

A Simulation Environment To Test Fuzzy Navigation Strategies Based On Perceptions

Lía García-Pérez

Instituto de Automática Industrial (IAI)

Consejo Superior de Investigaciones Científicas (CSIC)

28500 Arganda. Madrid. Spain.

María C.Garcia-Alegre

Instituto de Automática Industrial (IAI)*

Abstract

Current work deals with the design and development of a virtual environment to test perception based navigation strategies for a non-holonomic real robot. Physical sensors onboard the robot are described with their main characteristics as virtual perceptual agents. Real environments are represented as polygons to closely mimic geometric shape properties of obstacles and are recorded in a data structure as a virtual world. Elementary and ever complex navigational behaviors such as, Advance, AvoidObstacle and GoToPoint are designed and implemented through fuzzy rule bases to handle the perceptual concepts encapsulated by the perceptual Agents. Some examples of the trajectories described by the robot ROJO are displayed to show the robot performance in different simulated environments.

I. INTRODUCTION

One of the major attributes of an autonomous behavior is motion capability as a way to actively interact with the environment attempting to learn how to optimize time and energy consumption to survive. The well-known paradigm memory versus activity [1] accounts for this reinforcement in the activity to face the lack of knowledge or a scarce representation of the world, adaptation being the main cue of any behavior denoted as intelligent [2]. An increase in the degree of intelligence of artificial systems can be accomplished with the integration of the appropriate sensors to perceive the environment to achieve the proposed goals or level of competence [3].

A great research effort is now being devoted to the acquisition of ever-higher degrees of autonomy in mobile robots to be able to operate in dynamic and partially structured environments, but the complexity and uncertainty inherent to real environments and sensing systems slow down the success.

Computing with words offers a great challenge in the robotics field, as a framework to easily describe and model manual or mental human abilities. In usual every day interactions, human being use a reduced set of words that are related to their perception, deliberative processes and actions. Mobile robots are, ever more, designed to ease the communication with human operators by using commands that correspond with natural language terms and expressions. This is why Fuzzy Logic plays a fundamental role in the design and development of perceptual and reasoning systems in robotic applications.

Computing with words is highly recommended either when the available information is too imprecise to use numbers or when there is a tolerance for imprecision which can be exploited to get tractability and a suitable interface with the real world [4]. Within this paradigm, present works is a contribution to the development and test of navigation strategies based on the perception performed by a real robot in outdoor environments. To this aim a Simulator has been designed as a software tool to aid in the design and development of both virtual sensors and worlds to mimic as much as possible real

environments and robots. A set of virtual sensors are described as well as the results obtained by activating the perceptual capabilities of the virtual robot to validate and debug the fuzzy navigation strategies designed for the real robot ROJO.

II. REAL ROBOT ROJO

ROJO is a commercial low-cost, small size (2.0 x 0.7 m.) and moderate precision lawnmower fully automated at the IAI_CSIC, Fig. 1. It has been gradually endowed of sensors to perceive either its internal state or its relation to the environment.

A. Automation of Robot ROJO.

The automation of ROJO was achieved by substituting the manual steer and brake systems with two electropneumatic systems, composed of an electrovalve, a two effect pneumatic cylinder for each control axle and an air compressor and tank, shared by both control systems.

A fuzzy PD-controller was designed to control the steer-wheels [5]. Inputs are the current and previous error in piston positioning and output the Pulse Width Modulation (PWM) signal to the electrovalve. A high quality potentiometric sensor, coupled to one of the cylinders, perceives the piston position/wheel angle providing the feedback sensorial information. The brake system is a Stop/Start bi-valued uncomplicated on/off controller.

B. ROJO Sensorial Systems

During the automation process of ROJO robot, several sensors have been incrementally installed on board. At present moment they can be grouped in three different categories:

1. Proprioceptive sensors to perceive the internal state of the robot [6]:
 - A battery level sensor has been installed to detect its energy level or “tiredness”.
 - Two potentiometric sensors perceive the “legs muscles state”.



Fig. 1. ROJO Robot.

* This work has been fully funded by the CICYT_TAP98_0781 Research Project: “MultiAgent Architecture AMARA: Complex Behavior Generation For An Outdoor Spraying Robot” and a studentship Grant from the Spanish Ministry of Science and Technology.

2. Redundant positioning sensors

- An odometric sensor specially designed at the IAI-CSIC, well suited for outdoor applications. This sensor provides a continuous odometric position relative to the starting position.
- A DGPS system and a magnetic compass to obtained the absolute positioning of the robot, georeference.

3. Environmental perception sensors.

- A laser range finder, two inclinometers and a bumper.

C. Onboard Processors and Communications.

A PC- AMD K6 II processor has been installed onboard to deal with: (1) Sensor data acquisition, (2) Low level control of the steering and brake and (3) Aerial communication via Radio-Ethernet with IAI LAN clients, using the Window Socket protocol.

Within this framework the Simulator aims to be a useful tool to test and tune navigation strategies based on virtual perceptions. To this goal it has to mimic robot real sensor performances and the basic motion abilities required for navigation, Fig. 2. With the simulator the user can create virtual environments, defining both altitudes and obstacles, locate the robot at an initial position and determine a goal, by just clicking on a window. Different windows are available for displaying sensor measurements, robot location and trajectory.

III. VIRTUAL ROBOT

A Virtual Robot, the kinematic model [7], substitutes the Real Robot in the Simulator. Virtual sensors such as: inclinometers, laser scanner and odometers, Table I., simulate real sensors. Finally, real worlds are implemented as set of polygons of any shape able to reproduce different environmental conditions.

A. Kinematic Model of the Mobile Robot

The kinematic model emulates elementary motion, defined through a set of equations. These equations map the driving variables of the robot to 2D relative coordinates. ROJO can be modeled as a rigid solid that moves on a flat surface with four points of contact with the ground (four wheels).

The Kinematic model maps the steering angle of the front wheels, ϕ , the distance between contact points of the front and back wheels, L , and the velocity, v , to the relative coordinates, (x,y,θ) [8].

B. Virtual Sensors

To simulate the robot it is necessary to include virtual sensors to substitute ROJO real sensors. Three virtual sensors have been designed {odometer, laser scanner, inclinometers}, Table I. Virtual worlds are also required to interact with the robot.

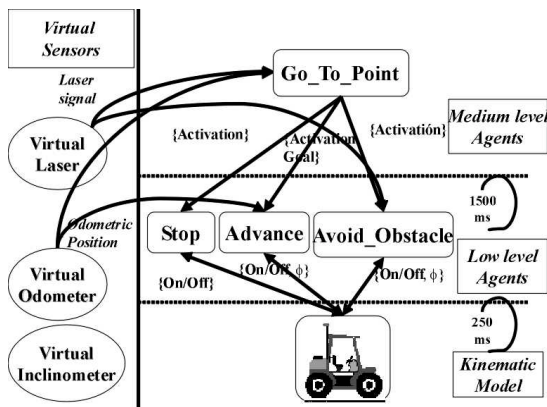


Fig. 2. Simulator Scheme

TABLE I
REAL AND SIMULATED ROBOT SENSORS

	Robot	Odometric Sensor	Inclinometer	Laser Scanner
Real Robot ROJO	Fuzzy STEER_WHEEL On/of BRAKE controllers	32 permanent magnets + Hall effect magnetic sensor	Two capacitive inclinometers	SICK LMS291 laser scanner
Virtual Robot	Analytic kinematic model	Analytic odometry equations and virtual Robot coordinates (kinematic model)	Analytic equations and environment altitude simulation	Analytic equations and simulated world
Agents are identical either for the Real Robot ROJO and for the Virtual Robot				

Virtual Odometer

Current version of the virtual odometer simulates an "ideal" odometer with a specific resolution, but neither systematic nor non-systematic are accounted for. The odometric position is calculated in three steps:

1. Using the robot coordinates, provided by the kinematic model, the traveled distance and the turn performed in the last computing cycle are computed.
2. From the odometer model, right and left wheels traveled distance is computed.
3. Assuming a finite resolution odometer, the robot coordinates are recalculated.

Virtual Inclinometer

To detect the terrain slope, different sloppy terrains are simulated in the virtual world. The user is allowed to interactively define the altitude maps on line. The inclinometer computes the slope translating the pixel color of two different extremes of the vehicle into the corresponding slope.

Virtual Laser Scanner

Virtual world obstacles can be interactively defined by the user just by clicking on a window and the perimeter of the obstacles is modeled with polygonal lines. The virtual laser checks the color of the pixel along lines departing from the front center of the vehicle, with a resolution of 0.5 degrees. The distances to the closest obstacle pixel along each ray form the laser data array.

IV. FUZZY NAVIGATION AGENTS

Fuzzy logic has successfully been applied in classification and control processes that are described through a set of linguistic expressions due to the difficulty to obtain analytical models [9]. Fuzzy sets conveniently model the uncertainty inherent to both the human expert approximate reasoning and the imprecision of the real sensor [10].

Navigation agents have been framed in the multiagent AMARA architecture [11], to generate ever-complex behaviours in robots. Within this model, three low-level fuzzy navigation agents have been implemented: START/STOP, ADVANCE and AVOID_OBSTACLES. Agents at this level communicate with the physical lower level agents in the real robot {BRAKE and STEER_WHEEL}. In the Simulator they send commands to the kinematic model that hold for the two former lower level agents.

The intermediate level agent GO_TO_POINT modulates the low-level agents in order to drive the robot from an initial to a final position, avoiding unexpected obstacles. All simulated agents get the required sensor signals from the appropriate virtual sensors.

A. Agents Structure and Functioning

Agents are embodied into a structure similar to that of real robot agents. They require three types of input variables that have to be sent by either the higher level agents or the virtual sensors:

1. Activation signal, that allows the agent execution, provided by the higher level agents.
2. Input parameters, provided by higher level agents, to modulate its behavior.
3. Sensorial perception provided by virtual sensors.

Navigation agents are designed using fuzzy variables and fuzzy rule base models. Defuzzification is always performed through the gravity centre algorithm.

START/STOP Agent

START/STOP Agent sends the on/off signal responsible for the vehicle stop or start motion. This agent has been included in the real robot [12], to isolate physical devices from the architecture, offering a more natural command language to communicate with humans or higher level agents.

ADVANCE Agent

ADVANCE Agent is an expert in driving the robot from one to another location using a set of rules based on fuzzy perceptions. From the robot motion capabilities and the goal coordinates, two fuzzy perceptions are built: GoalAngle and GoalDistance. These two perceptions are the inputs for the fuzzy rule-based system. Output variables are the Steer_Angle and the Start/Stop commands, that are defuzzified into a steer angle and an on/off numeric command sent to the kinematic model (Virtual Robot), Fig. 3.

The strategy displayed by the ADVANCE agent points to the alignment of the robot with the goal direction, to start moving to the goal. Rules have as antecedent two perceptions, expressed by the fuzzy variables GoalAngle and GoalDistance. GoalDistance has two linguistic values Far and Close, Fig. 4-a, and is used for decision making to stop the vehicle once the goal is reached. On the other hand GoalAngle has a high weight in the wheel angle determination. Three linguistic labels that integrate all possible angles within the goal and the robot locations, Fig 4-b, describe it.

Output variables are Steer_Angle and Stop. Steer_Angle embodies in three labels the wheel angle, Fig. 4-c. Stop decides whether or not the robot should stop or keep on going, and it is described by two singletons, Fig. 4-d. Rules base is displayed in Table II.

AVOID_OBSTACLE Agent

This Agent deviates the robot from the obstacle. It gets the distance and returns either the appropriate wheel angle or the stop command.

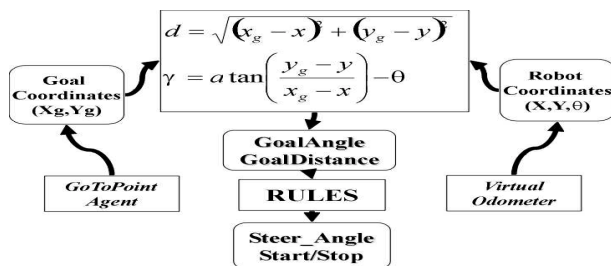


Fig. 3. ADVANCE Agent

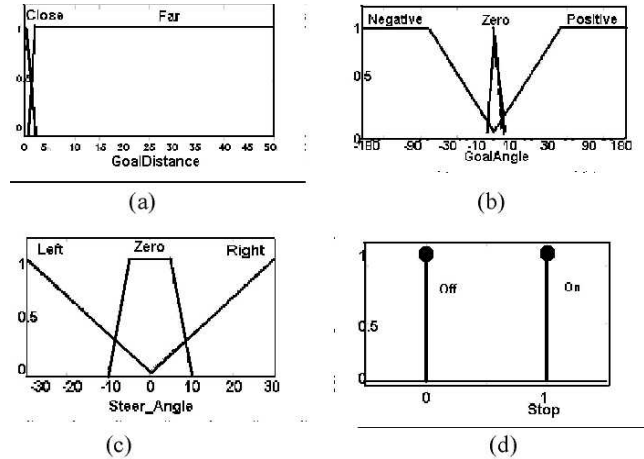


Fig. 4. Input and output variables.

Because of the scarce manoeuvrability of ROJO robot, AVOID_OBSTACLE strategy incorporates an evaluation of obstacle presence in three angular zones. If an obstacle is detected in the front central zone then it checks left and right. Finally, it drives the robot into one of these zones if no obstacle is present, in all different situations it just stops the vehicle.

Three input variables {Center_Distance, Right_Distance, Left_Distance}, each one defined by three linguistic terms, evaluate the perception of free space in the view range of the laser scanner, Fig. 5. Output variables are the same as those of the ADVANCE Agent, Fig. 4c-d. The Rules base or qualitative model decides on the best path to avoid the obstacle by evaluating the distance information in the three zones, Table III.

GO_TO_POINT Agent

The GO_TO_POINT Agent modulates lower level agents: ADVANCE, START/STOP and AVOID_OBSTACLE to drive the robot from its current position to the goal, avoiding unexpected obstacles. GO_TO_POINT needs the virtual odometric location and the visual laser range distance measurements.

It checks for free way in front of the robot, if positive then it switches the control to the ADVANCE Agent. If an obstacle is detected the AVOID_OBSTACLE Agent is activated until free way is again detected. On success or failure it reports to human or higher levels agents.

The simulator has been implemented using C++ Builder for Windows to maintain compatibility with the code already implemented in the real robot. All the algorithms are used in the real robot in real time.

V. RESULTS

Trajectories performed with the GO_TO_POINT Agent have been analyzed, Fig. 7. Results obtained with the Simulator has led us to an improvement of the algorithms and to realize on succes or failures easily than testing directly the algorithms on the Real Robot ROJO.

TABLE II
RULE BASE FOR ADVANCE AGENT

GoalAngle/ GoalDistance	Positive	Zero	Negative
Close	Stop = ON Steer_Angle= ZERO	Stop = ON Steer_Angle= ZERO	Stop = ON Steer_Angle= ZERO
Far	Stop = OFF Steer_Angle= LEFT	Stop = OFF Steer_Angle= ZERO	Stop = ON Steer_Angle= RIGHT

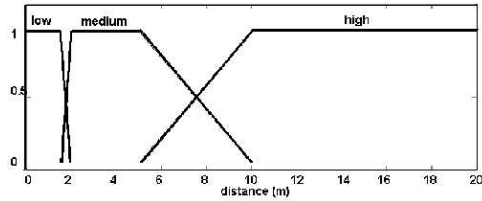


Fig. 5. Membership function for laser distance

TABLE III
RULE BASE FOR AVOID AGENT

Distance_Center	Distance_Right	Distance_Left	Steer_Angle	Stop
High			Zero	Off
Medium	High		Right	Off
		High	Left	Off
		Medium	Zero	Off
	Low	High	Left	Off
		Medium	Zero	Off
		Low	Zero	Off
Low	High		Right	Off
		High	Left	Off
		Medium	Zero	On
	Medium	High	Left	Off
		Medium	Zero	On
		Low	Zero	On
	Low	High	Left	Off
		Low	Zero	On

We have observed that certain goal points are unreachable by the robot. This effect is caused by the low manoeuvrability of the robot that makes impossible to reach a too close goal if the robot has to perform a wide turn. These results are now being improved by adding information about such unreachable locations and suggesting other strategies to reach them.

On the other hand results, Fig. 6a-h, allowed us to tune the fuzzy membership functions that describe the linguistic labels and decide the use of just three membership functions instead of five. Although the followed strategies are very simple, using well tuned parameters the simulated robot is able to navigate either in uncluttered, Fig.6a-d, or in crowded environments, Fig. 6e-h.

VI. CONCLUSIONS

The simulator is a valuable software tool to ease the debugging and tuning processes always required to test different navigational behaviors in a mobile robot. The simulator allows the design of either perceptual or behavioral control strategies based upon the basic set of components offered: virtual robot and sensors and virtual worlds. Perceptual and navigational strategies are easily described in terms of the capabilities of the virtual components.

The modular design of the Simulator permits a gradual growing of the components by adding and activating complementary modules. The facility to test multiple strategies, slightly different, in a short time is a great aid in any tuning process.

The simulator allows the emergence of aspects of navigational behavior not accounted for or never imagined by the human designer under particular sensorial and computational conditions. It gives the possibility to work at a higher granularity level in the development of behaviors, without a deep knowledge on the low-level implementation details.

ACKNOWLEDGMENT

Authors want to thank Prof. L. Zadeh for his seminal ideas, fruitful discussions and constant advice during a short stage of L.G.P. in Berkeley.

REFERENCES

- [1]. R. Bajcsy, "Active perception vs. passive perception". *Proc. 3rd IEEE Workshop on Computer Vision: Representation and Control*, pp.55-59, 1985.
- [2]. M.McFarland, T. Bossert, *Intelligent Behavior in Animal and Robots*, MIT Press, Cambridge,1993.
- [3]. R. Brooks, "A robust layered control system for a mobile robot". *IEEE Journal of Robotics and Automation*, RA-2, pp.14-23, 1986.
- [4]. Zadeh L. "From computing with numbers to computing with words -from manipulation of measurements to manipulation of perceptions." *IEEE Transactions on Circuits and Systems -I: Fundamental Theory and Applications*, 45(1), pp. 105-119, 1999.
- [5]. Garcia-Perez L., Cañas J.M., Garcia-Alegre M.C., Yañez P., Guinea D., "Fuzzy Control of an electropneumatic actuator." In: *Proc of the Spanish Conference on Fuzzy Logic, ESTYLF2000*, pp. 133-138, Sevilla, Sep. 2000.
- [6]. Everett. *Sensors for mobile Robots*. John Wiley and Sons, 1995.
- [7]. Garcia-Alegre M., Ribeiro A., Gasos J., Salido J. "Optimization of fuzzy behavior based robot navigation in partially known industrial environments." *Third Int. Conf. On Industrial Fuzzy Control and Intelligent System*, Houston TX. pp. 50-54, 1993.
- [8]. Laugier C.Th., Fraichard Ph., Garnier I.E., Paromtchik, Sheuer A. "Sensor-based control architecture for a car-like vehicle." *Autonomous robots 6*, pp. 165-185, 1999.
- [9]. Sugeno M., "An Introductory Survey of Fuzzy Control", *Information Science*, 36, pp.59-83, 1985.
- [10]. Klir, G.I., Yuan, B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995
- [11]. Garcia-Alegre M., Recio F, Basic "Visual and Motor Agents for Increasingly Complex Behavior Generation on a Mobile Robot." *Autonomous Robots 5*, pp. 19-28, 1998.
- [12]. Garcia-Alegre M., Ribeiro A., Garcia-Perez L., Martinez R., Guinea D., Pozo-Ruz "An. Autonomous robot in agriculture tasks", in *Proc. Of the 3th European Conference on Precision Agriculture, 3ECPA*, vol. 1, pp. 25-30, Montpellier, June 2001.

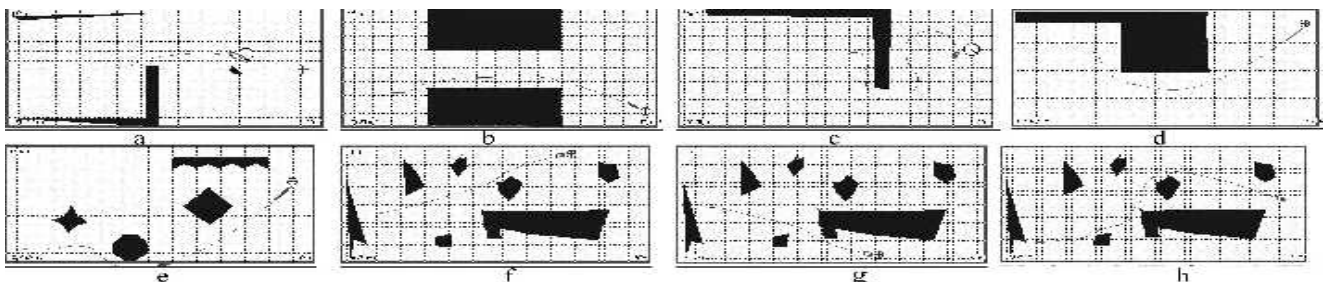


Fig. 6. Results.