, "Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment," in *Proceedings of the IEEE International Symposium on Intelligent Control*. IEEE, 2002, pp. 821–826. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1157868 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1157868

[5] H. Maaref and C. Barret, "Sensor-based navigation of a mobile robot in an indoor environment," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 1–18, Jan. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889001001658 http://linkinghub.elsevier.com/retrieve/pii/S0921889001001658

[6] M.-y. Zhao, H.-g. Wang, and L.-j. Fang, "Hybrid navigation for a climbing robot by fuzzy neural network and trajectory planning," in *2005 International Conference on Machine Learning and Cybernetics*, no. August. IEEE, 2005, pp. 1069–1075 Vol. 2. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1527102

[7] R. Acuña, A. Terrones, N. Certad-H, and L. Fermín-León, "Dynamic potential field generation using movement prediction," in *Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, vol. 23. World Scientific, 2012, p. 26.

[8] R. Vazquez-Martin, E. Perez, C. Urdiales, J. del Toro, and F. Sandoval, "Hybrid navigation guidance for intelligent mobiles," in *2006 IEEE 63rd Vehicular Technology Conference*, vol. 6, no. c. IEEE, 2006, pp. 2992–2996. [Online]. Available: http://webpersonal.uma.es/ EPEREZ/files/VTC06.pdf http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1683417

[9] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model." *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 20, no. 11, pp. 1724–39, Nov. 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5256181 http://www.ncbi.nlm.nih.gov/pubmed/19775961

[10] H. Huang and C. Tsai, "Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm," *SICE Annual Conference (SICE), 2011 . . .*, pp. 1338–1343, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6060543

[11] S. Zickler, "Physics-based robot motion planning in dynamic multi-body environments," Ph.D. dissertation, 2010.

[12] M. F. E. Rohmer, S. P. N. Singh, "V-rep: a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[13] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[14] E. Ruiz, R. Acuna, N. Certad, A. Terrones, and M. E. Cabrera, "Development of a Control Platform for the Mobile Robot Roomba Using ROS and a Kinect Sensor," in *2013 Latin American Robotics Symposium and Competition*. IEEE, Oct. 2013, pp. 55–60. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6693270

[15] Ioan A. Sucan, Sachin Chitta, "Moveit," 2014. [Online]. Available: http://moveit.ros.org

[16] Juan Rada-Vilela, "fuzzylite: a fuzzy logic control library," 2014. [Online]. Available: http://www.fuzzylite.com

[17] W. Medina-Melndez, L. Fermn, J. Cappelletto, C. Murrugarra, G. Fernndez-Lpez, and J. Grieco, "Vision-based dynamic velocity field generation for mobile robots," in *Robot Motion and Control 2007*, ser. Lecture Notes in Control and Information Sciences, K. Kozlowski, Ed. Springer London, 2007, vol. 360, pp. 69–79.

[18] P. Estevez, J. Capelletto, F. Alvarez, R. Acuña, and G. Fernandez-Lopez, "Coordinated navigation Using Dynamically Varying Velocity Fields," *9th International Conference on Artificial Intelligence and Soft Computing*, 2008.