VOL. 10, NO. 18, OCTOBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A FUZZY LOGIC POSITIONAL-BASED CONTROLLER FOR SENSOR-BASED ROBOTIC MOTION PLANNING

Weria Khaksar[1], Moslem Yousefi[1], Khairul Salleh Mohamed Saharia[1] and Firas B. Ismail[2]
[1]Center of Advanced Mechatronics and Robotics, Universiti Tenaga Nasional (UNITEN), Kajang, Selangor, Malaysia
[2]Center of Power Generation, Universiti Tenaga Nasional (UNITEN), Kajang, Selangor, Malaysia
E-Mail: weria@uniten.edu.my

ABSTRACT

Motion planning is a critical task for any robotic system. Despite the large volume of research in this field, the computational complexity of the motion planning and the increasing potential application domains require more accurate and efficient motion planning algorithms. In this paper, a novel positional-based motion controller is proposed which requires very little amount of information from the robot's sensory system. The proposed controller evaluates candidate positions of the robot and selects the most promising ones according to the output of a fuzzy logic controller. The designed controller defines three positional variables for destination evaluation in order to improve the path length and runtime of the solutions. The performance of the proposed planner is tested through computer simulation in different types of environments. Simulation studies indicate the efficiency and robustness of the proposed algorithm.

Keywords: robotics, motion planning, fuzzy logic, sensor-based, path length, runtime.

INTRODUCTION

Path planning is regarded as one of the primary challenges encountered within applied robotics which has been studied by many researchers over the past few decades resulting in various algorithms with different specifications. Path planning for a mobile robot is the process of finding a path from an initial position to a final one, while avoiding any collision with obstacles. In the simplest form of path planning problem, i.e. the piano mover's problem, the running time of any algorithm is exponentially the degrees of freedom, which makes the path planning problem NP-Complete. It is also shown that the path planner requires memory exponential in the degrees of freedom which indicate that the problem is PSPACE-Complete (Tsianos, Sucan, & Kavraki, 2007). When the environment is completely known for the robot, the path planning problem becomes a geometric programming problem, where there are a variety of techniques proposed to solve this problem (Ebrahimi, Janabi-Sharifi, & Ghanbari, 2014; Khoukhi, 2015; O Motlagh, Jamaludin, Tang, & Khaksar, 2013; Omid Motlagh, Nakhaeinia, Tang, Karasfi, & Khaksar, 2014; Omid Motlagh, Tang, Khaksar, & Ismail, 2012; Nayl, Nikolakopoulos, & Gustafsson, 2015; Vatcha & Xiao, 2014). A detailed review on these techniques can be found in (Tang, Khaksar, Ismail, & Ariffin, 2012). When the planner has no prior information about the environment, the path planning is called online, local, or sensor-based path planning. In an unknown environment, the robot obtains local information by means of a sensory system. A basic difference between these two types of path planning problems is that a path can be preplanned in known environment, but in an online path planning, path should be progressively computed as the robot explore the environment and obtain newer knowledge about its surrounding area (Choset, 2005).

The available sensor-based motion planning algorithms generally require extensive information about the surrounding area of the robot within its sensing range which normally results in high compuataional cost. On the other hand, as the degreed of freedom increase, the complexity and difficulty of the motion planning problem dramatically increases and more accurate, simple and cost-efficient algorithms are required to deal with real-life problems.

In this paper a new sampling-based navigation algorithm is proposed which is able to plan navigation in complex environments and also can handle sensor-based planning tasks. A fuzzy controller is applied to make the sampling procedure more intelligent. This controller evaluates the generated samples within the visible region and selects the best samples according to some criteria in order to solve sensor-based planning problems in complex environments. These criteria are designed based on three positional variables and are being converted to fuzzy variables to form the fuzzy reasoning system. The fuzzy controller considers three criteria to evaluate the samples, including distance of the robot's next position to the goal (μ_G), distance of the robot's next position to its previous positions (μ_(N-P)) and finally the robot current position's distances to the start position (μ_S). Based on these criteria, three variables will be defined and used in a single package for candidate positions evaluation. The simple logic behind the concept of these variables is that, the robot needs to get closer to the goal, get farther from the start and also not get close to its previous position.

Problem formulation

The robot is a 2D circle freely moving in a workspace filled with different shaped obstacles. The robot is equipped with a range finder which can sense the distance to the surrounding area within the reading range of the sensors. The reading of the range finder can be formulized as follows:

$$\omega(C,\theta) = \gamma \times \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \quad (1)$$

In which, γ is the reading range of the range finder. The value of $\omega(C,\theta)$ shows the distance of the robot's current position (C) to the closest obstacle along the direction of (θ) and if there is no obstacle in that direction, the distance equals to the reading range. Using this value, the robot is able to identify three important parameters including the distance of each position of the robot to the start configuration, the distance to the goal position and the interlude between next and previous positions. Figure-1 illustrates the sensing mechanism of the robot.

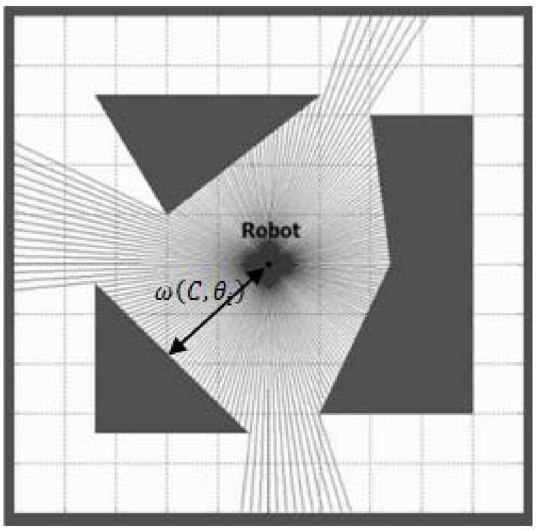

Figure-1. The perception procedure of the robot placed at point *C* in the middle of the environment.

Each time the robot moves, a new point on the perimeter of the reading range will be selected randomly according to a uniform distribution. Then, this point will be checked by a fuzzy logic controller based on the aforementioned variables and if this random point was decided to be suitable, the robot will move to this new coordinates instantly.

The proposed fuzzy logic controller

The proposed controller utilizes the fuzzy logic to evaluate the next position of the robot. This controller includes three input variables and one output variable all with bell-shaped membership functions. These fuzzy variables include μ_S, μ_{N-P} and μ_G which can be formulized as follows:

$$X_1 = \mu_G(New) - \mu_G(Current) \quad (2)$$

$$X_2 = \mu_{N-P} \quad (3)$$

$$X_3 = \mu_S(New) - \mu_S(Current) \quad (4)$$

As can be seen, these variables have different ranges and to simplify the fuzzy logic controller, they will be normalized as follows to all fall in the interval $[0,1]$.

$$X_1^{norm} = (X_1 + \gamma)/2\gamma \quad (5)$$

$$X_2^{norm} = X_2/2\gamma \quad (6)$$

$$X_3^{norm} = (X_3 + \gamma)/2\gamma \quad (7)$$

By defining these variables, the fuzzy logic controller is able to make sure that the robot is always getting closer to the goal, getting farther from the start and keeping distance from its previous position. Figure-2 shows the input membership functions.

All input and output variables follow similar bell-shaped membership functions. For X_1^{norm}, if the value is *Close*, then the robot is getting closer to the goal position. For X_2^{norm}, if the value is *Close*, then the robot is getting close to its previous position anf finally for X_3^{norm}, if the value is *Closer*, then the robot is getting closer to the start position. For the output value, i.e. Suitability of the randomly selected position, the position will be selected if its value is more than 0.7 and rejected otherwise.

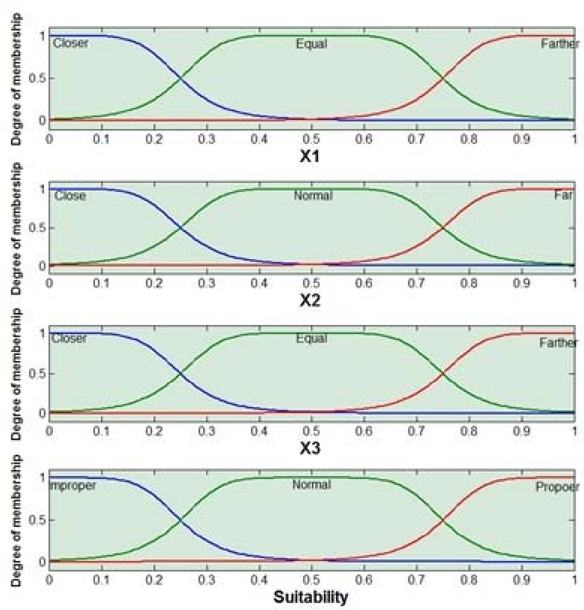

Figure-2. The corresponding membership functions for the inputs and output variables.

Figure-3 shows the rule-base surfaces for all pair of input variables against the output variable. Note that for

VOL. 10, NO. 18, OCTOBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

any pair of inputs, the third variable is set to be 0.5 in order to have standard and uniform surfaces.

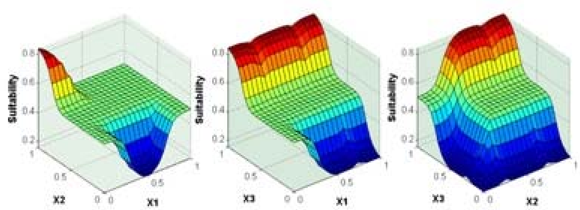

Figure-3. Corresponding decision surfaces based on rule-base system of the fuzzy logic controller.

The overall procedure of the algorithm is as follows: The robot scans the surrounding area and calculates the value of three input variables. Then the fuzzy logic controller takes place and calculates the suitability of the randomly selected position. If the output of the fuzzy system is below 0.7, the position will be rejected and otherwise, the robot will move directly to this position. This procedure will be repeated until the goal is reached.

Simulation results

The performance of the proposed algorithm was tested in several simulated environments in MatLab and also was compared to four different pioneer motion planning algorithms. Four instances of the resulted solutions are shown in Figure-4.

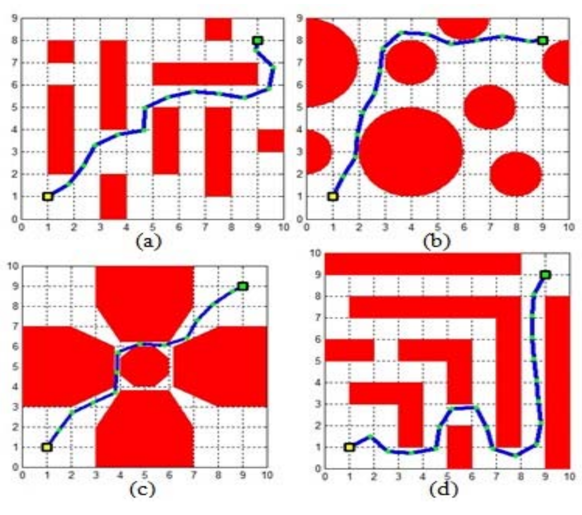

Figure-4. Some instances of the simulation results in 2D environments filled with different types of obstacles.

As can be seen in Figure-4, the generated paths are almost optimum, safe and smooth in all four environments. In the third environments where the presence of narrow corridors makes the motion planning problem even more difficult, the proposed fuzzy logic controller efficiently finds the entrance of the narrow region and navigates the robot through the narrow passage.

The summary of the comparison studies are presented in Table-1, where the path length and runtime of different planners are given in all test environments according to Figure-4.

Table-1. The summary of the simulation results in all test environments.

	Environments-1		Environments-2		Environments-3		Environments-4	
Algorithm	PL	RT	PL	RT	PL	RT	PL	RT
Bug I	25.36	14.87	21.49	12.74	17.20	8.05	34.66	16.15
Bug II	21.18	15.66	18.65	12.84	15.69	6.47	21.91	14.58
Tangent-Bug	15.98	18.17	13.25	16.67	13.70	8.32	19.88	17.18
Tabu-Based	14.55	21.45	13.13	17.52	14.56	8.64	22.52	18.27
Proposed	14.64	8.66	12.95	9.32	12.48	5.85	18.07	11.35

For comparison studies, for different motion planning algorithms have been included as stated in Table 1. These algorithms are Bug I, Bug II, Tangent-Bug (Choset, 2005) and Tabu-based (Masehian & Amin-Naseri, 2008) algorithms. They were selected for their powerful performances in sensor-based motion planning tasks. A comprehensive review of these algorithms van be found in (Tang *et al.* 2012). The results of Table-1 are visualized in Figure-5 and it shows the superiority of the proposed algorithm in terms of path length and runtime. Note that the results in Figure-5 are averaged over all test environments to provide a more general and uniform analysis of performances for all five algorithms.

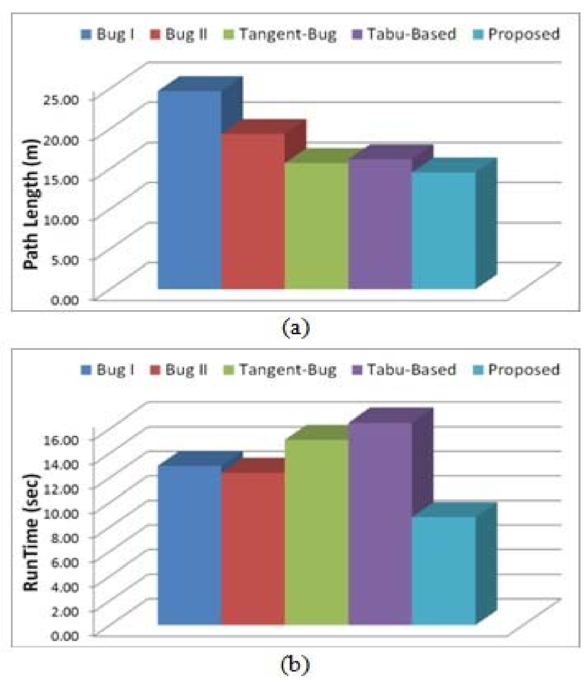

Figure-5. Performance comparison in terms of (a) path length and (b) running time. The results are averaged over all environments.

CONCLUSIONS

A novel fuzzy logic controller was proposed in this paper to deal with the challenging problem of motion planning in unknown environment. The proposed planner defines three variables according to the positional situation of the robot and evaluates the candidate destinations accordingly. The randomly generated candidate will be selected as the robot's next position if the output of the corresponding fuzzy system is more than 0.7. This value has been obtained over an extensive amount of simulation studies.

The performance of the proposed controller was tested in different environments and shown significant improvements in terms of length of the generated path and rinning time of the planner. Furthermore, the results of the introduced algorithm have been compared with four powerful sensor-based navigation algorithms to proof the superiority of the proposed fuzzy logic controller.

REFERENCES

[1] Choset H. M. *et al*. 2005. Principles of robot motion: theory, algorithms, and implementation: MIT press.

[2] Ebrahimi A., Janabi-Sharifi F. and Ghanbari A. 2014. UavisBug: vision-based 3D motion planning and obstacle avoidance for a mini-UAV in an unknown indoor environment. Canadian Aeronautics and Space Journal, Vol. 60, No. 01, pp. 9-21.

[3] Khoukhi A. 2015. Data-driven multi-stage multi-objective motion planning of mobile robots, application to near minimum power fuzzy parking. Computers & Electrical Engineering.

[4] Masehian E. and Amin-Naseri M. R. 2008. Sensor-based robot motion planning-A Tabu search approach. Robotics & Automation Magazine, IEEE, Vol. 15, No. 2, pp. 48-57.

[5] Motlagh O., Jamaludin Z., Tang S. and Khaksar W. 2013. An agile FCM for real-time modeling of dynamic and real-life systems. Evolving Systems, pp. 1-13.

[6] Motlagh O., Nakhaeinia D., Tang S. H., Karasfi B. and Khaksar W. 2014. Automatic navigation of mobile robots in unknown environments. Neural Computing and Applications, Vol. 24, No. 7-8, pp. 1569-1581.

[7] Motlagh O., Tang S. H., Khaksar W. and Ismail N. 2012. An alternative approach to FCM activation for modeling dynamic systems. Applied Artificial Intelligence, Vol. 26, No. 8, pp. 733-742.

[8] Nayl T., Nikolakopoulos G. and Gustafsson T. 2015. Real-Time Bug-Like Dynamic Path Planning for an Articulated Vehicle Informatics in Control, Automation and Robotics, pp. 201-215, Springer.

[9] Tang S., Khaksar W., Ismail N. and Ariffin M. 2012. A review on robot motion planning approaches. Pertanika Journal of Science and Technology, Vol. 20, No. 1, pp. 15-29.

[10] Tsianos K. I., Sucan I. A. and Kavraki L. E. 2007. Sampling-based robot motion planning: Towards realistic applications. Computer Science Review, Vol. 1, No. 1, pp. 2-11.

[11] Vatcha R. and Xiao J. 2014. Practical Motion Planning in Unknown and Unpredictable Environments. Paper presented at the Experimental Robotics.