A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation

Xiaoyu Yang, *Student Member, IEEE*, Mehrdad Moallem, *Member, IEEE*, and Rajni V. Patel, *Fellow, IEEE*

***Abstract*—Most conventional motion planning algorithms that are based on the model of the environment cannot perform well when dealing with the navigation problem for real-world mobile robots where the environment is unknown and can change dynamically. In this paper, a layered goal-oriented motion planning strategy using fuzzy logic is developed for a mobile robot navigating in an unknown environment. The information about the global goal and the long-range sensory data are used by the first layer of the planner to produce an intermediate goal, referred to as the way-point, that gives a favorable direction in terms of seeking the goal within the detected area. The second layer of the planner takes this way-point as a subgoal and, using short-range sensory data, guides the robot to reach the subgoal while avoiding collisions. The resulting path, connecting an initial point to a goal position, is similar to the path produced by the visibility graph motion planning method, but in this approach there is no assumption about the environment. Due to its simplicity and capability for real-time implementation, fuzzy logic has been used for the proposed motion planning strategy. The resulting navigation system is implemented on a real mobile robot, Koala, and tested in various environments. Experimental results are presented which demonstrate the effectiveness of the proposed fuzzy navigation system.**

***Index Terms*—Autonomous navigation, fuzzy logic, mobile robot, motion planning.**

I. Introduction

IN MOBILE robotics, creating autonomous robots is one of the major undertakings. By a real-world mobile robot, we mean a robot that needs to operate in an environment which is not especially engineered for the robot. In other words, any prior knowledge about the environment is limited and unreliable because of the complexity and unpredictable dynamics of the environment. Thus the ability of a robot to plan motions autonomously is of paramount importance.

Traditional methods for mobile robot motion planning, referred to as *model-based approaches*, use a model of the environment to generate a path for the robot to follow. Techniques for model-based path generation include road mapping, cell decomposition, and the potential field method [1]. Among road mapping methods, the *visibility graph* algorithm is one of the earliest path planning methods [2] and a main area of research in computational geometry [3]. In the visibility graph method, a semi-free path is constructed as a simple polygonal line, connecting the initial configuration q_{init} to the goal configuration q_{goal} using vertices of the union of all obstacles [1]. The advantage of the visibility graph method is that the resulting graph can be searched for the shortest semi-free path between q_{init} and q_{goal} according to the Euclidean metric in $\mathbb{R}^2$. This path is guaranteed to be found if it exists. But the visibility graph method, like other model-based approaches, would fail when dealing with the navigation problem of a real-world mobile robot. The reason is that it is usually difficult or impossible to obtain an accurate model of a dynamic environment. For the mobile robot navigation problem in unknown environments, there exist *sensor-based approaches* which generate control commands based on sensory data [4]–[7]. The main advantage of sensor-based approaches is that the robot can navigate safely in a dynamic environment by reacting to obstacles detected by sensors in real time. A major drawback is that due to the limitation of sensors the robot may get lost even if a path to the goal exists.

To develop algorithms for mobile robot navigation in an unknown environment, the following points should be considered.

1) The mathematical model of the environment is generally unavailable.
2) Sensory data are uncertain and imprecise due to noise.
3) Real-time operation is essential.

In this regard, fuzzy-logic based algorithms have been proposed for designing robust controllers that are able to deliver satisfactory performance in face of large amounts of parameter variations and noise. In addition, due to its simplicity of implementation, fuzzy logic control is well suited for autonomous robotics. Fuzzy logic has been utilized in navigation systems for mobile robots for almost two decades. The first reported uses of fuzzy control in mobile robotics belong to reactive approaches. In 1985, Sugeno and Nishida developed a fuzzy controller to drive a model car along a track delimited by two walls [9]. Shortly after, Takeuchi *et al.* [10] used fuzzy logic control in the obstacle avoidance behavior of mobile robots. Later in 1991, Yen and Pfluger [11] proposed a method of path planning and execution using fuzzy logic for mobile robot control. From then on, the efficiency of using fuzzy logic in mobile robot navigation systems has been well demonstrated [12]–[14]. A comprehensive study of fuzzy logic-based autonomous robot navigation systems is given in [8]. Recently, several new and improved solutions to the mobile robot navigation problem in unknown environments based on fuzzy logic have been proposed [7], [15], [16], [18], extensively demonstrating that the interpolative nature of fuzzy control results in smooth movement of the robot and graceful degradation in face of errors and fluctuations in sensory data. The goal-unreachable problems in reactive fuzzy

Manuscript received July 16, 2004; revised February 9, 2005. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grants RGPIN227612 and RGPIN1345, and by the Canada Foundation for Innovation under the New Opportunities Program. This paper was recommended by Associate Editor J. Wang.

The authors are with the Department of Electrical and Computer Engineering, University of Western Ontario, ON, Canada (e-mail: xyang23@engga.uwo.ca; mmoallem@engga.uwo.ca; rajni@eng.uwo.ca).

Digital Object Identifier 10.1109/TSMCB.2005.850177

navigation algorithms are identified and some effective solutions are proposed in [17], [18]. A new concept for terrain-based navigation using fuzzy rules is presented in [19]–[21]. In the literature, there also exist methods combining fuzzy logic with other algorithms, such as genetic algorithms [22]–[24], potential fields [25]–[27], and neural networks [28]–[30].

Considering the pros and cons of the above-mentioned approaches to mobile robot navigation, in this paper, we propose a layered goal-oriented motion-planning strategy based on fuzzy logic. The global goal position and long-range sensory data are used by the first layer of the planner to produce an intermediate goal point, called the way-point, with a favorable direction for reaching the goal. The second layer of the planner takes this way-point as a subgoal and uses short-range sensory data to guide the robot to reach the subgoal while avoiding collisions. The resulting path is similar to the path produced by the visibility graph method in that the way-point is located around vertices of the configuration region of the obstacles. But here no assumptions and prior knowledge about the environment are needed. The proposed navigation strategy is a hybrid approach consisting of model-based and sensor-based approaches.

This paper is organized as follows. In Section II, we briefly review the basic idea of fuzzy navigation algorithms for mobile robots. Section III depicts the development of the layered goal-oriented fuzzy motion planning strategy. In Section IV, the implementation issues are discussed and the experimental results with the Koala robot are presented. In Section V, we summarize the benefits of the proposed navigation algorithm and present possible future directions.

II. Fuzzy Logic in Mobile Robot Navigation

A fuzzy set is characterized by a mathematical formulation known as the *membership function*. Over a given universe of discourse X, the membership function of a fuzzy set $\tilde{F}$, denoted by $\mu_{\tilde{F}}(x)$, maps elements $x \in X$ into a numerical value in the closed unit interval, i.e.,

$$\mu_{\tilde{F}}(x) :\longrightarrow [0, 1].$$

The basic idea of fuzzy control in mobile robot navigation may be classified into the categories described below according to the form of the fuzzy rule.

The *direction-based* fuzzy rule takes the following form.

- IF disallowed-direction is A and desired-direction is B, THEN steering-direction is C

where A, B, and C are all represented by fuzzy sets, and $C = (1 - A) * B$ (the notation $*$ is a *t-norm* operation in fuzzy set theory). This form of fuzzy rule combines information about obstacles and goal position together and gives the final steering direction which is safe, in the sense of avoiding collisions, and desired, in the sense of seeking the goal. The fuzzy rule bases designed in [6], [11], [15], and [18] typically consist of direction-based rules.

The *speed-based* fuzzy rule takes into account obstacle repulsion and goal attraction to set the speeds for the motors [7], [27]. The general form of the rule is:

- IF obstacle-condition, THEN change-of-speed is ΔV_{o1} for left-motor, and ΔV_{o2} for right-motor;
- IF goal-condition, THEN change-of-speed is ΔV_{g1} for left-motor, and ΔV_{g2} for right-motor;

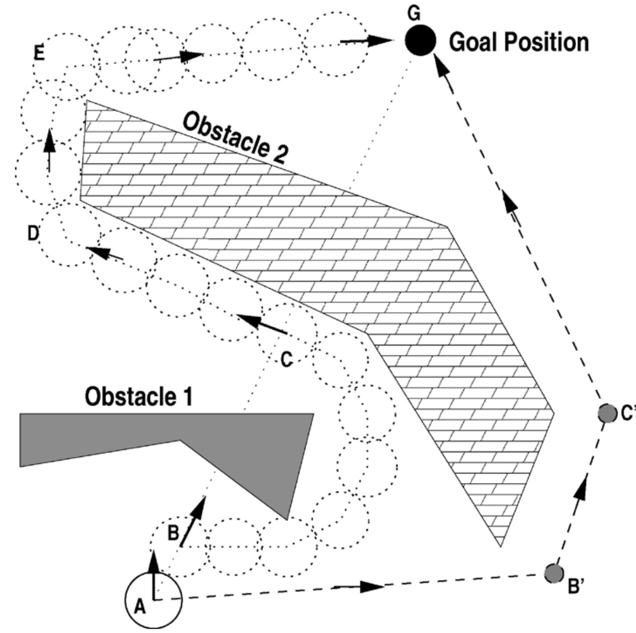


Fig. 1. Shortsighted behavior versus goal-oriented planning behavior.

where the *obstacle-condition* is decided by sensory data and the *goal-condition* by the relative position between the robot and the goal position. ΔV_{o1}, ΔV_{o2}, ΔV_{g1}, and ΔV_{g2} are represented by fuzzy sets. The final speed of each motor is an aggregation of the outputs of the fired rules. The resulting behavior of the robot is to pursue the goal position while avoiding obstacles.

The *traversability-based* fuzzy rule is a relatively new concept in autonomous navigation using a "traversability index" and is mainly used for terrain assessment of planetary rover navigation [19]–[21]. The traversability index can be obtained based on sensory data, which can be represented by fuzzy sets with linguistic labels, such as {*POOR, LOW, MODERATE, HIGH*}, corresponding to surfaces that are *unsafe*, *moderately-unsafe*, *moderately-safe*, or *safe* for traversal, respectively. Typical examples of the corresponding fuzzy rules are as follows.

- IF τ^* is *LOW*, THEN v is *SLOW*.
- IF τ^* is *MODERATE*, THEN v is *MODERATE*.
- IF τ^* is *HIGH*, THEN v is *HIGH*.

Here, τ^* is the traversability index and v is the speed of the robot denoted by fuzzy sets *SLOW*, *MODERATE*, and *HIGH*. The navigation strategy deals with uncertain knowledge about the environment and uses onboard terrain analysis to endow the planetary rover with the ability to autonomously select easy-to-traverse paths toward the goal.

III. Layered Goal-Oriented Fuzzy Navigation Strategy

The fuzzy algorithms mentioned in Section II mainly fall in the reactive approach to autonomous navigation. There is no explicit fuzzy rule for motion planning in terms of seeking the global goal used in these strategies. In other words, information about obstacles and goal position is used simultaneously as local information, which may result in a *shortsighted behavior* in some situations. For example, as shown in Fig. 1, using purely reactive fuzzy navigation algorithms, the robot turns toward the goal at position *A* since both obstacles are not considered close

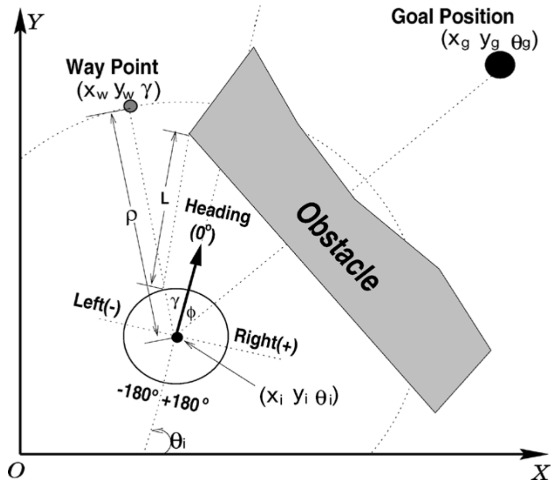


Fig. 2. Direction and terminology definitions.

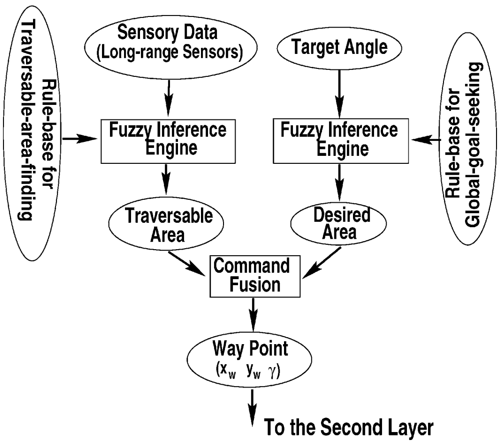


Fig. 3. Configuration of the first layer of the planner.

enough to fire the obstacle avoidance rules at this moment. The resulting path is likely *ABCDEG*, which is less optimal than path *AB'C'G*. To resolve this problem, we develop a layered goal-oriented fuzzy navigation algorithm taking into account information about the goal as global information in the first layer. Intermediate goals, called way-points, are then produced on-line for the robot to pursue during the navigation. In the second layer, the way-point is taken as the subgoal for the robot to reach, while avoiding obstacles locally. Using this strategy, the intermediate goal positions *B'* and *C'* are produced onboard respectively. As a result, the path *AB'C'G* would be followed as desired.

In this paper, we study the navigation problem in indoor two-dimensional (2-D) environments. However, the layered goal-oriented fuzzy navigation strategy is also applicable to outdoor navigation problems once the robot is equipped with appropriate sensors.

A. Terminology

Since a computer is usually used to control a mobile robot, it is efficient and helpful to select a discrete universe of discourse X. In our application, X is defined as

$$X : [-180^\circ, 180^\circ].$$

The definition of the direction used in our navigation system is as illustrated in Fig. 2. The robot heading direction is always taken as the 0° direction, with the negative direction to its left and the positive direction to its right. Referring to Fig. 2, the terms used in this paper are defined as follows.

- The *perceptive region* refers to a region around the robot which can be swept by the long-range sensors, as illustrated by the area within the dotted arc in Fig. 2.
- A sensor is called a *fired sensor* if it has a measurement of $d > \sigma$, where σ is a threshold related to the noise level of the sensors.
- The *target angle* is the angular difference between the robot heading direction and the ray from the robot center to the goal position, denoted by ϕ.
- The *way-point* is a via point on the path leading to the goal position, represented by (x_w, y_w, γ). The term γ depicts the orientation of the way-point with respect to the robot heading direction, which is one of the outputs of the first layer of the planner.
- The *navigating angle*, expressed by θ, is the orientation of the robot, i.e., the angle between the robot heading direction and the X-axis of the general coordinate system XOY attached to the environment.
- The *configuration* is defined by the position and orientation of the robot in the work space XOY. The configuration at time t_i is denoted by (x_i, y_i, θ_i), the initial configuration by (x_0, y_0, θ_0), the subgoal by (x_w, y_w, θ_w) and the global goal by (x_g, y_g, θ_g).

Generally, the initial and goal configurations are task-specific. The intermediate configuration (x_i, y_i, θ_i) is obtained by proprioceptive sensors on the robot, such as wheel encoders, or by GPS equipment. The signs of ϕ, γ and θ follow the definition of direction, negative to the left of the robot, and positive to the right. From Fig. 2, it follows that

$$x_w = x_i + \rho\cos(\theta_i - \gamma) \tag{1}$$

$$y_w = y_i + \rho\sin(\theta_i - \gamma) \tag{2}$$

$$\theta_w = \theta_i - \gamma \tag{3}$$

where ρ is the distance that the robot needs to travel to reach the way-point. Also

$$\rho = L + \delta \quad \text{and} \quad \rho \leq R \tag{4}$$

where L is the distance between the robot and the obstacle near the way-point detected by the long-range sensor in the corresponding direction, δ is an offset which depends on the size of the robot, and R is the radius of the perceptive region.

B. The First Layer of the Planner

The first layer of the planner consists of three modules, namely, *traversable-area-finding*, *global-goal-seeking*, and *command-fusion*. The architecture of the first layer is schematically shown in Fig. 3. The robot is required to be equipped with a ring of long-range sensors, such as high-resolution laser range-finders or ultrasonic sonar transducers. The data from the long-range sensors are used in the traversable-area-finding module to generate the *traversable area* in the perceptive region based on the fuzzy rules in its rule base. The global-goal-seeking module takes the target angle ϕ as input and outputs the *desired*

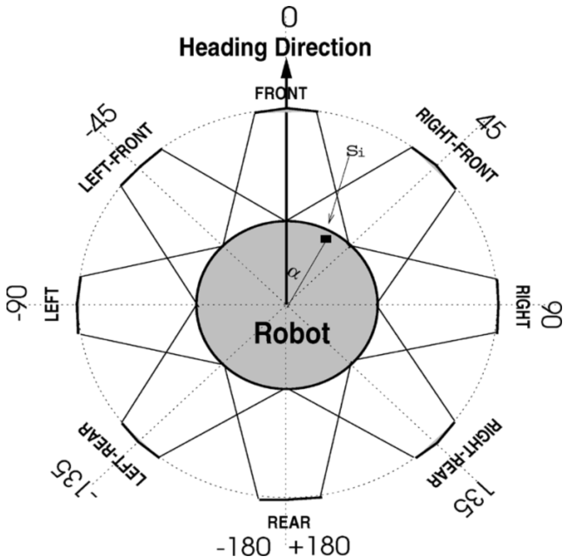


Fig. 4. Rule base for the traversable-area finding module.

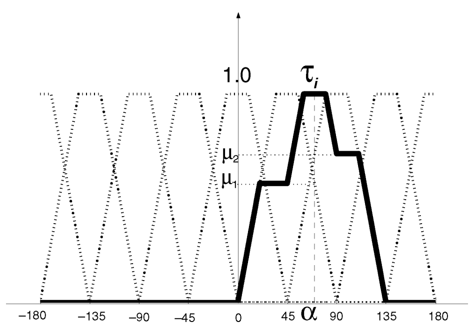


Fig. 5. Untraversable area identified by a fired sensor.

area. Both *traversable area* and *desired area* are represented by fuzzy sets. The command fusion module performs fuzzy conjunction upon incoming fuzzy sets and outputs the direction of way-point γ. The position of the way-point (x_w, y_w) is then calculated using (1) and (2). Combining (3), the subgoal (x_w, y_w, θ_w) is set for the robot to pursue in order to get to the final destination. It is obtained by combining perceptible knowledge about the environment and information about the global goal position.

Because piecewise linear functions are evaluated faster and more efficiently by computers used in embedded applications, the membership functions used in our navigation system take on triangular and trapezoidal shapes. The rule base in the traversable-area-finding module contains eight fuzzy sets with the linguistic labels {FRONT, RIGHT-FRONT, RIGHT, RIGHT-REAR, LEFT-FRONT, LEFT, LEFT-REAR, REAR}, depicted in Fig. 4. The general form of the fuzzy rule is as follows.

- IF s_i is fired, THEN the untraversable area is τ_i.

As shown in Fig. 4, s_i denotes the i^{th} sensor on the robot with distributive angle of α, and τ_i is a composed fuzzy set representing the untraversable area within the perceptive region corresponding to this sensor. The process of fuzzy inference is illustrated in Fig. 5. The dotted trapezoids are the equivalent depictions of the eight fuzzy sets shown in Fig. 4 in the common fuzzy-set representation system. With an input α, the rules for

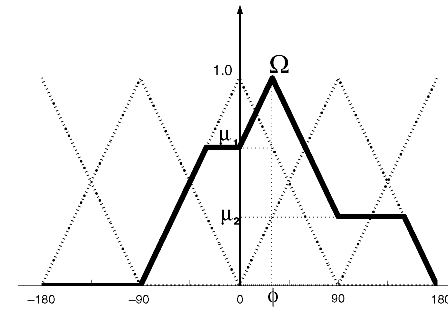


Fig. 6. Rule base for the global-goal-seeking module.

RIGHT-FRONT and *RIGHT* are fired with strengths μ_1 and μ_2, respectively. The term τ_i is obtained from

$$\tau_i = \mu_1 \oplus \mu_2 = \min\{1, \mu_1 + \mu_2\}. \tag{5}$$

Here, the *t-conorm* operation *bounded sum* $\oplus$ is used to emphasize the overlapping of the fuzzy sets [31]. This is reasonable since the area in the direction of the fired sensor should be most untraversable. Once every fired sensor on the robot identifies its corresponding untraversable area, the overall traversable area, denoted by Γ, is aggregated by

$$\Gamma = not \bigvee_{i=1}^{n} \{\tau_i\} = 1 - \max_{i=1}^{n}\{\tau_i\} \tag{6}$$

where n is the number of the fired sensors. The t-conorm operator $\max$ is used since the inference is consistent with the intuition that the degree of the traversability should be determined by the fired sensor which has the strongest opinion about it.

The rule base for the global-goal-seeking module consists of five fuzzy sets uniformly distributed in the universe of discourse X, as illustrated by the dotted triangular membership functions in Fig. 6. Note that this design of the fuzzy rule base gives the robot the broadest span of flexibility without losing the orthogonality property.[1] The flexibility is necessary for the robot to seek the goal position in a clustered environment. The orthogonality captures the fact that every direction should have an equal chance to be the desired direction, and the target angle is always the most desired direction to go for the sake of achieving the goal. With the target angle ϕ, the fuzzy inference engine gives the desired area Ω, which is an aggregation of the fired fuzzy sets. As shown in Fig. 6, Ω is given by

$$\Omega = \mu_1 \vee \mu_2 = sum\{\mu_1, \mu_2\} \tag{7}$$

where μ_1 and μ_2 are the firing strengths of the two adjacent fuzzy rules, respectively. By using the *t-norm* operator $\min$ over Γ and Ω, the direction of the way-point is determined by

$$\tilde{\gamma} = \Gamma \wedge \Omega = \min\{\Gamma, \Omega\}. \tag{8}$$

The set $\tilde{\gamma}$ is usually a fuzzy set with multiple peaks representing the area that is both traversable and desired. The crisp value of

[1]The sum of the membership values for any point in the universe of discourse is 1.0. The orthogonality is a desirable property for analyzing the stability and completeness of fuzzy systems [32].

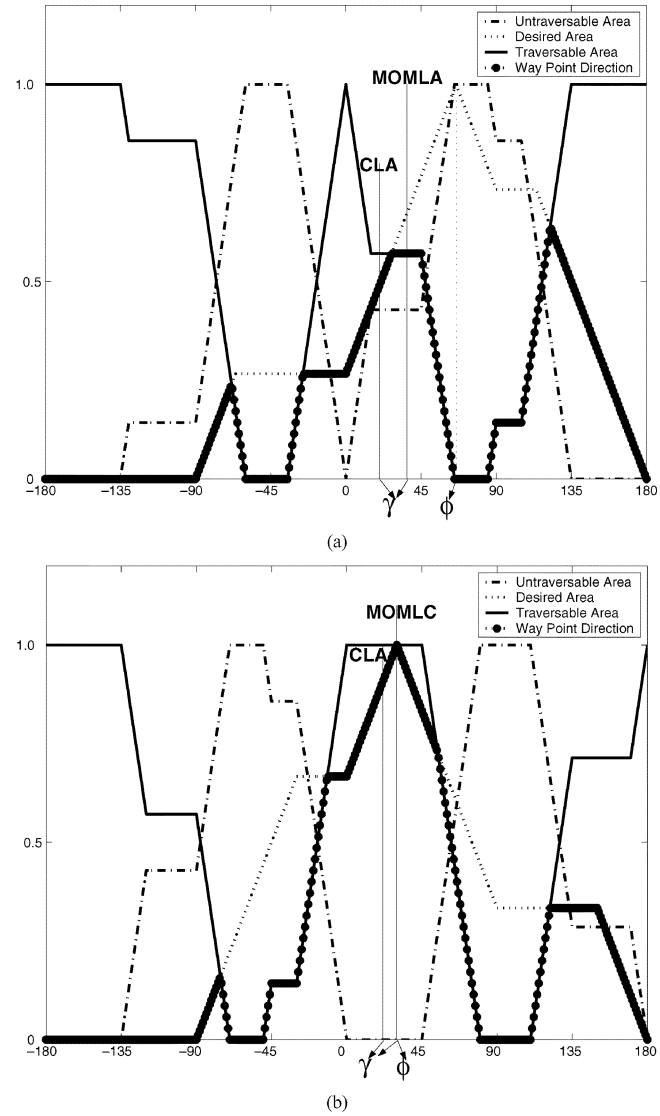


Fig. 7. Examples of the fuzzy inference mechanism in the first layer of the planner.

γ is obtained using the *Mean of Maximum of Largest Area* (MOMLA) defuzzification strategy upon $\tilde{\gamma}$. The position of the way-point (x_w, y_w can be obtained using (1) and (2). Together with the orientation θ_w obtained by (3), the way-point is taken as a subgoal for the second layer of the planner. Here we propose to use the *mean of maximum of largest area* (MOMLA) strategy which is a combination of the *mean of maximum* (MOM) [31] and *centroid of largest area* (CLA) [33] defuzzification methods. Fig. 7 shows two examples of the fuzzy inference mechanism in the first layer of the planner. Compared with the CLA defuzzification method, the output of the MOMLA strategy is closer to the target angle and hence is more desired as illustrated in Fig. 7(a). This is especially significant when the target angle is contained in the traversable area. As shown in Fig. 7(b), the final output is exactly the target angle, which means that the way-point should be in the goal direction despite the obstacles being in other directions. It should be noted that the CLA method results in a less desirable value of γ for reaching the goal position. It is worth noticing that, since the values of the *cross points* [31] in fuzzy rule bases are designed to be equal or bigger than 0.5 and the operator $\oplus$ is used in the fuzzy inference engine, the peaks of the fuzzy set $\tilde{\gamma}$ are uniquely separated. This saves computation time when using the CLA (or MOMLA) defuzzification strategy.[2]

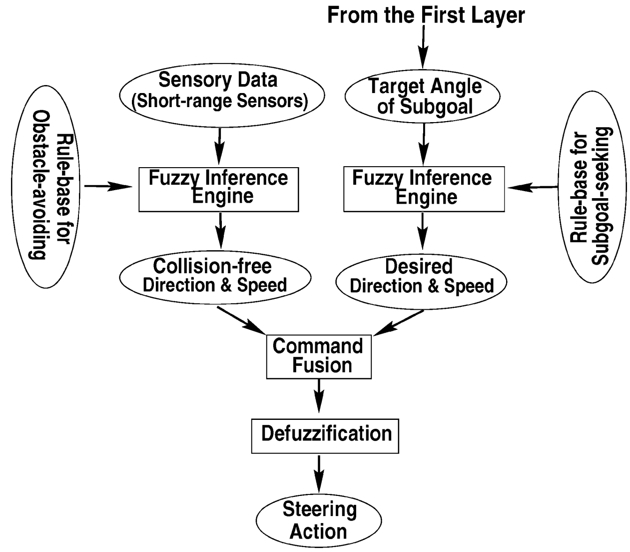


Fig. 8. Configuration of the second layer of the planner.

C. The Second Layer of the Planner

The objective of the second layer of the planner is to guide the robot to reach the subgoal, which has the configuration (x_w, y_w, θ_w), while avoiding collisions with obstacles. Since at this stage the robot needs to react to a relatively tight space, a ring of short-range sensors such as infrared proximity light sensors, or touch sensors, are used as a virtual bumpers to keep the robot free from collisions. The pure reactive fuzzy navigation algorithms such as *direction-based* [13], [18] and *speed-based* [7] controllers have been demonstrated to be efficient for this stage. In the following, we briefly summarize the configuration of the second layer of the planner. The interested readers may refer to [6], [7], [13], [15], [18], [27] for details.

As shown in Fig. 8, the architecture of the second layer of the fuzzy planner is generally composed of three modules: *obstacle-avoidance*, *subgoal-seeking*, and *command fusion*. The obstacle-avoidance module takes the short-range sensory data as input and outputs the direction and speed that is safe for the robot to take. The target angle of the subgoal is used to produce the desired direction and speed. The final steering action is a combination of outputs from the collision-free direction and desired-direction modules. Here, it is worth mentioning that we do not need to design the unique membership functions of the fuzzy set *NEAR* for individual sensors.[3] To reduce the redundant turning in the avoidance behavior, the robot is allowed to almost 'touch' an obstacle before avoiding it. It is sufficient to set a single threshold for sensory measurements,

[2]The basic concept of the strategy is to divide the nonconvex fuzzy control command into distinct control areas, and the centroid of the largest area is taken as the final control command. Dividing a nonconvex fuzzy set into several distinct areas is usually inconvenient and time consuming [33].

[3]The membership functions *NEAR* for different sensors are generally designed uniquely based on the consideration that an obstacle in front of the robot is considered more dangerous than one on the side [7], [13], [18].

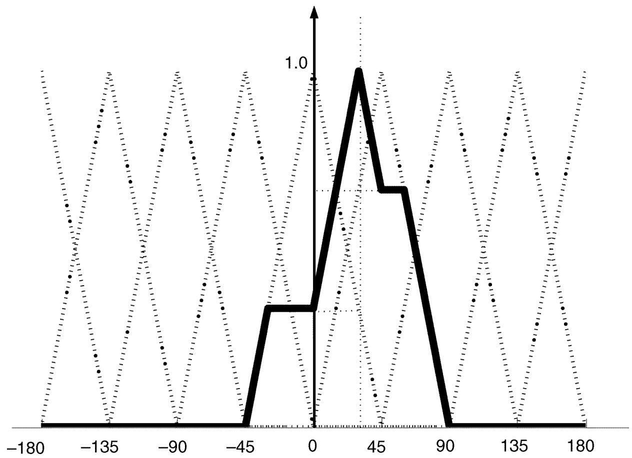


Fig. 9. Rule base for the subgoal-seeking module.

and apply the fuzzy rules in the following form for the avoidance behavior.

- IF sensor s is *fired*, THEN the direction taken care of by s becomes *fully disallowed*.

When a sensor gets *fired*, it means its measurement surpasses a threshold (the value is usually very high); and when a direction becomes *fully disallowed*, it means the fuzzy rule regarding this direction is fired with a full strength of 1.0. Additionally, redundant turnings may be further reduced by using a fine tuned fuzzy rule-base for the subgoal-seeking module as shown in the dotted membership functions in Fig. 9. The solid membership function in Fig. 9 is obtained using (7), which gives a narrower span of the desired direction compared with the output of the global-goal-seeking module shown in Fig. 6. This is reasonable in the sense that the robot should pursue the subgoal as much as it can at this stage, because the subgoal is the outcome of the first planning stage and is thus optimal to some degree for fulfilling the navigation task. But, unlike the global goal, the subgoal does not need to be attained definitely for the robot. In some cases, the subgoal may be unattainable. For instance, consider situations where the robot gets stuck itself, or the way-point is occupied because of the changing environment. In these situations, the robot is forced to give up the subgoal and wait for a new one from the first layer. This can be achieved by setting a deadline for the subgoal seeking behavior. The deadline is dynamically determined by

$$T = \frac{\rho}{v_b} \tag{9}$$

where v_b is the *base speed* representing the translational velocity of the robot when there is no front- or front-side-obstacles detected. Any obstacle which appears *near* on the heading way slows down the speed of the robot. So v_b is actually the maximum speed value in a navigation process which is set at initialization. A new way-point is generated whenever the deadline T expires. We can see that T is also the period of the first layer of the planner. If the way point cannot be reached at the expiration of the deadline, the robot takes the orientation of the subgoal and resumes the planning scheme in the first layer for a new way-point.

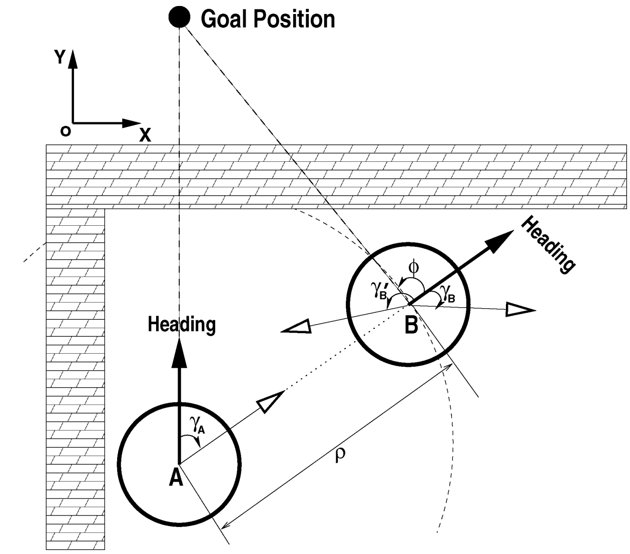


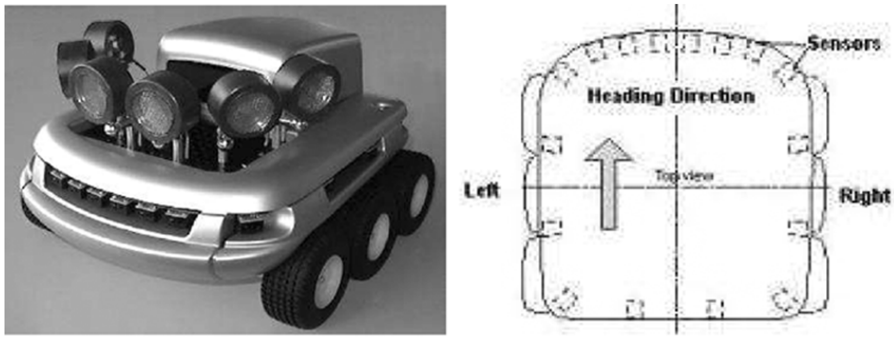
Fig. 10. Anti-deadlock mechanism.

In other cases, for example, when a more favorable path leading to the global goal is found on the way to the subgoal and the global goal is in the perceptive region, the robot should give up the current subgoal and resume the planning mechanism in the first layer. This can be done by timely checking the long-range sensory data and the relative position of the global goal.

D. Further Considerations

The proposed layered fuzzy motion planning strategy aims to mimic the human-like behavior of seeking a given goal position in an unknown environment. Without proper outside aids, it is quite possible for a human being to get lost in a strange and complex environment. Similarly, the robot with the current strategy may get lost in some situations without any external information such as images from a camera, self localization from GPS, and proper intervention or guidance from a human operator. To accept external help usually means to give up a part of automation for the robot, resulting in human-in-the-loop, semi-autonomous navigation algorithms for mobile robots. For an autonomous mobile robot, the possibility and ability of reaching the goal largely depend on the sensory system. Based on only range-finder sensors, we have implemented an *anti-deadlock mechanism* to improve the ability of reaching the goal for the robot autonomously navigating in an unknown environment.

Consider a common indoor situation as shown in Fig. 10, in which the robot needs to detour around a long wall to reach the goal. At position **A**, the robot is heading to the goal position and hence $\phi = 0°$. Due to the limitation of the sensory system, the end of the wall is out of the perceptive region at this moment. Within the perceptive region illustrated by the dotted arc in Fig. 10, the crisp output $\gamma_A = 65°$ is obtained by the fuzzy inference mechanism in the first layer. Using (1)–(4), with $\rho = R$, the next way-point is calculated to be at position **B**. For this illustrative example, the target angle ϕ at position **B** is $-95°$. With this target angle, the fuzzy inference mechanism in the first layer of the planner produces the crisp output



(a) (b)

Fig. 11. Koala robot (silver version).

$\gamma'_B = -156°$. If the robot takes this value as the direction of the next way-point, the robot would be wandering in front of the wall without making a detour. It is said the robot is *deadlocked*. The deadlock problem typically happens when the directions to the left, or right, are blocked and the target angle is large enough in the negative direction, or positive direction, respectively. In such a situation the ability of following a wall is needed for the robot. In nonfuzzy navigation schemes, wall-following algorithms [34], [35] enable the robot to follow object contours. The control goal in wall-following algorithms is to keep a constant distance between the robot and the object contour while the robot is moving with a constant speed. In most real-world cases, however, the sensory measurement is usually corrupted by noise, making it difficult to obtain the precise value of the distance. Furthermore, considering a human being's behavior of following a wall, keeping a constant distance between the robot and the object is unnecessary. For the implementation of the fuzzy navigation strategy, we have used a simple and efficient anti-deadlock mechanism that mimics human's behavior in following a wall. Based on the distributive angles of the fired sensors, a temporary target angle is used to produce the next way-point in a possible deadlock situation. At position **B** shown in Fig. 10, for instance, the robot checks its long-range sensors and finds all the fired sensors. The distributive angles of the fired sensors compose a vector as follows:

$$\{-135°, -105°, -75°, -45°, -30°, -15°, -5°, 5°, 15°\}$$

which corresponds to the sensors {$L6$, $L5$, $L4$, $L3$, $L2$, $L1$, $L0$, $R0$, $R1$} [16 sensors are placed around the Koala and are positioned and numbered as shown in Fig. 11(b)]. Then the distributive angle of $R0$, i.e., $5°$, is selected to be the temporary target angle. Using this temporary target angle, the output of the first layer of the planner is $\gamma_B = 45°$. It can be seen that γ_B is a reasonable direction for the next way point. With the anti-deadlock mechanism, the exact distance between the robot and a contour is no longer a main concern. The robot can make a detour when meeting with large or U-shaped obstacles. The measured distance, which need not to be accurate, is only used for setting the deadline in the second layer of the planner. Therefore, the sensory noise tolerance of the system is greatly improved.

IV. Implementation and Experimental Results

We have implemented the layered goal-oriented fuzzy motion planning algorithm on an actual mobile robot—the Koala, and tested it through extensive experiments in various environments.

A. Implementation Issues

The Koala is a small (32 cm $\times$ 32 cm), six wheeled, differential drive vehicle manufactured by K-team SA. Low-level motion and hardware control are performed by an on-board microprocessor (16-MHz Motorola 68 331). The Koala is currently equipped with a ring of 16 infrared proximity/ambient light sensors and incremental wheel encoders. Fig. 11(a) shows the image of the Koala with sonar extension of six ultrasonic sonar transducers that can detect obstacles over a wide range from 15 cm to 3 m. The IR sensors provide a range of measurements from 5 to 20 cm, and the physical distribution of the 16 IR sensors is depicted in Fig. 11(b) [36].

The fuzzy rule base of the traversable-area-finding module designed for the first layer of the planner is uniform and general, which can be applied to any robot equipped with a ring of long-range sensors. As for the second layer of the planner, the rule base for the obstacle-avoiding module was implemented on the Koala. The measurement distribution of the IR proximity sensors used on the Koala is illustrated in Fig. 12 [36]. We can see that these sensors have a field of view of about $\pm 10°$ degrees. Considering the measuring characteristic of the IR proximity sensor, the rule base for obstacle-avoiding behavior for the Koala is designed as shown in Fig. 14. Since the distribution of sensors on the Koala is left-right symmetrical, here we only show the rules regarding sensors on the right part for brevity. Every sensor is associated with a unique fuzzy set. The design of the rule base reflects the fact that the Koala has denser sensors on the head, as represented by fuzzy sets in solid lines in Fig. 14. The fuzzy sets associated with other sensors are swelled to cover the areas with lack of sensors as shown by the dotted-line fuzzy sets in Fig. 14. All of the fired fuzzy rules are aggregated using the *t-conorm* operator $\oplus$ and (5) and (6) are applicable. Thus, the overlap between the fuzzy sets is meaningful in the sense that, if two adjacent sensors are fired, then the direction between them becomes blocked in the eyes of the robot. In such a case, the way in this direction is too narrow to get through even if there is no obstacle in this direction.

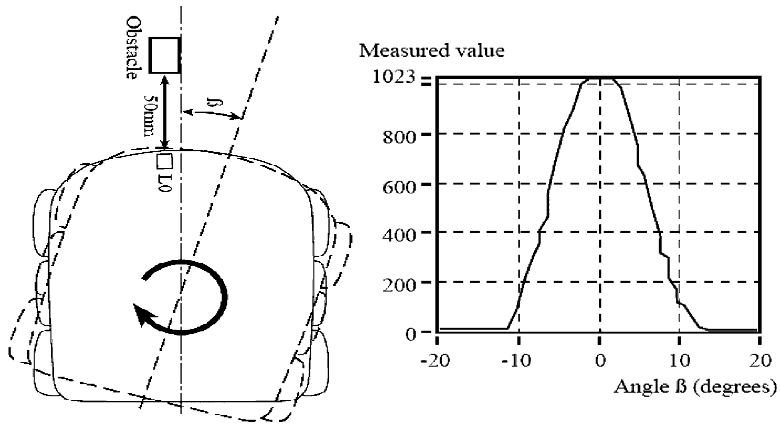


Fig. 12. Typical response of a proximity sensor in front of an obstacle (20 mm in width) viewed under a variable angle β (the measurement is in digital form and the maximum is 1023.).

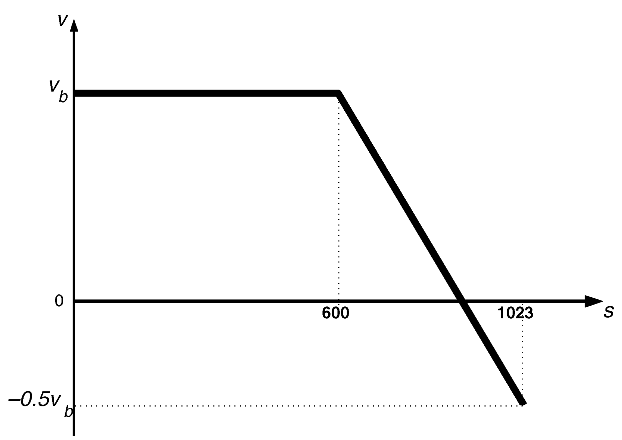


Fig. 13. Speed pattern for the Koala used in the experiments (s is the maximum reading from the front sensors, v_b is the base speed).

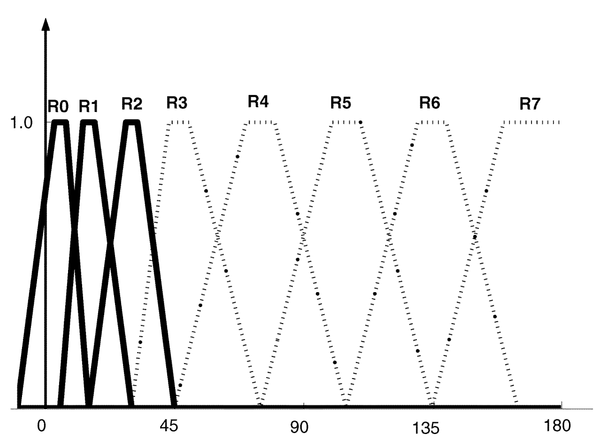


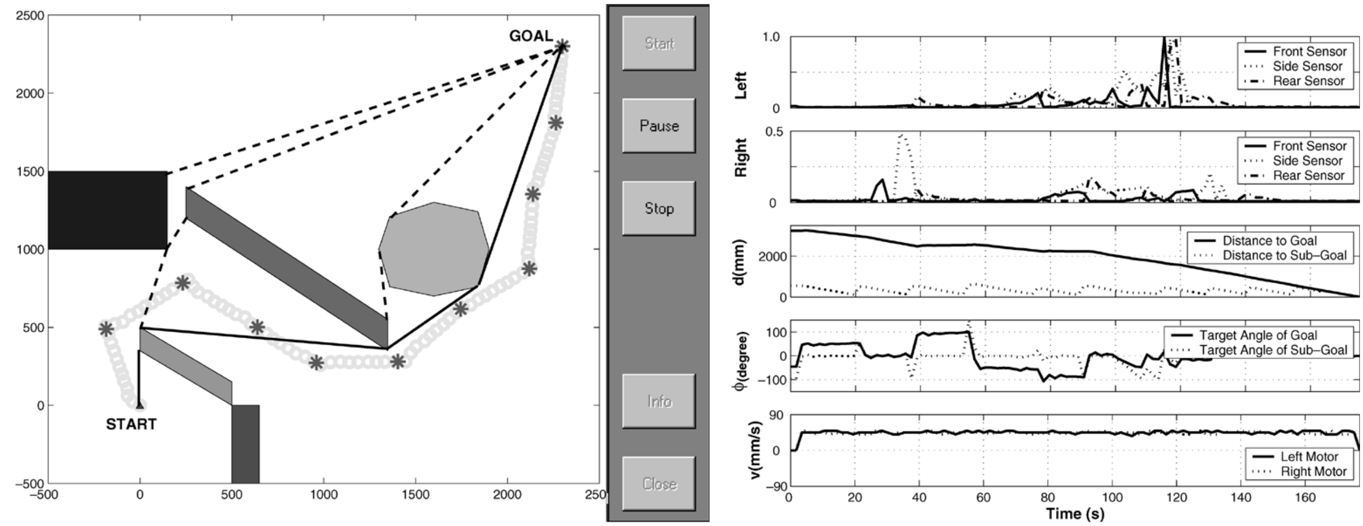
Fig. 14. Rule base of the obstacle-avoiding module for the Koala.

Since our Koala is only equipped with IR proximity sensors, in the experiments, we use these sensors as both long-range and short-range sensors with different values for the thresholds. The speed of the Koala used for the experiments is as shown in Fig. 13. We can see that the robot can move backward if required.

B. Experimental Results

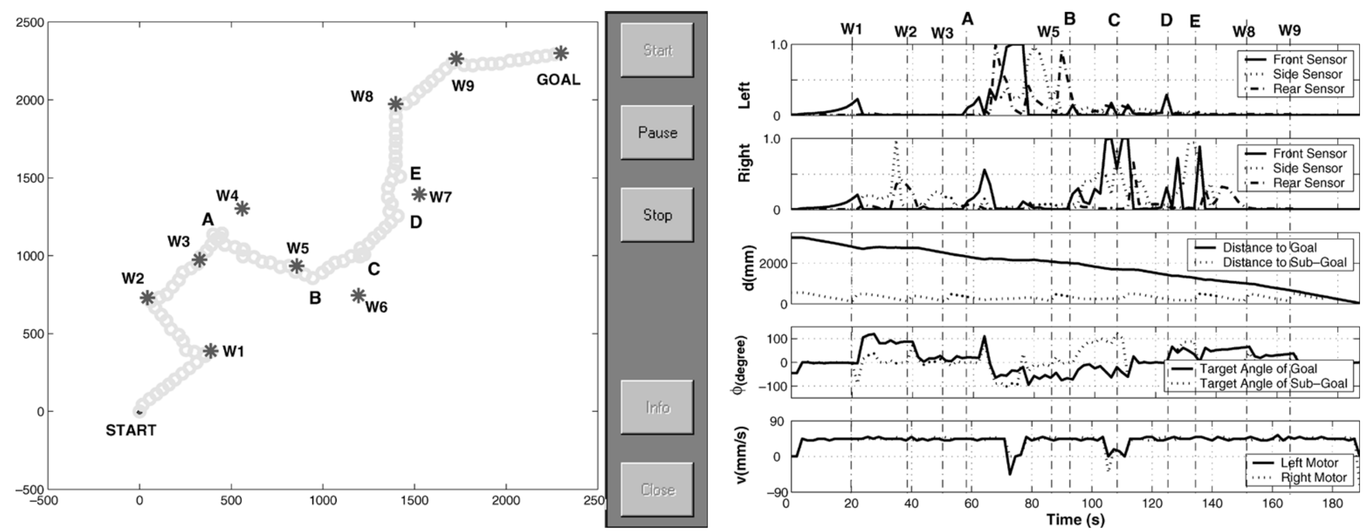
The performance of the algorithm is first tested in static environments. By a static environment, we mean that there are no moving obstacles in the environment during the navigation period. Fig. 15 shows one of the experimental results. The resulting path is depicted by the line made up of circles, and the way-points are illustrated by asterisks. The lines connecting the initial point and the goal position are possible outputs from the visibility graph method. We can see that, in this experiment, the robot can reach every way-point. Since the way-points are either along the edges, or around vertices of obstacles, the resulting path is quite similar to one of the paths generated by the visibility graph method as indicated by the solid line in Fig. 15. Fig. 15(b) illustrates the response of the robot to its environment. It can been seen that the sensory measurements are generally low, except when a way-point has nearby obstacles. This reflects the advantage of the proposed navigation algorithm in the sense that the robot can avoid obstacles before getting close to them. Thus, the "shortsighted" problem mentioned in Section III is overcome.

In order to test the algorithm in dynamic environments, we performed several experiments by changing the configuration of the obstacles during the navigation. One of the results is shown in Fig. 16. We can see that, in Fig. 16(a), not all the way-points can be reached. Referring to Fig. 16(b), we trace the behavior of the robot in the dynamic environment as follows. At the starting point, the sensory data is low and the first layer of the planner gives way-point $W1$, which is in the goal direction. When the robot reaches point $W1$, the first layer generates $W2$ as the new way-point since some front sensors are fired. $W2$ and $W3$ are obtained respectively without disturbances. While the robot is approaching $W4$, an obstacle suddenly obstructs the direction toward $W4$. It can be seen that the robot gets stuck at point A for a while, then it begins to pursue the next subgoal $W5$ when the deadline is due for the second layer of the planner. At point B, the robot is disturbed and is driven away from $W6$. At point C, the robot is wandering in an effort to reach $W6$. When the deadline is due, it gives up the subgoal $W6$ and turns toward the new subgoal $W7$. Similar behaviors are performed around $W7$, $W8$ and $W9$, and finally the global goal is attained.



(a) (b)

Fig. 15. Koala robot navigates an unknown static environment.



(a) (b)

Fig. 16. Koala robot navigates in an dynamic environment.

The effectiveness of the proposed navigation strategy is demonstrated by comparing it with pure reactive fuzzy algorithms. The resulting paths and response data are presented in Figs. 17–20. As we can see, with either the *direction-based* [18] or *speed-based* [7] reactive fuzzy navigation algorithms, the avoidance behaviors occur only when the robot gets close to an obstacle. This may not only result in "shortsighted" problems, but can also lead to failures in navigation. Fig. 17 shows an example when the robot meets a large corner-like obstacle. In such a situation, the robot easily gets stuck with the pure reactive fuzzy navigation algorithm. While with the layered goal-oriented navigation strategy, the robot can make an early detour, without stepping close to the obstacle.

To investigate the efficiency of the proposed algorithm, the Koala was programmed to navigate the same environment with different fuzzy algorithms. Compared to the direction-based and speed-based algorithms, the navigation time was reduced by 27.6% and 16.3%, respectively. This improvement is mainly

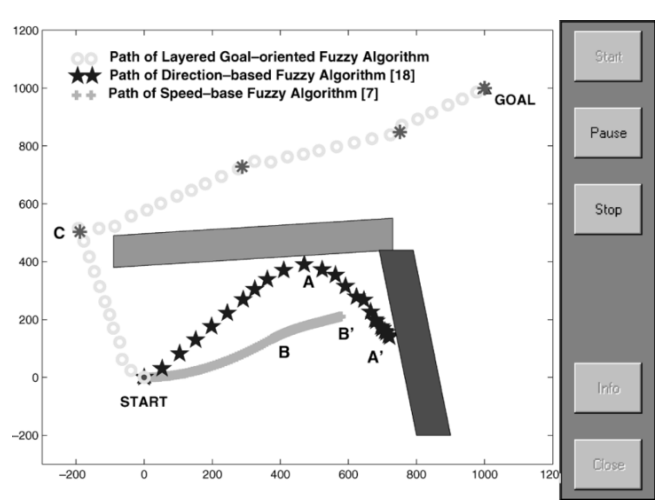


Fig. 17. Comparisons of the proposed strategy with reactive fuzzy navigation algorithms [7], [18].

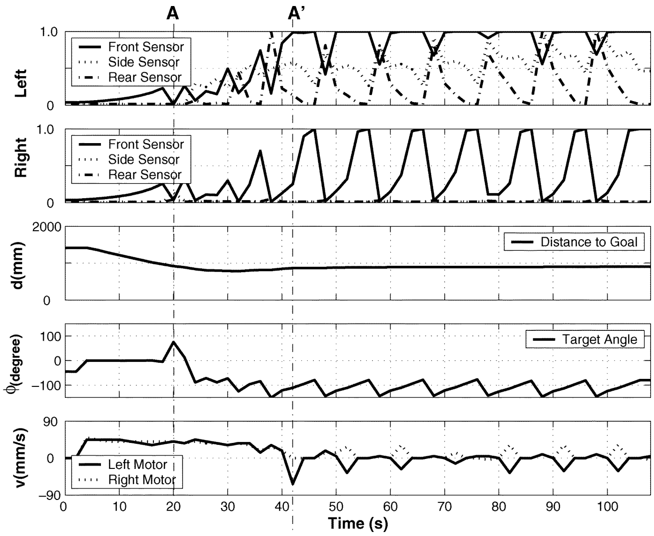


Fig. 18. Direction-based reactive fuzzy algorithm (path A in Fig. 17).

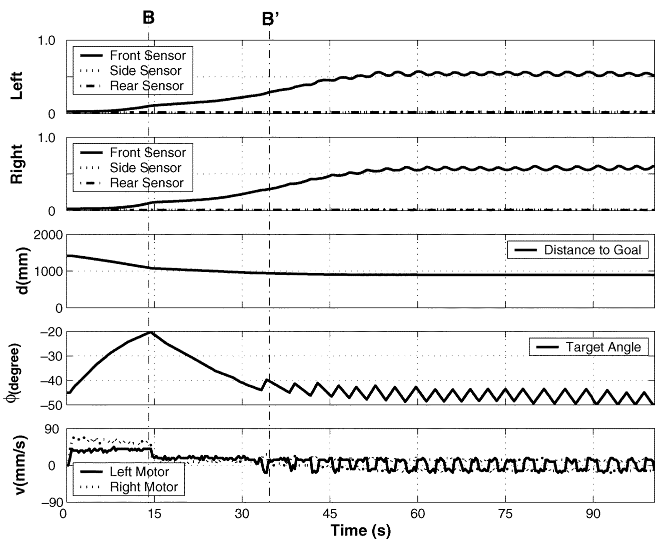


Fig. 19. Speed-based reactive fuzzy algorithm (path B in Fig. 17).

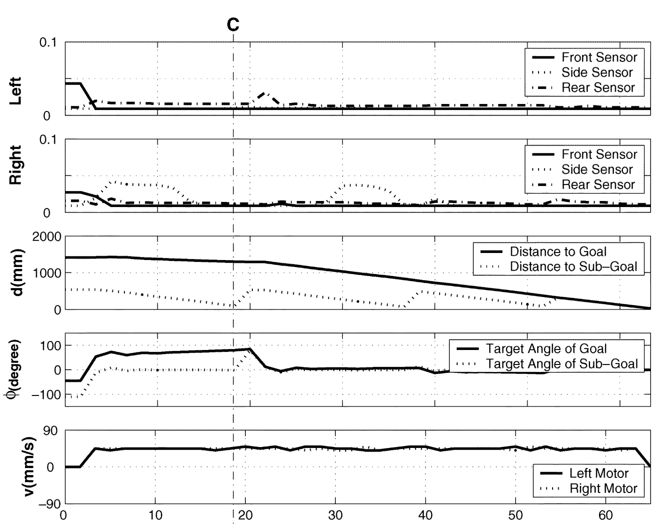


Fig. 20. Layered goal-oriented fuzzy algorithm (path C in Fig. 17).

due to the subgoal mechanism in the layered navigation system. With the layered goal-oriented algorithm, the subgoal, which is usually in a collision-free space, is generated to guide the robot to take a relatively favorable path and thus the robot does not need to frequently negotiate with close obstacles. With purely reactive fuzzy algorithms, however, the robot takes a lot of time negotiating with close obstacles. In these cases, the speed of the robot has to be reduced from time to time because of the possibility of collision with close obstacles.

V. Conclusion

In this paper, the development of a layered goal-oriented fuzzy motion planning strategy was discussed. Experimental results indicate that the algorithm is efficient and effective in terms of goal seeking and obstacle avoidance behavior and real-time performance. The new algorithm endows the robot with a human-like ability of reasoning about the environment, thus improving the navigation performance.

We would like to emphasize some merits of the new algorithm as follows. Firstly, due to its layered modular design and the interpolative reasoning mechanism, the new motion planning strategy is very simple and concise. This brings several advantages, such as small sizes of the rule-bases, reusability of modules, and easy extensibility. Secondly, the real-time property of the system is improved because of the reduction of computation time. Thirdly, the design task is greatly simplified compared to the traditional methods [13], [18]. We do not need to consider the degree of *nearness* between the robot and obstacles due to the layered construction of the navigation strategy. The design pattern is thus of uniformity and generality. Finally, the layered planning scheme also endows the robot with the capability of escaping from stuck conditions because of the deadline mechanism incorporated in the subgoal seeking behavior. With the anti-lock mechanism involved, the robot can behaves like a human being when detouring a large obstacle without caring about the exact distance in between.

The current implementation of the methodology described in this paper is based on an elementary sensory system and was tested in indoor environments. Equipping the robot with more sophisticated sensory systems, such as sonar sensors, cameras, and GPS, would improve performance of the navigation system in terms of path optimization.

References

[1] J. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.
[2] N. Nilsson, "A mobile automation: An application of artificial intelligence techniques," in *Proc. 1st Int. Joint Conf. Artificial Intelligence*, Washington, DC, 1969, pp. 509–520.
[3] J. Mitchell, "Planning Shortest Paths," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1986.
[4] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, Feb. 1986.
[5] W. L. Xu, S. K. Tso, and Y. H. Fung, "Fuzzy reactive control of a mobilt robot incorporating a real/virtual target switching strategy," *Robot. Auton. Syst.*, vol. 23, no. 3, pp. 171–186, 1998.
[6] H. Maaref and C. Barret, "Sensor-based navigation of a mobile robot in an indoor environment," *Robot. Auton. Syst.*, vol. 38, no. 1, pp. 1–18, 2002.
[7] X. Yang, R. V. Patel, and M. Moallem, "A fuzzy-Braitenberg navigation strategy for differential drive mobile robots," in *Proc. 3rd IFAC Symp. Mechatronic Systems*, Sydney, Australia, Sep. 2004.

[8] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Comput.*, vol. 1, pp. 180–197, 1997.
[9] M. Sugeno and M. Nishida, "Fuzzy control of a model car," *Fuzzy Sets Syst.*, vol. 16, pp. 103–113, 1985.
[10] T. Takeuchi, Y. Nagai, and N. Enomoto, "Fuzzy control of a mobile robot for obstacle avoidance," *Inform. Sci.*, vol. 43, pp. 231–248, 1988.
[11] J. Yen and N. Pfluger, "Path planning and execution using fuzzy logic," in *AIAA Guidance, Navigation and Control Conf.*, vol. 3, New Orleans, LA, Aug. 1991, pp. 1691–1698.
[12] P. Lee and L. Wang, "Collision avoidance by fuzzy logic control for automated guided vehicle navigation," *J. Robot. Syst.*, vol. 11, pp. 743–760, 1994.
[13] J. Yen and N. Pfluger, "A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 6, pp. 971–978, Jun. 1995.
[14] C. H. Lin and L. L. Wang, "Intelligent collision avoidance by fuzzy logic control," *Robot. Auton. Syst.*, vol. 20, no. 1, pp. 61–83, 1997.
[15] M. Montaner and A. Serrano, "Fuzzy knowledge-based controller design for autonomous robot navigation," *Expert Syst. Applicat.*, vol. 14, pp. 179–186, 1998.
[16] L. Doitsodis, K. Valavanis, and N. Tsourveloudis, "Fuzzy logic based autonomous skid steering vehicle navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, Washington, DC, May 2002, pp. 2171–2177.
[17] X. Yang, M. Moallem, and R. V. Patel, "A novel intelligent technique for mobile robot navigation," in *Proc. IEEE Int. Conf. Control Applications*, Istanbul, Turkey, Jun. 2003.
[18] ——, "An improved fuzzy logic based navigation system for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, Oct. 2003.
[19] H. Seraji, "Fuzzy traversability index: A new concept for terrain-based navigation," *J. Robot. Syst.*, vol. 17, no. 2, pp. 75–91, 2000.
[20] A. Howard and H. Seraji, "An intelligent terrain-based navigation system for planetary rovers," *IEEE Robot. Autom. Mag.*, vol. 17, no. 4, pp. 490–497, Aug. 2001.
[21] H. Seraji, "Rule-based traversability indices for multi-scale terrain assessment," in *Proc. IEEE Int. Conf. Control Applications*, Istanbul, Turkey, Jun. 2003.
[22] D. Pratihar, K. Dep, and A. Ghosh, "A genetic-fuzzy approach for mobile robot navigation among moving obstacles," *Int. J. Approx. Reason.*, vol. 20, pp. 145–172, 1999.
[23] K. K. Tan, K. C. Tan, and K. Z. Tang, "Evolutionary tuning of a fuzzy dispatching system for automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 632–636, Aug. 2000.
[24] S. I. Lee and S. B. Wang, "Emergemt behaviors of a fuzzy sensory-motor controller evolved by genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 6, pp. 919–929, Dec. 2001.
[25] Y. Makita, M. Hagiwara, and M. Nakagawa, "A simple path planning system using fuzzy rules and a potential field," in *Proc. IEEE Int. Conf. on Fuzzy Systems*, Orlando, FL, 1994, pp. 994–999.
[26] T. Hebert, N. Tsourveloudis, and K. Valavanis, "Fuzzy control of an autonomous vehicle utilizing electrostatic potential fields," in *Proc. IEEE Int. Conf. Control Applications*, Trieste, Italy, 1998, pp. 658–662.
[27] N. Tsourveloudis, K. Valavanis, and T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic," *IEEE Trans. Robot. Autom.*, vol. 8, no. 6, pp. 9–17, Dec. 2001.
[28] R. Araujo and A. T. de Almeida, "Learning sensor-based navigation of a real mobile robot in unknown worlds," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 164–178, 1999.
[29] J. Wang and C. Lee, "Self-adaptive recurrent neural-fuzzy control for an autonomous underwater vehicle," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, Washington, DC, May 2002, pp. 1095–1100.
[30] C. Ye, N. H. C. Yung, and D. Wang, "A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 1, pp. 17–27, Feb. 2003.
[31] C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller–Part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–418, 1990.
[32] R. Langari and M. Tomizuka, "Analysis and synthesis of fuzzy linguistic control systems," *Intell. Control*, vol. 23, pp. 35–42, 1990.
[33] N. Pfluger, J. Yen, and R. Langari, "A defuzzification strategy for a fuzzy logic controller employing prohibitive information in command formulation," in *Proc. 1st IEEE Int. Conf. Fuzzy Systems*, San Diego, CA, Mar. 1992, pp. 717–723.
[34] H. Noborio, T. Yoshioka, and S. Yominaga, "On the sensor-based navigation by changing a direction to follow an encountered obstacle," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Grenoble, France, 1997, pp. 510–517.
[35] T. Yata, L. Kleeman, and S. Yuta, "Wall following using angle information measured by a single ultrasonic transducer," in *Proc. IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, 1998, pp. 1590–1596.
[36] *Koala User's Manual*, 2.0 ed., K-Team S.A., Lausanne, Switzerland, 2001.

Xiaoyu (Sherry) Yang (S'04) received the B.Sc. degree in electrical engineering from Beijing University of Chemical Technology, Beijing, China, in 1992 and the M.Sc. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2001. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Western Ontario (UWO), London, ON, Canada.

Mehrdad Moallem (M'00) received the B.Sc. degree from Shiraz University, Shiraz, Iran, in 1986 and the M.Sc. degree from Sharif University of Technology, Tehran, Iran, in 1998, both in electrical and electronic engineering. He received the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 1997.

From 1997 to 1999, he held postdoctoral and research positions at Concordia University and Duke University, Durham, NC. He is currently with the Department of Electrical and Computer Engineering, University of Western Ontario (UWO), Londo, ON, Canada. He is also an Associate Scientist at the Canadian Surgical Technologies and Advanced Robotics (CSTAR) Research Group at UWO Hospital.

Dr. Moallem is a Registered Professional Engineer in the province of Ontario.

Rajni V. Patel (M'76–SM'80–F'92) received the B.Eng. degree in electronics (with first-class honors) from the University of Liverpool, Liverpool, U.K., in 1969, and the Ph.D. degree in electrical engineering from the University of Cambridge, Cambridge, U.K., in 1973.

From 1973 to 1998, he held postdoctoral and faculty positions at the University of Cambridge, Lund Institute of Technology (Sweden), NASA Ames Research Center (U.S.), the University of Waterloo (Canada), Delft University of Technology (The Netherlands), the Control Systems Centre, UMIST (U.K.), and Concordia University (Canada). He is currently a Professor and Tier–1 Canada Research Chair in Advanced Robotics and Control in the Department of Electrical and Computer Engineering, University of Western Ontario, Londo, ON, Canada. He also serves as Director of Engineering for Canadian Surgical Technologies and Advanced Robotics (CSTAR), a research initiative of the London Health Sciences Centre. He has co-authored a textbook and five research monographs on robotics and control, and co-edited *Numerical Linear Algebra Techniques for Systems and Control* (New York: IEEE Press, 1994).

Dr. Patel is currently serving on the Editorial Boards of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and the IEEE TRANSACTIONS ON ROBOTICS. He is a registered Professional Engineer in the Province of Ontario, Canada, and is a Fellow of the ASME.