2015 IEEE International Conference on
Automation Science and Engineering (CASE)
Aug 24-28, 2015. Gothenburg, Sweden

A hybrid control architecture for autonomous mobile robot navigation in unknown dynamic environment

Danial Nakhaeinia, Pierre Payeur, Tang Sai Hong and Babak Karasfi

Abstract— **This paper introduces a new hybrid control architecture for solving the navigation problem of mobile robot in an unknown dynamic environment based on an actual-virtual target switching strategy. This hybrid architecture is a combination of deliberative and reactive architectures which consists of three layers: modeling, planning and reaction. The deliberative architecture produces collision-free with shortest-distance path, while using the reactive architecture generates safe and time minimal navigation path. The proposed approach differs from previous ones in its integration architecture, the control techniques implemented in each module, and interfaces between the deliberative and reactive components. Validity and feasibility of the proposed approach are verified through simulation and real robot experiments.**

I. INTRODUCTION

The most significant issue in the development and design of autonomous mobile robots is the ability of the robot to plan collision-free motions and perform reliable navigation within its environment. Different control architectures have been proposed for autonomous navigation of mobile robots. These control architectures could be classified into three categories: Deliberative (Global) navigation, Reactive (Behavior-based) navigation, and hybrid (Deliberative-Reactive) navigation (see [1] for a review of control architectures).

The deliberative control architecture [2-4] consists of three modules: perception, planning and action. First, the robot uses a global model of the environment which is provided by user input or creates a model of a static environment by combining sensory information. Then it employs a planning module to search for an optimal path and generates appropriate plan to steer the robot towards the goal. Finally, the robot executes the desired actions to reach the target. Reactive (behavior-based) navigation architecture was developed by Brooks [5] to tackle the navigation shortcomings of the deliberative approaches in dynamic and unknown environments. Proposed reactive methods [6-9] employ a Planning-Reaction configuration where it is not necessary to build a complete model of the environment. The action generation is based on the currently perceived environment and the sensed data directly couples to the robot's actuators. Although the deliberative and reactive architectures established a successful framework for mobile robot navigation, they cannot solve the navigation problems individually. Some features of deliberative architecture can be combined with the reactive architecture to achieve a comprehensive navigation in a real world which is called a hybrid architecture. The hybrid control architecture [10-12] involves the advantages of planning in deliberative architectures for high level issues to develop an optimal plan and the quick response of reactive architectures in dynamic or unknown environments on the low level.

Review of characteristics, advantages and drawbacks of different control architectures [1] and various path planning methods [6] show that: 1) the hybrid control architecture which utilizes the advantages from both deliberative (global) and reactive architectures is more robust and has better performance in unknown and dynamic environments, 2) the fuzzy logic navigation method is simple, fast and more coherent for reactive navigation and velocity control [13], and 3) the actual/virtual sub-goal approaches are more promising in the way to help the basic tasks of obstacle avoidance and to cope with the local minimum problem.

This paper introduces a new hybrid control architecture for mobile robot navigation in an unknown and dynamic environment. This architecture is a combination of the deliberative and reactive navigation architectures which is developed based on a modeling-planning-reaction configuration. The modeling layer processes and interprets sensory information to create a local model of the environment. The planning layer is responsible for decision making to avoid obstacle collision and local minimum trap situations. This layer is developed based on the actual-virtual target switching strategy. The robot motion generation is handled by the reaction layer. The latter applies a fuzzy controller to control the robot's rotational and translational velocities for fast reaction to the obstacles and optimization of the navigation time.

II. PROPOSED APPROACH OVERVIEW

The proposed hybrid control architecture is a combination of the deliberative and reactive navigation architectures which is founded on the use of three layers: *Modeling, Planning and Reaction* (Fig.1). The integration of the layers is based on a perception-planning–reaction configuration where both the planning and reaction layers concurrently use the local model of the environment constructed by the first layer in execution time.

Initial locations of the robot and the global target are set arbitrarily by the user for each navigation task. The action selection and the interaction of the modules of each layer are based on obstacles configuration. As shown in Fig. 2, the action selection algorithm starts by constructing a local occupancy map using information from a laser scanner. Then, two conditions are checked based on the obstacle

D. Nakhaeinia and P. Payeur are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada, K1N 6N5 (e-mail: [dnakhaei, ppayeur]@uottawa.ca).

S.H. Tang and B. Karasfi are with University Putra Malaysia, 43300 UPM, Serdang, Selangor, Malaysia (e-mail: saihong@eng.upm.edu.my, karasfi@gmail.com).

position and the obstacle-free areas in the robot path toward the target:

Condition 1- If there is not any obstacle in a straight-line path between the robot and the actual target, the reaction layer modules generate robot's motion toward the actual target.

Condition 2- If an obstacle obstructs the robot's path toward the target, the planning layer generates a plan to move the robot away from the obstacle using an obstacle avoidance planner (OAP) or a local minimum planner (LMP) module. At the same time, the motion generation to move towards the actual-virtual target is executed by Steering control and Velocity control modules in the reaction layer.

In the next section, the design and functionalities of the layers and their modules are detailed.

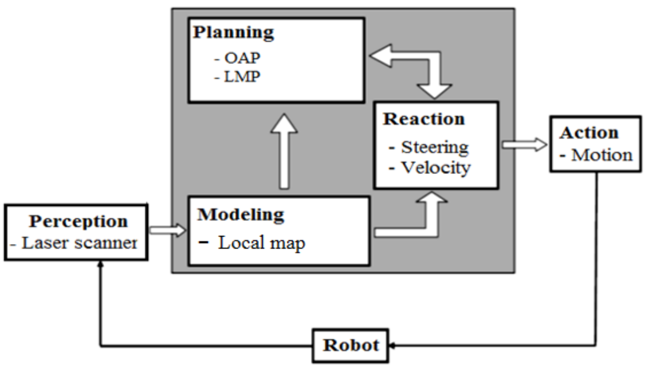


Fig. 1: Proposed hybrid control architecture.

III. Hybrid Control Architecture Design

A. Modeling Layer

This layer integrates the sensory information to construct a local model of the environment. The local model of the environment represents the obstacles distribution in a part of the work space. It updates when new information about the environment is received by the sensor. A laser range finder (LRF) (Hokuyo URG-04LX) is mounted on the robot to facilitate navigation and obstacle detection due to its high precision in indoor environments. The laser scanner detectable range is from 20mm to 5.6m (1mm resolution) in a 240° arc area scanning range (0.36° angular resolution) and takes 100msec for a complete scan. In this work, the maximal scanning range of the LRF is limited to a 180° arc, from -90^o to 90^o with respect to the robot heading direction. Therefore, there are 500 beams (180/0.36=500), with each laser beam line (W_i) representing a vector (d_i, a_i), where d_i is the distance to an obstacle and a_i is the angle of that obstacle from the robot heading. The output of each scan is a sequence of reflection points (L^P) to locate a detected object in polar (p) coordinates:

$$L^P = \{L_i^P = (d_i, a_i) | L_i^P \epsilon P; i=0,1,\ldots,W_i; 0 \le W_i \le 500\} \quad (1)$$

W_i is the number of reflection points. The detected range set (L^p) represents only the reflected points on the laser beams. A silhouette of the detected objects can be created based on the recorded ranges of d_i and a_i [14]. As shown in Fig. 3a, a reflection point (Ox_i, Oy_i) is produced by determining the intersection point between the i-th laser beam line and the surface of an object in the environment. To simplify the recorded data in polar coordinates they should be converted into Cartesian coordinates:

$$\begin{bmatrix} Ox_i \\ Oy_i \end{bmatrix} = \begin{bmatrix} d_i \cos(a_i) \\ d_i \sin(a_i) \end{bmatrix} \quad (2)$$

$$L^C = \{L_i^C = (Ox_i, Oy_i) | L_i^C \epsilon V; i=0,1,\ldots, W_i\} \quad (3)$$

Where $L_i^C = (Ox_i, Oy_i)$ represents the position data of the recorded object in Cartesian coordinates.

In presence of reflection points, the d_i value is labeled with

$$d_{min} \le d_i \le d_{max} \quad (4)$$

Where, d_{max} is the maximal range of the LRF. The maximal range of the LRF is limited to 3m. However, if there is no intersection point between the laser beam and the object surface in the environment ($d_i > d_{max}$), then $d_i = -1$. Therefore, the presence of the obstacles and obstacle-free areas are identified by checking the d_i value. The consecutive points by which the $d_i > d_{min}$ are clustered as obstacles and other points ($d_i = -1$) are clustered as obstacle-free areas (Fig. 3b).

In summary, this module integrates the sensory information and creates a local model of the robot's surroundings. Furthermore, by updating the sensory data, the changes in a dynamic environment are reflected rapidly. Next, the obstacle avoidance planner uses the obstacle-free areas to plan an optimum path toward the target based on an actual-virtual target strategy.

B. Planning Layer

The planning layer generates a set of actions that steer the robot to a desired location. This layer is developed based on an actual-virtual target strategy to avoid obstacle collision and trap situations. If there are obstacles over the straight-line path between the robot and the global target, the planner layer is applied to generate a path and guide the robot to an obstacle-free area. The planned path provides the next robot's direction, but the motion generation will be handled by the reaction layer.

This layer consists of two modules: an obstacle avoidance planner (OAP) and a local minimum planner (LMP). The OAP generates a virtual target to define an optimal collision-free path toward the global target. The LMP obtains a virtual target and computes a path to avoid local minimum trap situations. To choose a proper module dealing with obstacles, the first step is to find navigable areas among the obstacles. An area is navigable when it is wide enough so that the robot can pass through it toward the target. A navigable area is called a safe region. The safe region is computed as follows. First, a safety zone is defined around the robot bound to have more security. This safety zone is a circle with radius of **r** from the robot center. According to the obstacles position and the obstacle-free areas, the distance between obstacle edges (DOE) is calculated. Then, if DOE is greater than 2r, this area is a safe region. As shown in Fig. 4(a), there are three obstacle-free areas and only two regions are navigable (Fig. 4(b)). Since the LRF scanning area is limited to 180°, to identify safe regions at the left and right sides of the robot, it is assumed that if there is no reflection point for the W_0 or W_{499} (−90◦ or 90◦ readings), then the d_0 and d_{499} values are assumed equal to d_{max}. Fig. 4b-d illustrates the identification of the DOE and of safe regions in various situations that may occur for the robot during its navigation.

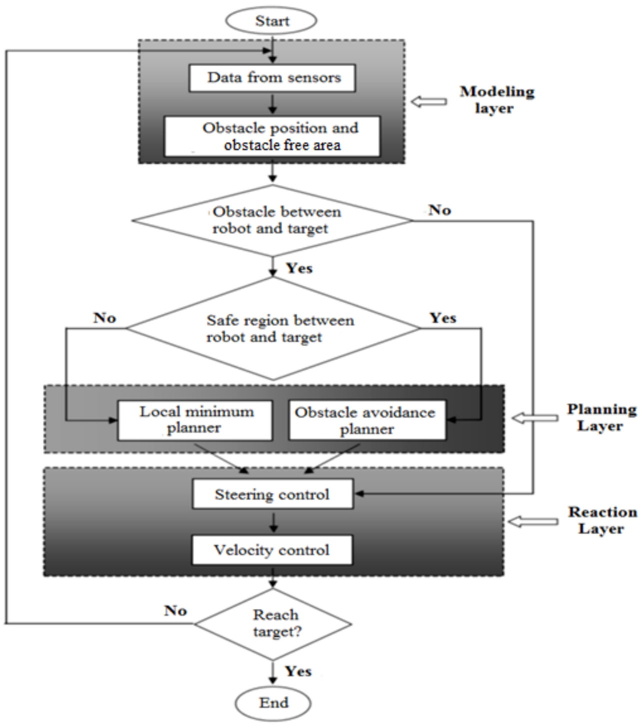
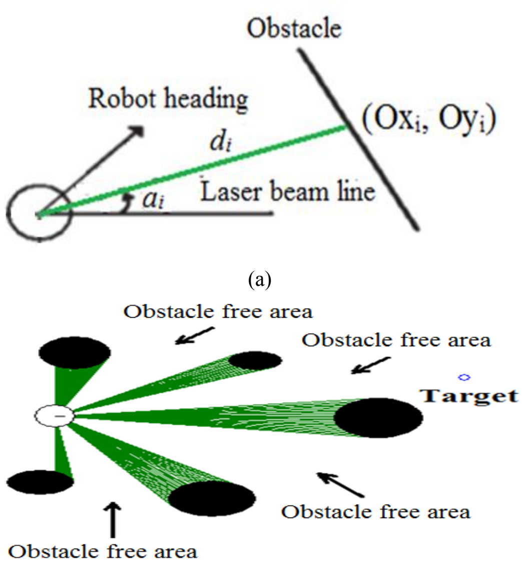


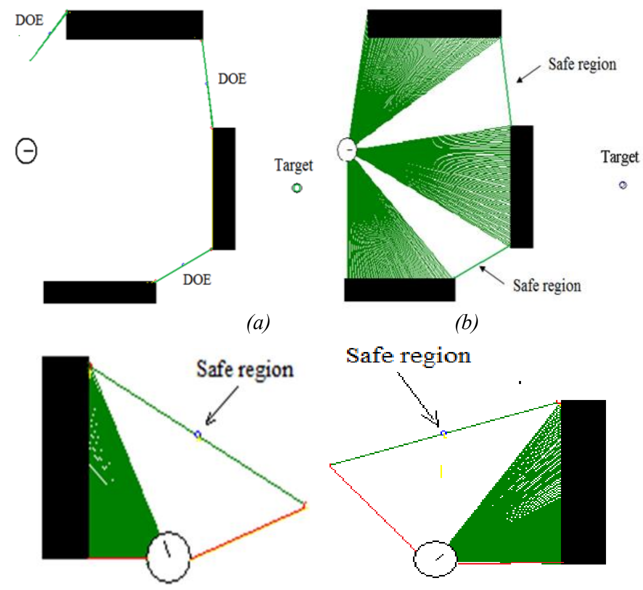
Fig. 2: Action selection algorithm.



(b)

Fig. 3: a) Definition of the robot's coordinates and intersection point, b) obstacles position and obstacle-free areas.

In such situations where there are safe regions, the OAP is activated and the nearest virtual target (NVT) method is applied to compute the collision-free path toward the target. The NVT employs a modified virtual target concept to obtain a safe path toward the global target in presence of obstacles.



(c) (d)

Fig. 4: Definition of a) DOE, and b-d) identification of safe regions in various situations.

Once the safe regions are identified, the middle point (*MP*) of each DOE is calculated. Each *MP* can be considered as a virtual target. However, in determining the priority in choosing the shortest path, the closest *MP* to the global target and the robot has the highest priority. Therefore, in order to identify the shortest path towards the global target, the distance between the robot and *MP* (*RMD*), and the distance between the target and *MP* (*TMD*) are calculated. The sum of the *RMD* and *TMD* is computed for each *MP* (Eq. 5).

$$S_i = RMD_i + TMD_i \tag{5}$$

Where $i = \{1, 2, \ldots\}$ represents the number of MPs.

The minimum value of S_i indicates the shortest path from the current robot position to the global target. Eventually, the related *MP* which generated the shortest path is chosen as the virtual target. As shown in Fig. 5, there is two *MPs* according to the two safe regions, and MP_1 has the minimum distance to the target and the robot. Therefore, MP_1 is considered as the virtual target.

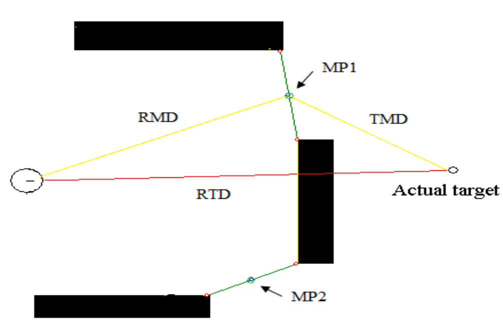
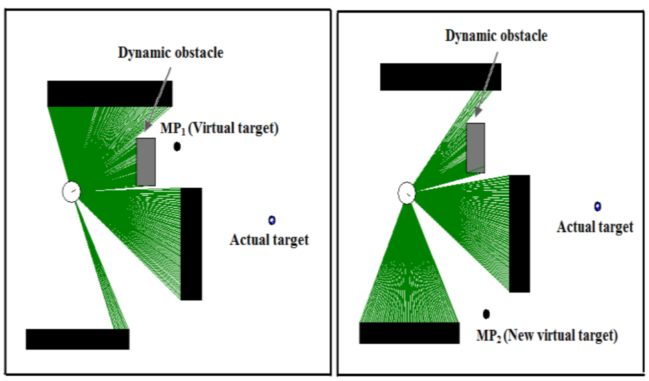


Fig. 5: MP_1 produces a virtual target.

As a result, the *NVT* generates an optimal path using virtual targets from the robot location to the destination. This instantaneous path is used to advise about the motion of the robot among obstacles without collision. Since the nature of the real world is generally full of uncertainties, it is necessary for the robot to have the capability of fast reaction to dynamic obstacles. To avoid collision with moving obstacles, each time the sensory information is updated while robot is moving toward a virtual target, wherever the straight-line between the robot and the actual-virtual target is obstructed by an unforeseen dynamic obstacle, a new virtual target is generated and the previous virtual target will be eliminated. Therefore, the robot follows a new obstacle-free path towards the target. For example, in Fig. 5, while the robot is moving toward $\boldsymbol{MP_1}$, if the instantaneous path is obstructed with a dynamic obstacle (Fig. 6a), then the robot translational velocity is reduced and $\boldsymbol{MP_2}$ is considered as a new virtual target. $\boldsymbol{MP_1}$ is eliminated and the robot turns towards $\boldsymbol{MP_2}$ **(**Fig. 6b**)**.



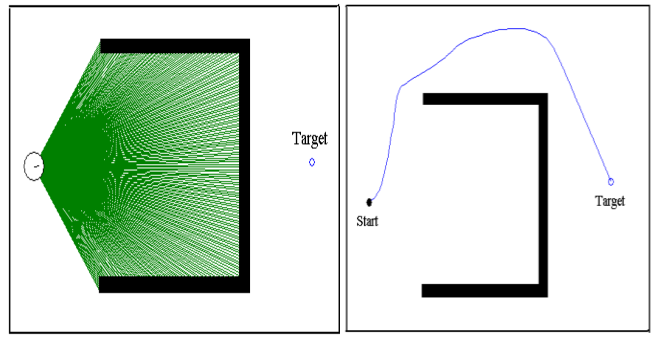
(a) (b)

Fig. 6: Dynamic obstacle avoidance: a) dynamic obstacle detected, and b) new virtual target generated.

Furthermore, a fuzzy logic controller is applied to get more safety and faster reaction, as the robot's velocity changes based on the obstacles position. Translational velocity reduces and rotational velocity increases in dealing with static or dynamic obstacles in close proximity of the robot. However, when the robot is surrounded by the obstacles and there are not any safe regions, a trap situation is likely to occur. Therefore, the LMP is responsible to plan a path to guide the robot outside the trap.

C. *Local Minimum Planner (LMP)*

A local minimum situation typically occurs only when the target is aside a long-wall, concave obstacles, or in maze-like and u-shaped environments. In this work, the local minimum situations are divided into two categories: visible and invisible. A local minimum is visible when the robot can detect the local trap situation completely, that is when the problematic configuration of space lies within the range sensor field of view and depth of field, as exemplified in Fig. 7a. When the local minimum is visible, the OAP creates a path for the robot to move away from the local minimum (Fig. 7b). However, the robot may get trapped in invisible local minimum situations, that is when the local minimum cannot be detected using the local model of the environment.



(a) (b)

Fig. 7: (a) Example of a visible local minimum; (b) the OAP steers the robot to move away from a visible local minimum.

As shown in Fig. 8, because of the sensory limitations the robot is not able to detect the local minimum completely and there is a navigable area in front of the robot to move towards the target. In such situation, the robot moves towards the inside of the local minimum and gets trapped. This section introduces a local minimum planner (LMP) using the "actual-virtual target switching" strategy to avoid the trap situations and find reliable and traversal paths towards the target. The LMP is a set of heuristic rules that require no memorizing.

Each time a local minimum trap criterion is satisfied, a new virtual target is generated and the virtual target is appointed to replace the global target temporarily until the robot gets out of the trap and reaches to the virtual target. The virtual target location is computed as a function of the distance between the robot and the current actual virtual target (RTD), the obstacles position, and the difference angle between the robot heading orientation and the relative target direction (RTA), as shown in Fig. 9.

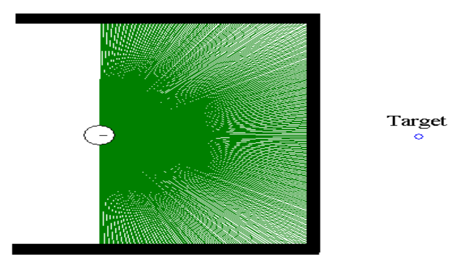


Fig. 8: Example of an invisible local minimum trap situation.

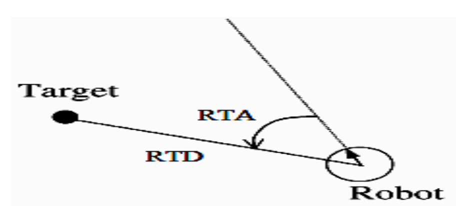


Fig. 9. Definition of RTA and RTD [7].

Since the environment is unknown and dynamic, the virtual target might be placed on an obstacle or not in a reachable location. Therefore, it is not required that the robot reaches exactly to the virtual target. Once the robot gets close to the virtual target, then the current target switches back to its previous location (either that of the global target, or that of a previous virtual target if there was one defined). Whereas a local minimum is likely to occur, the target translates and rotates around the robot center according to the obstacles configuration and the RTA. The virtual target translation (VTD) is defined using the first or last obstacles position, within its possible scanning directions (Fig. 10a). Then, if the RTA > 0, the target rotates counter clockwise, and if RTA<0, then the target rotates clockwise about the robot center. Therefore, the new virtual target location is calculated as follows:

$$\textbf{\textit{If}} \quad RTA > 0 \quad \textbf{\textit{then}}$$
$$\boldsymbol{\Theta} = RTA \text{ and } \textbf{\textit{VTDx}} = ROL_x + \alpha_X\,,\ \ \textbf{\textit{VTDy}} = ROLy + \alpha_y \tag{6}$$

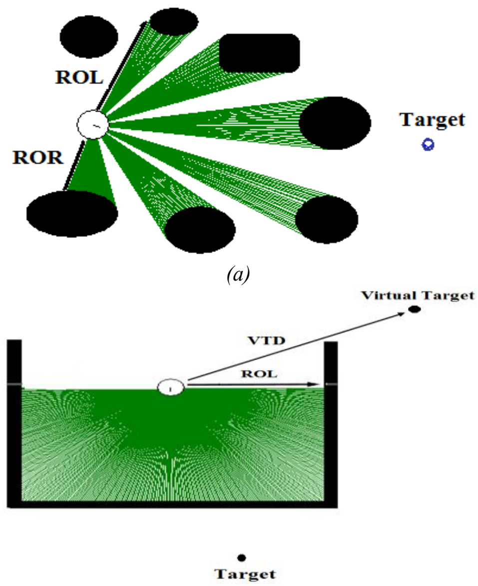
$$\textbf{\textit{If}} \quad RTA < 0 \quad \textbf{\textit{then}}$$
$$\boldsymbol{\Theta} = RTA \text{ and } \textbf{\textit{VTDx}} = ROR_x + \alpha_X,\ \textbf{\textit{VTDy}} = RORy + \alpha_X \tag{7}$$

$$\begin{aligned} \boldsymbol{\alpha}_{x} &= \alpha\cos(\Theta_O) \\ \boldsymbol{\alpha}_{y} &= \alpha\sin(\Theta_O) \end{aligned} \tag{8}$$

Where the α parameter is an experimentally determined distance to locate the virtual target out of the trap with a safe distance from the obstacles (in this work α= 60 cm), ROR and ROL are detected obstacles on the left side and the right side of the robot respectively. Θ_O is the ROR or ROL angle with respect to the base frame and Θ is the rotation angle about the robot center. The new i^{th} virtual target location can be computed as follows:

$$\begin{bmatrix} T_{xi} \\ T_{yi} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_R \\ 0 & 1 & Y_R \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cos\,\theta & -Sin\,\theta & 0 \\ Sin\,\theta & Cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_R \\ 0 & 1 & -Y_R \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} VTDx \\ VTDy \\ 1 \end{bmatrix} \tag{9}$$

where $i = \{1,2,\ldots.\}$ shows the number of virtual targets created each time a new trap is detected; T_{xi} and T_{yi} are the i-th virtual target coordinates, X_R and Y_R are the robot coordinates, T_{x0} and T_{y0} refer to the actual global target coordinates which are defined by the user, and Θ is the rotation angle (Fig. 10b).



(b)
Fig. 10. a) ROL and ROR definition, b) virtual target position in a local minimum situation.

D. Reaction Layer

The reaction layer generates the robot's motion based on the model of the world or the path generated by the planning layer. The reaction layer steers the robot to reach the actual virtual target. Whether the robot should execute the obstacle avoidance, local minimum avoidance or target seeking, the modeling and planning layers are to recognize the plan and send it to the reaction layer. In other words, the planning and the modeling modules provide the input for the reaction layer. The modeling module provides information about the obstacle position (OP). The planning module obtains the virtual target information for the reaction layer. The outputs of this layer are the direction of the motion and the velocity. In the proposed hybrid architecture, the reaction layer consists of two main modules: the steering control module and the velocity control module.

The steering control module is proposed to compute the motion direction. This module enables the robot to change the direction of travelling, which depends on the actual-virtual target direction. The modeling and planning layers provide input for this module. The input of this layer is the actual-virtual target direction. This module output (RTA) is the angle between the robot current heading direction and the target direction. The RTA is used as the reference for the robot's steering command (Fig. 9). The value domain of RTA is [-180°, 180°].

The velocity control is responsible for the control of the robot's velocity. A proper way to control the velocity of the robot is to use a fuzzy logic controller (FLC). The proposed fuzzy controller [6] has two inputs and two outputs. The FLC inputs are the obstacle position (OP) and the obstacle distance (OD). For 3-set partitioning of the OP and 5-set partitioning of the OD the fuzzy rules base contains 15 rules (Table 1). After fuzzyfication of inputs, the fuzzy inference converts fuzzy input sets to outputs. These fuzzy outputs are the rotational velocity (R_V) and the translational velocity (T_V). The rotational and translational velocities change according to the obstacles distribution. Where the robot is not surrounded with obstacles and the workspace is not very dense and cluttered, the robot can move with a higher speed towards the target in areas free of obstacles. However, the robot speed is reduced in the presence of obstacles to prevent collision with them over the robot path towards the target.

TABLE 1. The Fuzzy Rule Base.

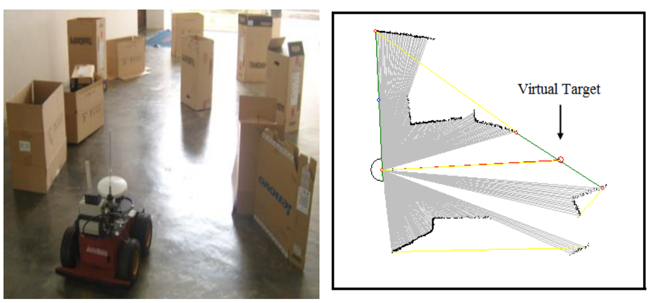
If				Then		
Rule no.	OP		OD	Rv		Tv
1	D	&	VN	Large	&	Slow
2	U	&	N	Medium	&	Medium
3	S	&	M	Smooth	&	Fast
4	D	&	F	Large	&	Medium
5	U	&	VF	Medium	&	Fast
6	S	&	VN	Medium	&	Medium
7	D	&	N	Large	&	Slow
8	U	&	M	Medium	&	Medium
9	S	&	F	Smooth	&	Fast
10	D	&	VF	Large	&	Medium
11	U	&	VN	Medium	&	Slow
12	S	&	N	Smooth	&	Medium
13	D	&	M	Large	&	Slow
14	U	&	F	Medium	&	Medium
15	S	&	VF	Smooth	&	Fast

IV. Setting and Experimental Results

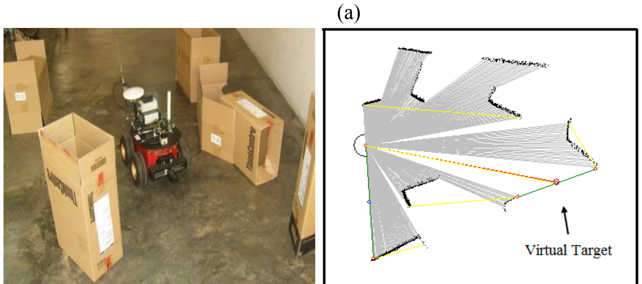
To validate that the proposed approach complies with the objectives of this work, some representative results are carried out through real robot experiments. The experimentation was conducted on an ActivMedia P3AT robot in unknown and dynamic environments. The P3AT is a 4-wheel drive rectangular shaped holonomic vehicle from ActivMedia Robotics. The maximum translational velocity is set to 30 cm/s and the maximum rotational velocity is set to 60 deg/s.

Experiment 1: *Motion in very dense, cluttered and complex scenario.*

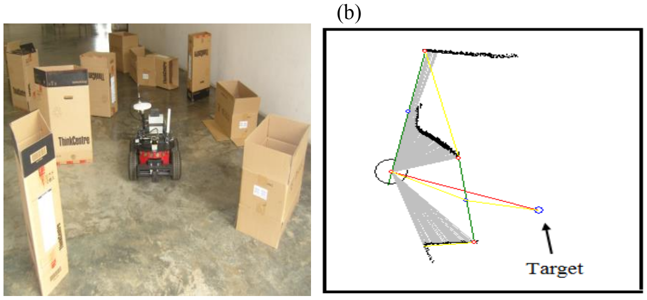
In this experiment, the robot navigated in a dense, complex, and cluttered environment among very close obstacles (Fig. 11). Where the robot detects obstacles, the selection of navigable areas is carried out by the OAP module and a number of virtual targets are generated until there is not anymore obstacle in the robot's path toward the global target (Fig. 11a-c). As shown in Fig. 11, the robot successfully passed through a narrow passage towards the target while choosing the shortest path. Since there is little space for the robot to maneuver, the robot moves slowly among the obstacles, while when there is no obstacle on its way it moves faster (Fig. 11d).



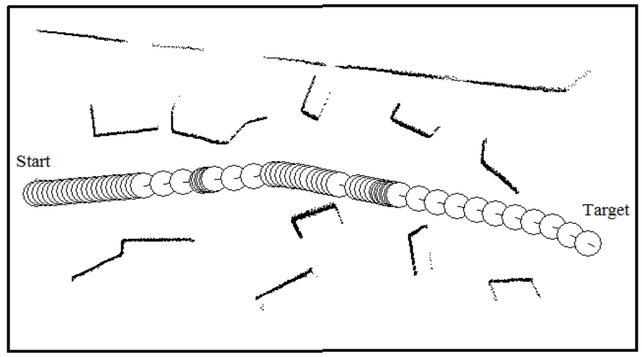
(a)



(b)



(c)

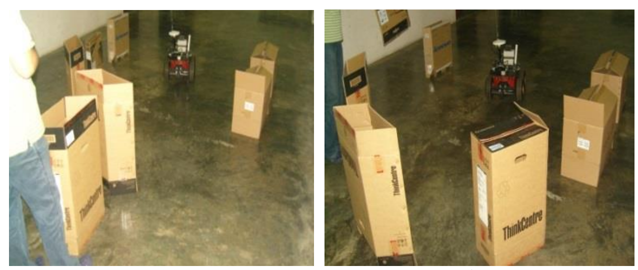


(d)

Fig. 11. Robot trajectory in a very dense, cluttered and complex scenario.

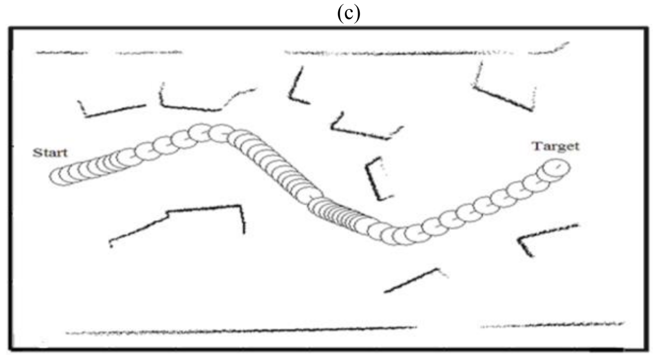
Experiment 2: *Dynamic environment*

This example demonstrates the robot performance when dealing with dynamic obstacles (Fig. 12). As shown in Fig. 12a, while the robot is moving towards the global target, a dynamic obstacle (Fig. 12b) obstructs the robot's path towards the target. At the same time another obstacle on the right side of the robot is removed. Therefore, a new virtual target is generated at this point using the updated sensory information (Fig. 12c) and the robot changes its heading toward the new virtual target. Fig. 12d shows how the robot successfully passes through the moving obstacles and reaches the global target.



(a) (b)

(c)

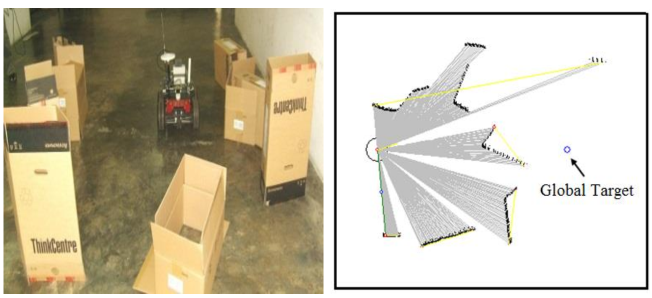


(d)

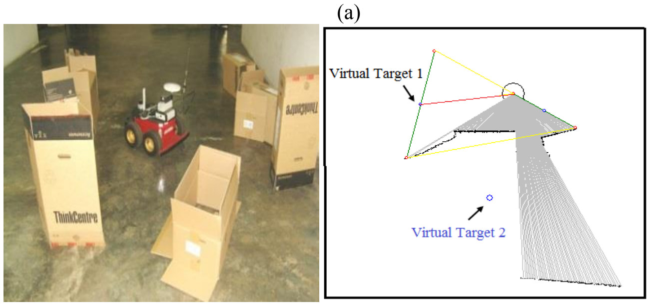
Fig.12. Robot performance in a dynamic environment.

Experiment 3: *Avoiding local minimum situation*

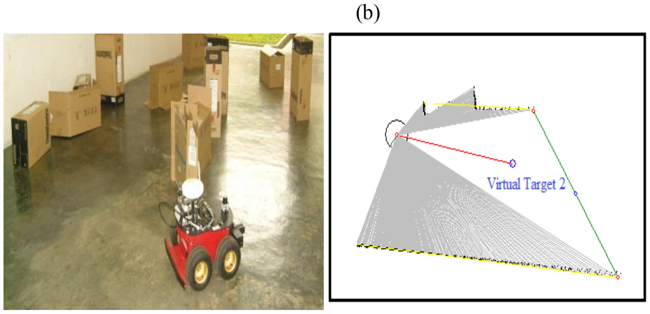
This example highlights the robot performance where it is surrounded by obstacles and there is not enough space for the robot to pass among the obstacles towards the target (Fig. 13a). The situation is considered as a trap situation and the local minimum planner is responsible to generate a virtual target outside of the trap. Therefore, the LMP generates a virtual target (virtual target 2) which temporarily replaces the global target (Fig. 13b) and then the program switches to the obstacle avoidance mode and the OAP steers the robot towards the new virtual target by generating some more virtual targets (virtual target 1) in safe regions according to the updated sensory information (Fig. 13b). Once the robot reaches the virtual target 2, the target switches back to its previous location (Fig. 13c) and the robot moves toward the global target (Fig. 13d). The fuzzy controller is applied to increase safety while dealing with obstacles and reducing the navigation time. Fig. 14 shows the robot steering and velocity profiles in experiment 2. The robot is moving smoothly toward the target and it has minimum oscillation (Fig. 14a). The rotational and translational velocities are also changed according to the obstacles distribution (Fig. 14b-c).



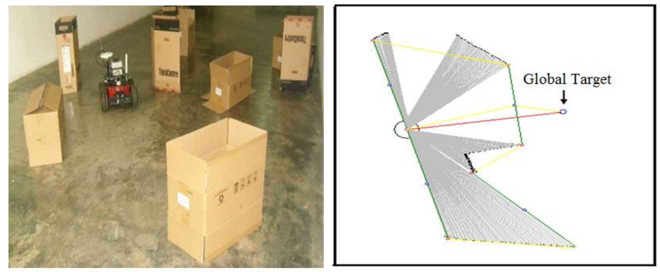
(a)



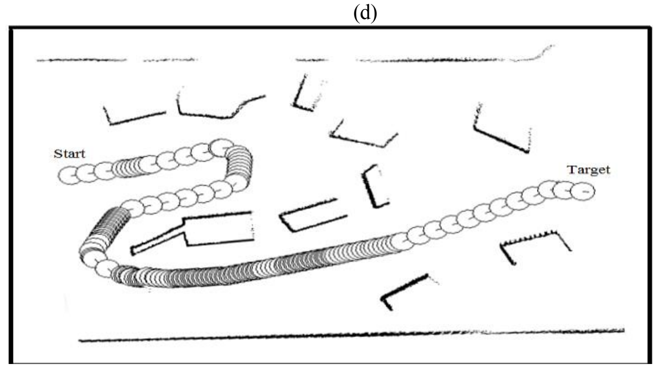
(b)



(c)

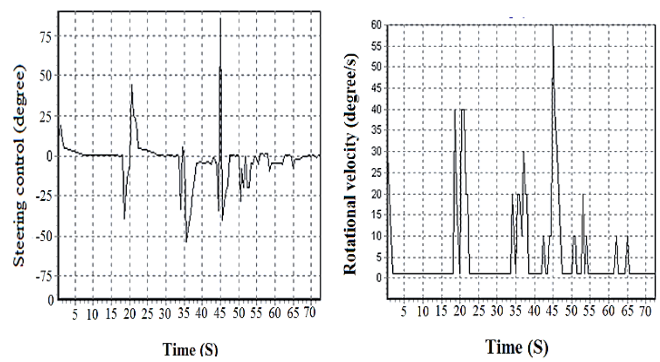


(d)

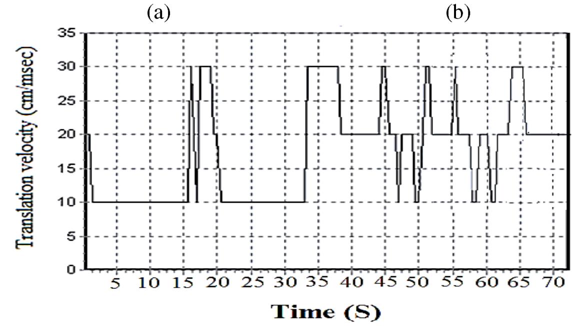


(e)

Fig. 13. Trajectory executed when the robot is surrounded by obstacles creating a local minimum.



(a) (b)



(c)

Fig.14. a) Steering control, b) rotational velocity, c) translational velocity for experiment 2.

Comparison of performance of the proposed approach with some related works shows that most of the existing hybrid control architectures have difficulties for driving in very dense, cluttered and complex scenarios due to the typical limitations of their methods [15]. In some hybrid systems [16], the planning layer is used to compute the motion and perform an "any-time" planning instead of the reaction layer. In such architectures, the modeling and the planning layers can be synchronous or asynchronous. However, they do not benefit from the reflexive and responsive part of the reaction module responding to unforeseeable circumstances in unknown environments. While alternative hybrid control architectures used reactive method for motion generation or obstacle avoidance and offered better performance in troublesome scenarios, they suffer from local minimum traps [17, 18], oscillations in dense scenarios [19, 20], and the inability to obtain shortest paths [21].The proposed hybrid control architecture benefits from the advantages of the reactive and deliberative layers which couple the high level motion planning with the low level one. The methods and techniques applied for each layer result in a reliable, safe and robust motion in troublesome scenarios.

V. Conclusion

The navigation results demonstrate that the integration of the three layers generates a robust motion. First, the modeling layer creates the local model of the environment. Based on the obstacles configuration, the planning layer generates some virtual targets in obstacle-free areas to avoid obstacles and trap situations. Then the reaction layer steers the robot to move toward the actual virtual target. Eventually, the interaction and cooperation between the OAP module, the LMP module in the planning layer and the fuzzy controller in the reaction layer improve the navigation performance.

References

[1] Nakhaeinia, D., Tang, S.H., Mohd Noor, S.B. and Motlagh, O., “A review of control architectures for autonomous navigation of mobile robots (2011),” International Journal of Physical Sciences, Vol. 6 (2), pp. 169-174.

[2] Laird, J. E. and Rosenbloom, P. S. (1990). Integrating execution, planning, and learning in SOAR for external environments, in AAAI, pp. 1022-1029.

[3] Diéguez, A.R., Sanz. R., and López, J. (2003). Deliberative On-Line Local Path Planning for Autonomous Mobile Robots, Journal of Intelligent and Robotic Systems, 37 (1), 1 – 19.

[4] Ibrahim, M.T.S., Ragavan, S.V., Ponnambalam, S. G., "Way point based deliberative path planner for navigation," Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on, pp.881-886, 14-17 July 2009.

[5] Brooks, R. (1986). A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2:14-23.

[6] Nakhaeinia, D. and Karasfi, B. (2012). A Behavior-Based Approach for Collision Avoidance of Mobile Robot in Unknown and Dynamic Environments. Journal of intelligent and fuzzy systems, 24 (2): 299-311. DOI 10.3233/IFS-2012-0554.

[7] Motlagh, O., Nakhaeinia, D., Tang, S.H, and Karasfi, B. (2013), Automatic Navigation of Mobile Robots in Unknown Environments, Neural Comput. & Applic., 24 (7-8), pp. 1569-1581.

[8] Belkhouche, F., "Reactive Path Planning in a Dynamic Environment," Robotics, IEEE Transactions on, vol. 25, no. 4, pp. 902-911, Aug. 2009. doi:10.1109/TRO.2009.2022441.

[9] Tang S.H., Nakhaeinia, D., Karasfi, B., and Motlagh, O. (2013). A reactive collision avoidance approach for mobile robot in dynamic environments. Journal of Automation and Control Engineering, Vol. 1, No. 1, pp. 16-20.

[10] Minguez, J. and Montano, L. (2005). Sensor-based robot motion generation in unknown, dynamic and troublesome scenarios. Robotics and Autonomous Systems 52:290-311.

[11] De Carvalho Santos, V., Motta Toledo, C.F., and Osorio, F.S., "A Hybrid Approach for Path Planning and Execution for Autonomous Mobile Robots," Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014 Joint Conference on , pp.124-129, 2014. doi: 10.1109/SBR.LARS.Robocontrol.2014.44

[12] Pshikhopov, V.K. and Ali, A.S. "Hybrid motion control of a mobile robot in dynamic environments," Mechatronics (ICM), 2011 IEEE International Conference on, pp. 540-545, 13-15 April 2011.

[13] Tang, S.H., Nakhaeinia, D., and Karasfi, B. (2012). Application of Fuzzy Logic in Mobile Robot Navigation. Fuzzy Logic - Controls, Concepts, Theories and Applications, ISBN 978-953-51-0396-7. InTech Publisher.

[14] Nakhaeinia, D., Tang, S.H., and Payeur, P. “Development of a Sensor-Based Approach for Local Minima Recovery in Unknown and Dynamic Environments”, Robotic and Sensors Environments (ROSE), 2013 IEEE International Symposium on. pp. 166-171.

[15] Krishna, K. M. and Kalra, P. K. (2001). Perception and remembrance of the environment during real-time navigation of a mobile robot. Robotics and Autonomous Systems 37:25-51.

[16] Xu, W. L., Tso, S. K., and Lu, Z. K. (1998). “A Virtual Target Approach for Resolving the Limit Cycle Problem in Navigation of a Fuzzy Behaviour-based Mobile Robot”, in Conference on Intelligent Robots and Systems, Victoria, B.C., Canada.

[17] Stachniss, C. and Burgard, W. (2002). An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments, in IEEE-RSJ International Conference on Intelligent Robots and Systems, Switzerland, pp. 508-513.

[18] Nakhaeinia, D., Tang, S.H., Karasfi, B., Motlagh, O., and Ang, C.K (2011). A virtual force field algorithm for a behaviour-based autonomous robot in unknown environments. Part I: J. Systems and Control Engineering, Vol. 221 (1), pp. 51-62.

[19] Giesbrecht, J., Global Path Planning for Unmanned Ground Vehicles. Technical Memorandum, Defense Research and Development Suffield (ALBERTA), 2004.

[20] Ge, S.S. and Cui, Y.J. (2000). New potential functions for mobile robot path planning. IEEE Transactions on Robotics and Automation, 16, 5 (Oct. 2000), 615-620.

[21] Wang, H. J. and Xiong, W. (2009). Research on global path planning based on ant colony optimization for AUV. Journal of Marine Science and Application, 8: 58-64.