

Knowledge-based Intelligent Diagnosis of Ground Robot Collision with Non Detectable Obstacles

José M. Alonso · Luis Magdalena · Serge Guillaume · Miguel A. Sotelo · Luis M. Bergasa · Manuel Ocaña · Ramón Flores

Received: 14 June 2006 / Accepted: 27 November 2006 /
Published online: 23 January 2007
© Springer Science + Business Media B.V. 2007

Abstract The paper describes the application of fuzzy techniques to analyze motion problems in a mobile robot. The robot is equipped with ultrasound sensors used for obstacle detection, but, in some cases, small obstacles are out of the range of the sensors and can be dragged by the robot without being detected. Using other variables as, measured velocity, undershoots of that velocity or changes in battery voltage, a fuzzy system is able to determine those situations. The paper

J. M. Alonso (✉) · L. Magdalena
ETSI Telecomunicación, Universidad Politécnica de Madrid (UPM),
Ciudad Universitaria s/n, 28040 Madrid, Spain
e-mail: jmam@mat.upm.es

L. Magdalena
European Centre for Soft Computing, Edificio Científico-Tecnológico,
C/ Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Asturias, Spain
e-mail: luis.magdalena@softcomputing.es

S. Guillaume
Cemagref Montpellier, BP 5095, 34196 Montpellier Cedex, France
e-mail: serge.guillaume@montpellier.cemagref.fr

M. A. Sotelo · L. M. Bergasa · M. Ocaña · R. Flores
Department of Electronics, Escuela Politécnica Superior, Universidad de Alcalá,
Campus Universitario s/n, Alcalá de Henares, Spain
e-mail: michael@depeca.uah.es

L. M. Bergasa
e-mail: bergasa@depeca.uah.es

M. Ocaña
e-mail: mocana@depeca.uah.es

R. Flores
e-mail: flores@depeca.uah.es

also analyzes the knowledge extraction process for the application using expert and induced knowledge (from data collected during navigation tasks) in a cooperative way, dealing with integration and simplification issues. The expert knowledge was used for describing the robot behaviour in order to identify the variables that should be used with the aim of detecting a collision of the vehicle against an undetected obstacle, as well as proposing a suitable recovery action. Data collected in real trials were used for inducing knowledge so as to complete and validate the expert knowledge. Both kind of knowledge were integrated in the final fuzzy-based system. The aim is to build up a knowledge base, which is interpretable and accurate at the same time, and it is used by our fuzzy system in order to solve the motion problems under consideration.

Key words accuracy-interpretability tradeoff · expert and induced knowledge · fuzzy logic · mobile robot

1 Introduction

It is widely admitted that autonomous robots, independent of their application, must have efficient locomotion systems (low power consumption subsystems, highly precise sensors, and large autonomy batteries are the essential key points), reliable navigation and operational systems, and be able to work safely in their environment. Thus, the technology required to realize robust, reliable and safe robots is given considerable worldwide attention. As a consequence, the use of autonomous or semi-autonomous robots in real applications is only possible when those robots exhibit a certain level of intelligence, being capable of fulfilling the previous requirements.

Soft computing techniques have been considered as a way of adding that level of intelligence from different points of view. The most common approach integrating soft computing techniques in mobile robotics is that of application in navigation [1, 13] and control [9, 22]. Some other approaches have considered these techniques in sensor information processing (ultrasound, vision, ...) applied to localization [6, 20], path following (corridors, walls) or obstacle avoidance [23]. In some cases [10], the fuzzy techniques are introduced at different levels (control, path planning, and obstacle avoidance).

The use of fuzzy techniques in diagnosis problems has been considered [19] but mostly in the field of automation, without considering autonomous robots. It is also possible to find similar problems in the automobile industry, where some fuzzy approaches have been applied [21, 27]. When considering a semi-autonomous robot working in real conditions, the integration of any kind of on-board fault diagnosis system seems to be quite important. Such systems are needed to prevent abnormal behaviour that would damage either the robot or the environment. In that sense, the effort is quite reduced and only a few applications have been proposed considering model based diagnosis, mostly using artificial neural networks, and centred on the diagnosis of actuator problems. Duan [7] presents a recent survey on fault detection and diagnosis for wheeled mobile robots. Under normal circumstances, all sensorial capabilities on board the robot work properly and they provide valuable and sufficient information about the environment so as to correctly perform obstacle avoidance manoeuvres. Nonetheless, there are some obstacles (short and heavy ones,

indeed) that are not detected by sensorial means. As a result, in case the robot finds an obstacle of this type along its way, the vehicle will inevitably collide with the obstacle, and the collision could damage the robot.

In order to focus on this problem, a general architecture for integrating fault diagnosis and recovery modules into autonomous robots has been developed in the framework of the European research project ADVOCATE II [28], whose main goal is to increase the safety, the efficiency and the reliability of autonomous robotic systems. As pieces of this architecture, different Artificial Intelligence based modules devoted to diagnosis and recovery are developed. Neuro symbolic systems, Bayesian belief networks, and Fuzzy systems are developed and integrated into the existing control architecture of different robotic platforms.

The present paper describes one of the application problems of the ADVOCATE II project: Diagnosis of collision of a ground robot with non detectable obstacles, with the aim of providing recovery actions. The diagnosis module is made up of a fuzzy system whose originality relies on its designing process: It is based on expert knowledge, but integrates data sample information.

The strength of Fuzzy Inference Systems (FIS) relies on their twofold identity: On the one hand they are able to handle linguistic concepts; on the other, they are universal approximators able to perform non linear mapping between inputs and outputs. These two characteristics have been used to design two kinds of FIS. The first kind of FIS to appear focused on the ability of fuzzy logic to model natural language [24]. These FIS contain fuzzy rules built from expert knowledge; they can be seen as a fuzzy extension of expert systems. Sugeno [30] was one of the first to propose self learning FIS and to open the way to a second kind of FIS, those designed from data. As this field of fuzzy logic has become very popular, a lot of methods are available [14].

In this framework we propose combining both the advantages of fuzzy inference system. Their natural language ability is used to collect expert knowledge as qualitative linguistic rules. Their learning capability is used to induce rules from data. Thanks to fuzzy logic interpretability, the human expert is able to check the validity of the induced rules.

The proposed process is iterative, starting from expert knowledge; the knowledge base is then enriched by induced rules with supervision of the expert. This process involves various types of data management: Experimental data to train the system, simulation data to test the rule base containing both types of rules and finally tests in real world environment.

The use of different kinds of information (expert and induced knowledge from data provided by the sensorial capabilities of the vehicle) leads us to think about the JDL model [29, 32], a model focused on data fusion from several sensors and information sources developed by the US Joint Directors of Laboratories (JDL). This model, indeed, is not an architecture, but only a framework for data fusion system. Although the topic of this work is not a data-fusion system, the proposed system may represent a good example of implementation of higher levels of JDL model. However, this work does not go into details about JDL framework for several reasons: First, our fuzzy system is added and integrated into an existing control architecture, on the other hand when a system with data fusion is constructed in many cases the sensors and the communication structure are decided beforehand. Second, the problem under analysis is solved in a high abstraction level through using

a compact and easily understandable fuzzy knowledge base; on the contrary realistic implementations of data fusion processing often are complex. Finally, data fusion is useful if complementary types of sensors detect the same object, which depends on the availability of a big number of sensors and on a correct association of data. Our system let us increase the robustness, reliability, and efficiency of the robot, without adding more sensors.

The structure of the paper is as follows. Section 2 briefly describes the diagnosis problem considered in this work, related to motion problems produced when the robot collides with non detected (due to a reduced height) obstacles. Then, Section 3 analyzes the knowledge extraction process considering only the expert knowledge. After that, the introduction of induced knowledge is considered in Section 4. This section includes data generation in order to test the suitability of the expert knowledge base, rule induction to extract additional knowledge from data, and the fusion of both rule bases. The section ends with a complete analysis of the obtained results. Finally, Section 5 highlights the main conclusions.

2 Problem Description

The application described in this paper was developed in the framework of ADVOCATE II project. This project aims to define a generic software architecture for intelligent diagnosis in autonomous robots. We concentrate on the application related to the ground robot BART (Basic Agent for Robotic Tasks), a Pioneer2-AT robot, see Fig. 1. The robot has the following configuration: Orinoco PCMCIA Gold wireless card, Linux Red Hat 9.0 operating system, wireless tools for WIFI managing, Orinoco driver patch, a 16 ultrasound sensor ring, and a SONY pan-tilt-zoom camera.

BART is a ground vehicle commanded in teleoperated mode (a Remotely Operated Vehicle) by a human operator from a base station through a wireless link. The vehicle is intended to perform surveillance tasks after hours in a large building composed of corridors, halls, offices, laboratories, etc.

Fig. 1 Basic Agent for Robotic Tasks (BART)



The overall approach of the project envisages different motion problems to be detected. The problems considered in the case of BART are: Vehicle trapped, limited capability of motion, or actuator blocked (tangled).

A deep analysis and understanding of the problem to be solved can only be achieved if a detailed description of the vehicle dynamics and its interaction with the environment is carried out. The model for the AGV (Autonomous Ground Vehicle) is a classical and simple one in the ground robotics domain. The AGV has four driving wheels, two at each side of the vehicle mechanically connected and constrained to move at the same velocity. Thus, the robot presents a differential kinematic model where in-place rotation is feasible. Concerning the nomenclature of the variables involved in the robot kinematic model, v stands for the vehicle linear velocity, and Ω represents the rotational or angular velocity. According to this, the commanded velocities for the left and right wheels of the vehicle, $\omega_{leftcmd}$ and $\omega_{rightcmd}$ (in rad/s), can be easily calculated based on the commanded linear and rotational velocities, V_{cmd} and Ω_{cmd} , as shown in Eq. 1.

$$\begin{aligned} \omega_{rightcmd} &= \frac{V_{cmd}}{R} + \frac{\Omega_{cmd}L}{2R} \\ \omega_{leftcmd} &= \frac{V_{cmd}}{R} - \frac{\Omega_{cmd}L}{2R} \end{aligned} \tag{1}$$

where L stands for the inter-wheels distance (vehicle width) and R is the wheel radius. The commanded velocities V_{cmd} and Ω_{cmd} , are the inputs to the system while the resulting velocities V_{res} and Ω_{res} , constitute the outputs and can be finally reconstructed by inverse kinematic using Eq. 2.

$$\begin{aligned} V_{res} &= \frac{R}{2}(\omega_{rightres} + \omega_{leftres}) \\ \Omega_{res} &= \frac{R}{L}(\omega_{rightres} - \omega_{leftres}) \end{aligned} \tag{2}$$

where $\omega_{leftres}$ and $\omega_{rightres}$ stand for the resulting angular velocities of the left and right wheel respectively. There exists a small time delay td between the issuing of the commanded angular velocities of the wheels ($\omega_{leftcmd}$, $\omega_{rightcmd}$) by the active task (after kinematic conversion from V_{cmd} , Ω_{cmd}) and their real application to the system. This time delay must necessarily be considered, especially for simulation purposes, as it has a non negligible effect on the overall system dynamics. The low level velocity controller is implemented by a separate PI (Proportional-Integral) controller providing the input control voltage for each wheel pair motor ($V_{control,left}$, $V_{control,right}$). The coefficients of the PI controllers are empirically tuned. On the other hand, a Kalman filter is utilised for filtering the measured velocities of each wheel pair $\omega_{leftmeas}$ and $\omega_{rightmeas}$ so as to provide the quasi noise free estimations, $\omega_{leftres}$ and $\omega_{rightres}$, that serve to compute V_{res} and Ω_{res} by inverse kinematics as previously mentioned.

The paper describes a diagnosis system whose objective is to provide, when the vehicle piloting module generates and sends an alarm, one of the diagnoses described below. This alarm is provided whenever the vehicle piloting module¹ detects some

¹The vehicle piloting module is one of the elements of the ADVOCATE II architecture.

disturbances in the measurements that could be related to a collision not detected by the ultrasound array. Alarms are produced after any potential faulty situation, and are based on a low level analysis. So, diagnosis modules deploy a deeper analysis to disregard false alarms or to determine the specific problem that occurred. Several diagnosis modules are integrated in the system, but this paper concentrates on only one of them.

To build up this diagnosis system we choose the fuzzy logic formalism [33] for its well known ability for linguistic concept modelling. The fuzzy rule expression is close to expert natural language, while, being universal approximators, fuzzy inference systems can also be used for knowledge induction processes.

The potential diagnoses considered by the system are:

- *Normal*: It means that a false alarm has been received, as no real collision has occurred.
- *Vehicle drags obstacle*: The vehicle has collided against an obstacle not heavy enough to block vehicle movement. Thus, after a transient interval the vehicle controller regains the velocity commanded and keeps on moving by dragging the obstacle on its way.
- *Vehicle stalled*: In this case the obstacle is heavy enough to impede the vehicle from moving. The vehicle, stopped as a result of the collision, gets trapped by the obstacle.
- *Test needed*:² It may happen that no accurate diagnosis can be issued based on the available information. In this case, a test is recommended in order to gather further data so as to launch a more precise diagnosis. The tests are mainly oriented to discriminating whether the vehicle is moving or not. Based on this, slippery situations on the vehicle wheels can be easily deduced. In most of the cases, the test involves additional manoeuvres or measurements dedicated to diagnosis, requiring the robot mission postponement. Consequently, tests will only be considered if primary data are not enough for accurate diagnosis.

The problem under consideration in this paper requires the integration of both, expert and induced knowledge, since none of these sources of information seems to offer a complete view of the problem. These two approaches, expert and data driven knowledge, will be considered as complementary sources of information prioritizing expert knowledge.

Once diagnosis is made, if and only if it is a critical diagnosis, i.e. it corresponds to *Vehicle stalled* or *Vehicle drags obstacle*, the system will provide a recovery action so as to rid the vehicle of the obstacle and resume the mission. Three situations are analyzed:

1. *Liberate Left*: A liberation manoeuvre is started, if feasible (there is enough space on the left in the surrounding area), by moving BART backward, then slightly to the left, until the vehicle can regain the intended track and resume the mission.

²It is important to notice that Test Needed is not an output of the inference system, it is a post processing output produced when inferred values for several diagnoses are very similar. By default, Test Needed is not activated.

2. *Liberate_Right*: In this case the same liberation manoeuvre is made on the right side.
3. *Go_Back*: When there is not enough space in the surroundings for the vehicle to avoid the obstacle and proceed the mission. For example, the robot is moving in a narrow corridor.

BART is able to determine the state of the environment according to information provided by the ultrasound array. Thus, depending on the free space in the place (corridor, hall, office, ...), the suitable recovery action is achieved. Nevertheless, the implementation is not detailed in this paper where the main topic deals with collision diagnosis. Although the diagnosis system developed in this work is ideal to be applied indoors, it can also be used in outdoor environments whenever the terrain is not uneven.

3 Expert Knowledge Extraction

Usually, knowledge based systems are considered when there is a difficulty in describing or analyzing a model of the problem with an appropriate accuracy, or when the uncertainty of the process advises against such an approach. However, the use of a model (even one that is approximate or partial) may provide us with a qualitative view of the problem, and can serve as a method to extract such qualitative knowledge by appropriate interaction with the expert. In most cases that will not be enough to solve the problem but, it will provide us with quite useful information to determine involved variables, as well as qualitative behaviour rules.

The expert knowledge extraction process can be kept at a “high” abstraction level. The process can be limited to the expert domain, without considering implementation details which are not part of expert knowledge. For example the expert can only indicate the number of linguistic terms he needs for a given variable without defining the corresponding fuzzy sets.

The overall approach to knowledge extraction is that defined in [17]. First of all, we will consider the qualitative modelling of the behaviour of the robot upon obstacle collision. This covers the first step of the expert knowledge extraction, according to what we have stated in the previous paragraph.

3.1 System Behaviour Upon Obstacle Collision

The interaction between the vehicle and the colliding obstacle is of particular importance in order to appropriately identify and characterize the different variables involved in the process, and the expected global behaviour of the system. On the other hand, diagnosis on motion problems should be performed based on the observation of commanded and measured variables such as vehicle linear and angular velocities, battery voltage, etc, making the task of diagnosis quite complex. As described in the following paragraphs, the global system dynamics are highly non linear and difficult to model with accuracy.

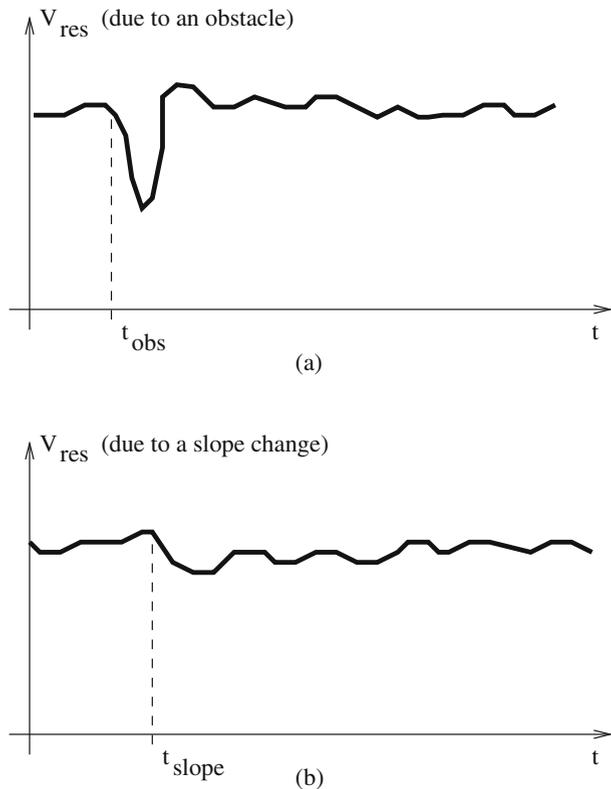
Before starting the description of the system behaviour, it is mandatory to summarize the nomenclature that relates the dynamics of the system and the knowledge of the fuzzy system. The variables involved in the vehicle dynamics that will also be considered in the knowledge base, are basically those related to velocity

(commanded or measured, and linear or angular) as well as those derived from these variables. The list and the correspondence are provided below, where the name relates to the variable used by the fuzzy system and the symbol to the variable of the analytical study:

- *Measured_linear_velocity* corresponds to v_{res} .
- *Commanded_linear_velocity* corresponds to v_{cmd} .
- *Measured_angular_velocity* corresponds to ω_{res} .
- *Undershoot_depth* corresponds to the depth of the undershoot in v_{res} .
- *Undershoot_width* corresponds to the width of the undershoot in v_{res} .

An obstacle collision may induce a change in the vehicle velocity with regard to the commanded value. This is true, even after using a Kalman filter to remove noise from velocity measurements, as mentioned in the previous section. But, this is not the only source for such behaviour, since a change in vehicle velocity also occurs during uphill-downhill slopes. These two situations are illustrated in Fig. 2, and are distinguishable through a visual analysis. A colliding obstacle produces a sudden variation of vehicle linear velocity that the low level velocity controller rapidly compensates for, producing a short but deep undershoot in the value of v_{res} , as depicted in Fig. 2a. This situation is different from that caused by a change of slope as shown in Fig. 2b, where a change of 10 degrees in slope occurred. There is a very

Fig. 2 Variation of robot linear velocity due to a colliding obstacle that is dragged by the robot (a) and a change of slope (b)



clear difference between a velocity modification caused by a colliding obstacle and a velocity modification due to a change in slope. In the case of colliding obstacles, an abrupt and fast velocity drop takes place in almost all cases, for obstacles whose weight is above 1 kg. For very light obstacles, the effect of the collision is negligible and no intelligent action is needed. On the contrary, a change in slope always causes a very smooth velocity variation. Indeed, the variation is so smooth that it is absolutely indistinguishable from normal robot velocity variations during navigation. Besides, variations in slope are very small in real indoor environments where this robot is supposed to operate. In conclusion, the only remarkable external effect over robot velocity during navigation is that caused by collision with heavy obstacles.

Under these circumstances, vehicle consumption will be greater than necessary (due to the over-actuation required to compensate the overload produced by the obstacle) leading to an undesired situation that should be detected and corrected. Remember that we are considering the case of small obstacles that cannot be detected by the obstacle avoidance system on-board the robot (ultrasound based), and thus, only commanded and measured variables can be used to infer that a collision has occurred. For this purpose, the sudden variation of vehicle velocity caused by the collision, followed by the fast actuation of the low level controller that takes the velocity back to its commanded value, will be a key for diagnosis.

The dynamics of vehicle velocity, as a function of an external torque T_{WL} (representing the effect of the collision or the slope), can be described by computing the relation between T_{WL} and the angular velocity of the wheels ω_{res} . Equation 3 provides the expression for $\omega_{res}(s)/T_{WL}(s)$ in Laplace domain, assuming no saturation of the variables, as demonstrated in [25].

$$\frac{\omega_{res}(s)}{T_{WL}(s)} = \frac{R_a + L_a s}{(R_a + sL_a) \cdot (sJ + B) + k_T k_E} \tag{3}$$

R_a and L_a represent the internal electrical resistance and inductance components in the motor armature, k_T is the motor torque constant, k_E stands for the motor voltage constant, and J and B denote the total equivalent inertia and damping, respectively of the motor-load combination. Parameters J and B depend on several factors such as vehicle geometrical configuration, vehicle weight, and the operation surface (texture or roughness). Consequently, the estimation of the exact values of J and B are difficult to compute for use in a model based approach. Appropriate handling of Eq. 3 leads to the simplified expression in Eq. 4.

$$\frac{\omega_{res}(s)}{T_{WL}(s)} = \frac{R_a + L_a s}{k_T k_E (s^2 \tau_m \tau_e + s \tau_m + 1)} \tag{4}$$

τ_m and τ_e stand for the mechanical and electrical time constant of the system, respectively, and are computed as shown in Eq. 5.

$$\begin{aligned} \tau_m &= \frac{R_a \cdot J}{k_T k_E} \\ \tau_e &= \frac{L_a}{R_a} \end{aligned} \tag{5}$$

Equation 4 can then be approximated by the expression in Eq. 6 as τ_m is several hundred times greater than τ_e in practice. Thus, it can be demonstrated that vehicle velocity upon obstacle collision follows approximately a first order dynamic response

with time constant described by τ_m . Accordingly, the duration of the undershoot is more or less proportional to τ_m . Likewise, τ_m is a function of J , the system inertia, which depends on the obstacle weight among other parameters. This means that the heavier the obstacle, the higher the value of τ_m and, consequently, the wider the duration of the undershoot. This simple reasoning could be used for diagnosis purposes in order to detect the collision and, possibly, to make the difference between a heavy and a non heavy obstacle. But this is only an approximate reasoning, as long as the controller dynamics plays a predominant role in the process, and should mandatory be considered.

$$\frac{\omega_{res}(s)}{T_{WL}(s)} \approx \frac{R_a + L_a s}{k_T k_E (s \tau_m + 1)} \quad (6)$$

On the other hand, a similar reasoning will be followed so as to find a relation between the maximum undershoot $\Delta\omega$ and the system parameters. In a first and intuitive approach, it can be stated the $\Delta\omega$ depends basically on two variables:

1. External torque produced by the obstacle T_{WL} , which is a function of the obstacle weight, geometry, position, and orientation.
2. Vehicle velocity.

After the collision, the total amount of system energy should remain unaltered. This implies that the condition expressed in Eq. 7 should be met (an elastic conservative collision is assumed for simplicity).

$$m \cdot v_i = (m + m_{obs}) \cdot v_f \quad (7)$$

m and m_{obs} represent the mass of the vehicle and the obstacle, respectively, v_i stands for the initial velocity of the vehicle (before the collision), and v_f is the final velocity of the system composed by the vehicle and the obstacle after the collision. The variation of velocity Δv experimented by the system (assuming no initial reaction of the velocity controller) can be derived from Eq. 7 yielding the expression in Eq. 8. Obviously, $\Delta\omega$ should be proportional to Δv ($\Delta\omega = k \Delta v$).

$$\Delta v = v_i - v_f = v_i \cdot \left(\frac{m_{obs}}{m + m_{obs}} \right) \quad (8)$$

Two main conclusions can be derived from Eq. 8: On the one hand, the heavier the obstacle the higher the amplitude of Δv , i.e., the higher the amplitude of the undershoot. On the other hand, the higher the initial velocity of the vehicle the higher also the undershoot. Although these are simple approximate statements, as no assumptions about the friction coefficient have been made, they can serve as a basic support for linguistic reasoning providing useful information in order to construct an intelligent diagnosis module. Besides, the velocity controller dynamics should also be accounted for, as it has a strong influence in the settling time needed to restore the commanded velocity after the collision. Throughout the previous reasoning process no initial reaction of the controller has been assumed. In other words, the effect of the collision has been divided in two stages: In a first stage, the vehicle velocity gets decreased to a final value as computed in Eq. 7. During that short interval the controller is assumed not to have enough time for reaction. In a second stage, the controller pushes the vehicle velocity back to its commanded value. The duration of the second interval depends also on the obstacle weight.

Due to the existence of non linearity in the system, which is difficult to model and identify, and considering the previous statements concerning the linguistic relation between the velocity undershoot amplitude and duration and the occurrence of a collision, the use of a fuzzy logic based diagnosis module becomes apparent and convenient.

3.2 Expert Knowledge

To deal with complex problems such as robot motion, expert knowledge is of prime importance. The expert knows the main influential variables and is able to describe their behaviour. This kind of information has been used in the previous section to define a behaviour model that will now be considered to create qualitative rules.

The first step is the definition of the number and nature of variables involved in the diagnosis. The second step is the definition of qualitative diagnosis rules.

3.2.1 Linguistic Variables

Let us consider now those previously defined variables and introduce some others related to additional knowledge.

As it has been previously shown (Fig. 2), a fast but deep undershoot in vehicle velocity takes place upon collision with an obstacle, until the velocity controller regains the commanded reference. This constitutes the key hint to properly providing a diagnosis. Consequently, depth of velocity undershoot (*Undershoot_depth*) upon collision is a crucial variable for determining whether the vehicle has really collided with an obstacle that is being dragged, or on the contrary, whether the undershoot is due to measurement noise or entering uphill slopes.

The depth of the velocity undershoot is tightly related to the commanded vehicle velocity as deep peaks occur at high velocity while small ones take place at low speed, as have been analytically considered in Eq. 8. As a consequence, this undershoot will be considered in percentage instead of using the absolute value.

One more hint is the effect of collisions on battery voltage. Battery voltage suffers a decrease when colliding against an obstacle. This decrement is directly linked to the vehicle consumption that should increase upon collision to regain the commanded speed, subsequently causing the battery voltage to go slightly down. Under specific circumstances, the contribution of some subsystems of the robot to energy consumption should also be considered. As in the previous case, this variable needs some pre-processing made by comparison of the mean value of battery voltage over an interval before and after peak (minimum) value of velocity. The obtained value is defined as *Decrease_of_battery_voltage*.³

A ring of 16 ultrasound based sensors is used to provide range measurements around the robot. The analysis of those range measurements as well as their derivatives, provide us with information concerning robot movement with respect to its environment. To do so, the variables *Range_sonar* and *Derivative_of_range_sonar* (the absolute value is considered) are introduced.

³This variable is computed as the difference between the Battery Voltage, just before the peak, minus that Voltage just after the peak. So, a positive value means a decrease of voltage, while a negative one represents an increase of voltage.

Table 1 Input variables

Variable	Interest range	Labels	Linguistic terms	Units
Undershoot_width	[1, 5]	7	null, very small, small, medium, medium large, large, very large	ms
Undershoot_depth	[10, 70]	7	null, very small, small, medium, medium large, large, very large	%
Decrease_of_battery_voltage	[0, 0.8]	7	zero, very small, small, medium, medium large, large, very large	v
Range_sonar	[20, 3000]	2	zero (20), NOT(zero) (30)	mm
Derivative_of_range_sonar	[0, 1500]	2	zero (0), NOT(zero) (50)	
Commanded_linear_velocity	[0, 250]	6	zero, very low, low, medium, high, very high	mm/s
Measured_linear_velocity	[0, 250]	6	zero, very low, low, medium, high, very high	mm/s

In a first step, expert knowledge extraction process was kept at a “high” abstraction level. The expert defined linguistic variables, their ranges, and the number of linguistic terms needed for each given variable without defining the corresponding fuzzy sets. Table 1 shows the main parameters of input variables defined by the expert. *Undershoot_width*, *Undershoot_depth*, and *Decrease_of_battery_voltage* have 7 labels because initially, the expert did not know how to model their behaviour. This number, seven, has been considered as providing enough granularity for the analysis. *Range_sonar* and its derivative, *Derivative_of_range_sonar*, have 2 labels because they provide information concerning whether the robot is moving or not with respect to its environment. Finally, *Measured_linear_velocity* and *Commanded_linear_velocity* have 6 labels, since the usual operation of the robot is based on the range from 0 to 250 considering changes of commanded speed in steps of 50 mm/s.

As expert rules use linguistic terms, it is of prime concern to design highly interpretable fuzzy partitions. The necessary conditions for interpretable fuzzy partitions have been studied by several authors [5, 8, 12, 15, 26]. In short, the variable definition should satisfy at least the next conditions:

1. Define a small number of labels, linguistic terms, for each variable.
2. Link linguistic terms and fuzzy sets without ambiguity.
3. Work with normal membership functions. These kinds of functions satisfy the Eq. 9, where U is the universe, N represents the number of labels, and $\mu_{A_i}(x)$ is the degree of membership of x to the fuzzy set A_i .

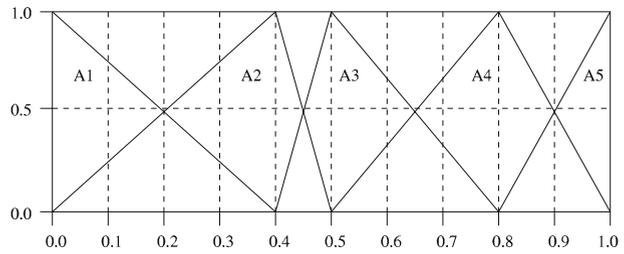
$$\forall i = 1, 2, \dots, N, \exists x \in U, \mu_{A_i}(x) = 1 \quad (9)$$

4. Use membership functions that cover the entire universe, with an overlapping which ensures Eq. 10, where ϵ is the coverage degree.

$$\forall x \in U, \exists i : 1 \leq i \leq N, \mu_{A_i}(x) > \epsilon, \epsilon \geq 0 \quad (10)$$

5. Define convex fuzzy sets, which are defined by Eq. 11 in [33]. Note that although non-convex fuzzy sets are not used in our approach, they are considered in other

Fig. 3 Strong fuzzy partition with five fuzzy sets



contexts [11]. For instance, they let us represent the mealtimes (breakfast, lunch, and dinner) by time of day.

$$\forall x, y \in U, \lambda \in [0, 1], \mu_{A_i}(\lambda x + (1 - \lambda)y) \geq \min(\mu_{A_i}(x), \mu_{A_i}(y)) \quad (11)$$

We use strong fuzzy partitions, as illustrated in Fig. 3. These partitions satisfy all the stated conditions, with a coverage level $\epsilon = 0.5$, and moreover:

$$\forall x \in U, \sum_{i=1}^N \mu_{A_i}(x) = 1 \quad (12)$$

In the present case, an initial family of fuzzy partitions has been produced using regular strong fuzzy partitions adapted to the range defined by the expert. For that purpose, in addition to the physical range of the variable, the expert will define a range of interest to fit more adequately those values of the variable that represent useful information to solve the considered problem. So, regular strong fuzzy partitions [26] in the range of interest were generated for each input variable, with a number of fuzzy sets equal to the number of linguistic terms given by the expert. In addition, for *Range_sonar* and *Derivative_of_range_sonar*, as their ranges are too broad and only two labels are considered, the expert defined the modal points, i.e. the fuzzy set centres (see values in brackets in Table 1, attached to each linguistic term of these variables).

3.2.2 Linguistic Rules

The next step of expert knowledge extraction process consists in writing linguistic rules using the linguistic terms.

As a first example, let us consider the use of range measurements to get information about a possible situation of a stalled robot. To gain further knowledge on how expert rules are stated, let us determine whether the vehicle is moving or not. The following options are possible:

1. The value of variable *Range_sonar* is different from zero (something is detected within the detection range) and its derivative (that will be considered in absolute value without any special filtering, only the low level filtering made by the software on board the robot) is different from zero. It could mean that the robot is moving in a static environment, that the robot is moving in a dynamic

- environment, or that the robot is not moving but the environment is dynamically changing (due to some moving obstacle). Consequently, no deterministic diagnosis could be provided under these circumstances.
2. The value of variable *Range_sonar* is different from zero (something is detected within the detection range) and its derivative is zero. This means that the vehicle is not moving.
 3. The value of variable *Range_sonar* is zero (nothing is detected within the detection range). In this case there is no information at all about the environment, and thus, no diagnosis could be issued either.

According to the three previous possibilities an expert rule could be constructed by considering the following reasoning. If range measurements are different from zero and their derivatives are zero then the environment surrounding the robot is not changing. If on these circumstances the vehicle odometry system measures a velocity equal to zero while being commanded with a linear velocity different from zero, it can be deduced that the vehicle is stalled. The rule can be formalized as follows:

IF *Undershoot_depth* is very large **AND**
 Range_sonar is not(zero) **AND**
 Derivative_of_range_sonar is zero **AND**
 Commanded_linear_velocity is not(zero) **AND**
 Measured_linear_velocity is zero
THEN *Vehicle_stalled*

This piece of knowledge contains several fuzzy propositions that are affected by one of the main questions considered when designing fuzzy systems: the use of negation. This paper will not analyze this situation in deep, but there are two possible interpretations of the next fuzzy proposition:

Commanded_linear_velocity is NOT(zero)

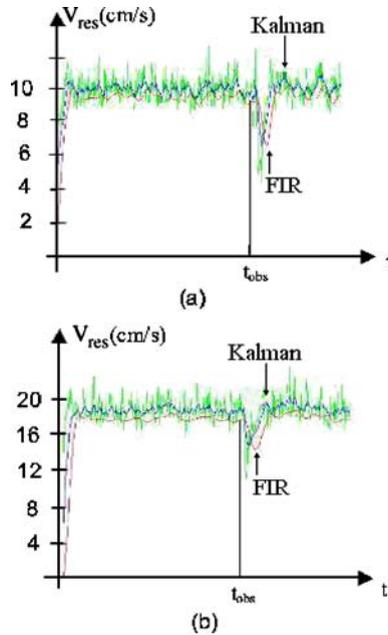
NOT can be interpreted at the level of membership functions (numerical level) or at the level of terms (linguistic level). We work at the second level where NOT(zero) is computed as an OR combination of the remaining linguistic terms (a deeper analysis could be found in [16]), which in our case, according to linguistic terms defined in Table 1, represents *very low OR low OR medium OR high OR very high*. Consequently, the proposition will be interpreted as:

Commanded_linear_velocity is
 (very low OR low OR medium OR high OR very high)

The semantics of the linguistic terms considered in these rules, as well as the method to define them, will be described in the following section.

Another example of expert rule can be produced for the collision and drag case, introducing the variable *Decrease_of_battery_voltage* previously described. To do so, raw data has to be processed to be put in the form of the variable used by the expert. For example, the velocity signal shown in Fig. 4, is processed through Kalman or FIR (Finite-input response) filtering in order to define a clearer *velocity undershoot* produced by the collision.

Fig. 4 Vehicle velocity (instantaneous, Kalman and FIR) upon obstacle collision



Considering this new variable, an expert rule is that described below:

IF *Undershoot_width* is (small OR medium) **AND**
Undershoot_depth is (very small OR small OR medium OR medium large OR large) **AND**
Decrease_of_battery_voltage is small
THEN *Vehicle_draggs_obstacle*

The expert knowledge extraction process yields a qualitative (and potentially incomplete) rule base that in the case of this application is made up of the two rules previously described.

3.2.3 Knowledge Base Quality

This subsection explains the criteria for evaluating knowledge base quality. It is measured according to data through considering the following indices for the output variable:

- *Error cases*: Number of covered cases from data set that produce error, i.e. observed and inferred values are different, in inference.
- *Ambiguity cases*: Number of covered cases from data set that produce ambiguity, i.e. difference between the two highest confidence levels is smaller than an established threshold.
- *Unclassified cases*: Number of cases from data set which do not fire at least one rule with a degree higher than Δ .

These indices convey complementary information. A good knowledge base should minimize them by offering an accurate (reducing error cases), consistent (reducing ambiguity cases) and complete (reducing unclassified cases) set of rules.

The application of those indices to the set of two rules provided by the expert is described in the following section by considering a data set that will also be used for knowledge induction.

4 Induced Knowledge Integration

Complementary to expert knowledge, data are likely to give a good image of variable interaction. So starting from the knowledge described by the expert, some additional information is required in order to be able to solve the problem considered.

The process to get that additional information implies several steps. First, data is required, so experimental or simulated data should be provided. Second, additional rules will be generated on the basis of the provided data, and the linguistic variables previously defined by the expert. At a third stage, expert and induced rules will be integrated to check the whole knowledge base. Finally, the whole knowledge base, data base and rule base, will be simplified in order to achieve a more compact knowledge base, with a smaller size in order to increase interpretability [4] while preserving accuracy.

4.1 Generating Data for Knowledge Induction

Some real experiments have been performed so as to collect data concerning all the variables of interest, and especially the vehicle battery voltage and linear velocity. All practical experiments were conducted in an absolutely real environment, which is the Department of Electronics at the University of Alcalá, in Madrid. The test environment has a surface of 60x60 square metres.

Thus, in a first trial, a small but heavy obstacle was deliberately introduced in the environment in order to interrupt the vehicle trajectory during autonomous operation. Due to its small size, the obstacle cannot be detected by the ultrasound-based obstacle detection module onboard the vehicle. Accordingly, the vehicle collides with the obstacle, yielding a temporary decrease in its linear velocity. Upon collision, the velocity controller adapts to this situation by increasing the actuator torque so as to rapidly reach the commanded reference velocity. This causes the vehicle to drag the obstacle along its way by increasing the battery current consumption, and consequently, the battery voltage goes down slightly.

For generalization purposes, Table 2 shows a complete list of all the different obstacle weights used in the experiments, as well as the vehicle velocity at each test. A total of 90 different tests were carried out. All obstacles above 0.5 kg produce a sudden decrease in robot velocity that is compensated for by the velocity controller after a few milliseconds. The effect can then be considered as a minor disturbance, although the problem is that energy consumption remarkably increases while dragging an obstacle.

Let us concentrate on the simple example depicted in Fig. 5 to illustrate the knowledge extraction process, including its cooperation with expert knowledge. The vehicle battery voltage and linear velocity are depicted for a real collision-and-drag

Table 2 Experimental setup

No. of experiments	Velocity (mm/s)	Weight (Kg)
6	50	0.5
6	100	0.5
6	150	0.5
6	200	0.5
6	250	0.5
6	50	1.0
6	100	1.0
6	150	1.0
6	200	1.0
6	250	1.0
6	50	2.0
6	100	2.0
6	150	2.0
6	200	2.0
6	250	2.0

case, for two different commanded linear velocities. Although in both cases there are two substantial drops in the velocity, only the first ones (those inside the circles) correspond to the collision-and-drag situation and they are correctly classified. The undershoot in the left figure has a depth of 25% of velocity, i.e. 25 mm/s, while the one on the right has a depth of 44%, i.e. 11 mm/s. These two values, according to the partitions that were generated in the Section 3.2.1, can be considered as small and medium undershoots, respectively. The rest of drops in the velocity are considered as normal situations.

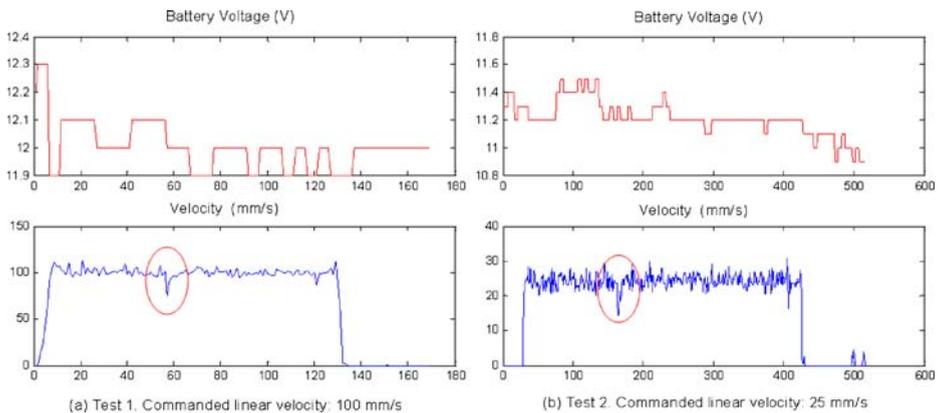


Fig. 5 Vehicle battery voltage and linear velocity during a collision-and-drag case

Table 3 Results based only in expert knowledge

Total cases	Error cases (%)	Ambiguity cases (%)	Unclassified cases (%)	Diagnosis
89	2.25	0	70.79	Vehicle_drags_obstacle
34	0	0	2.94	Vehicle_stalled
44	2.27	0	97.73	Normal

One hundred sixty seven instances (the 90 cases shown in Table 2 were obtained in real trials, and 77 more ones were generated in software simulations⁴), corresponding to various configurations (according to changes in the vehicle velocity and in the obstacle weight), have been collected, and the set has been randomly divided into 2 different subsets (84 and 83 examples), maintaining the ratio of each diagnosis in both data sets. The number of cases for each possible diagnosis is detailed in Table 3.

How are these samples managed using the two expert rules? As expected, the expert knowledge base is incomplete: It covers only 35% of the cases.

Table 3 shows clearly that only the case of *Vehicle_stalled* diagnosis is handled by the expert rules. The diagnosis of *Vehicle_drags_obstacle* seems to be more difficult to define, only 26.96% of the cases over 89 are correctly classified. Finally, as no rule has been defined for *Normal* diagnosis, all the corresponding examples are unmanaged, except one which is misclassified.

As previously stated, the expert is only able to express a partial view of the problem. Data may also be used to provide new knowledge elements.

4.2 Rule Induction

Once the variable partitions have been designed, they can be used to define fuzzy rules. To the two kinds of knowledge, expert knowledge and data, correspond two kinds of rules. As both types of rules use the same linguistic labels defined by the same fuzzy sets, rule comparison can be achieved by considering the linguistic level only.

From the expert side, as previously described, rules are merely written in a linguistic way using the available connectives between variables.

The process of generating rules from data is called induction. It aims to produce general statements, expressed as fuzzy rules in our case, valid for the whole set, from partial observations. The observed output, for each sample item, is part of the training set allowing supervised training procedures.

A lot of methods are available in the fuzzy logic literature [14, 18]. We choose a prototype oriented strategy according to the classification proposed in [3]. It starts from examples. The alternative strategy is called input space strategy, as it examines all the fuzzy regions.

⁴Five simulations were made for each configuration in Table 2, and two more simulations were made in the case of 50 mm/s and 0.5 kg, because this case was the more critical due to the huge level of noise.

The used method is adapted from the Wang and Mendel (WM) algorithm [31]. Given the fuzzy partitions, it starts by generating one rule for each data pair of the training set.

The i^{th} pair one is written:

IF x_1 is A_1^i **AND** x_2 is A_2^i ... **AND** x_p is A_p^i **THEN** y is C^i .

The fuzzy sets A_j^i are those for which the matching degree of x_j^i is maximum for each input variable j from pair i . The fuzzy set C_i is the one for which the matching degree of the observed output, y_i , is maximum.

A degree is assigned to each rule. For a given rule it is equal to the rule firestrength for the considered pair. In case of identical premises for two rules, only the one with the higher degree is kept.

This procedure is very easy to use as it does not require any parameter. Moreover, it allows the rule base to be adaptive: new rules may compete with existing ones.

Rule induction may be carried out from the whole training set, but this way some induced rules are likely to be redundant with expert rules. An alternative process consists of removing from the training set the samples managed by the expert knowledge base. In this paper both ways are studied and compared. An item is considered as managed by an expert rule if its firestrength is higher than a threshold (in this case we have considered 0.3, a value obtained empirically).

As the induced rules are defined using all the available variables, they are likely to be simplified.

4.3 Integration and Simplification Process

Induced and expert rules have to be integrated into a unique rule base. As explained previously, rule comparison can be achieved by considering the linguistic level only, because both types of rules use the same linguistic labels defined by the same fuzzy sets. The merging of two rule bases may be a source of conflict. The integration process preserves the new rule base from two important defects: Inconsistency and redundancy.

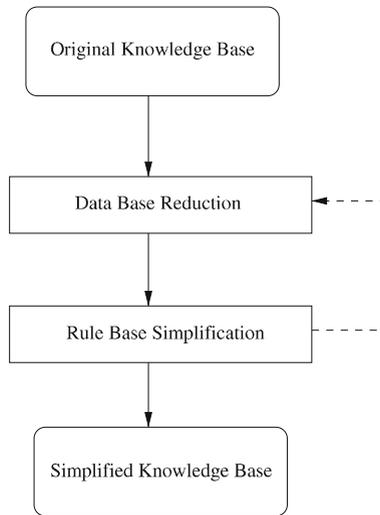
A simplification procedure is also carried out within the integration phase. It aims to reduce the number of rules in the rule base and the number of variables that are considered in each rule. Figure 6 shows a schematic diagram of knowledge base simplification process. It is a cyclical process. Depending on the original knowledge base, the whole process may involve several iterations as data base reduction affects to rule base simplification and vice versa.

4.3.1 Data Base Reduction

Data base is related to variable definitions, i.e. qualitative and quantitative information about variable behaviour. It includes partition definition for each input or output variable, as well as the semantic meaning, linguistic term, attached to each fuzzy set.

The strong fuzzy partitions are made up of elementary fuzzy sets. They can be combined to define composite OR and NOT linguistic terms. The implementation of OR and NOT operators is detailed in [16]. The OR composite labels are defined as

Fig. 6 Knowledge base simplification



the convex hull of the combined terms. And the NOT composite labels are defined according to Eq. 13, generating two OR composite linguistic terms.

$$NOT(MF_i) \Rightarrow \begin{cases} MF_1 \text{ OR } \dots \text{ OR } MF_{i-1} \\ MF_{i+1} \text{ OR } \dots \text{ OR } MF_m \end{cases} \quad (13)$$

This reduction process includes the next steps:

1. Look for labels which are used by none of the rules and propose to remove them.
2. Look for labels which are always used together and propose to merge them into a new one.

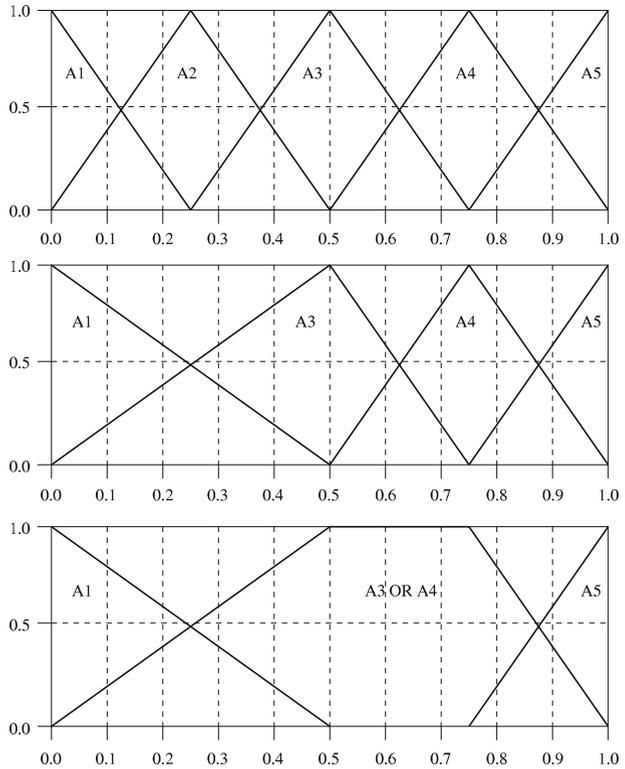
Removing or merging labels affects the elementary fuzzy sets of the partition.

If one label is used by none of the rules, then it can be removed. However, in order to keep a strong fuzzy partition, adjacent fuzzy sets are expanded. The right boundary of the prior fuzzy set and the left boundary of the subsequent one are moved up to the centre of the other.

Labels which are always used together can be grouped into a unique one. The procedure also stands for OR combinations: When a OR composite label is used but its elementary components aren't used neither alone nor within another composite label, then the elementary labels can be merged at the partition level. The number of labels is decreased and a new label with trapezoidal or semi trapezoidal shape substitutes to the old component labels.

For instance, Fig. 7 illustrates how a strong fuzzy partition with five linguistic terms (picture on the top) is altered through two consecutive modifications: First, in the middle part A_2 has been deleted, then A_3 and A_4 have been merged in the bottom part of the figure. Let us note that final partition is still a strong fuzzy partition.

Fig. 7 Removing and merging labels within a strong fuzzy partition



4.3.2 Rule Base Simplification

Rule premises are made up of tuples (*input variable, linguistic term*), where the absence of an input variable in a rule means that the variable is not considered in the evaluation of the rule. On the other hand, rule conclusions are made up of tuples (*output variable, linguistic term*), where the absence of an output variable means that this rule does not concern that output variable. For each input variable, the user can choose between basic labels or composite labels (by using OR/NOT operators).

As rules can be either defined by an expert or induced from data. Rule nature is taken account into the simplification process: In case of conflict priority is given to expert rules.

The overall reduction process involves two steps.

First, the *Simplify RB* procedure is applied in order to remove redundant rules. Redundancy can be total, i.e. identical rules, or partial: The input space covered by one rule is included into the one covered by the other, and both rules have the same conclusion. The most specific rule is removed, but only if it is an induced rule, or it is an expert rule and the most general rule is an expert rule too.

Second, the *Merge RB* procedure is used for building more compact and general rules. A linguistic analysis of the rule base is made in order to detect rules that can be merged. Two rules can be merged if they are of the same nature (expert or

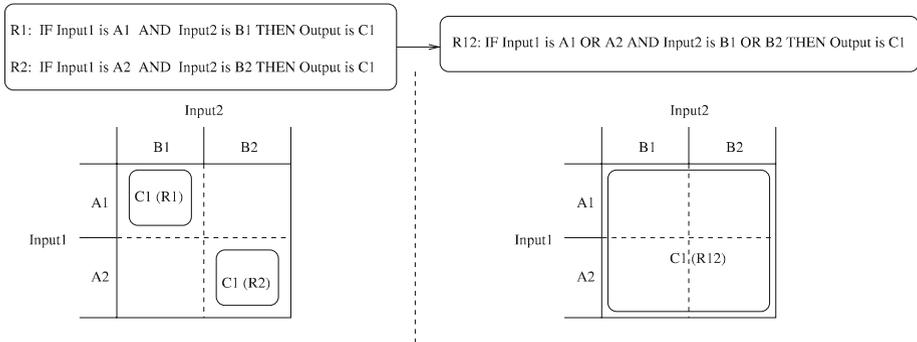


Fig. 8 Rule generalization

induced rules) and also satisfy next condition: They have the same conclusion and their premises can be merged using composite linguistic terms.

The procedure for merging rules follows next sequence of steps, which are repeated until the whole rule base is analyzed:

1. Evaluate quality of knowledge base.
2. Look for two rules which can be merged.
3. Generate a temporal copy of the knowledge base.
4. Substitute both rules by the merged rule in the temporal knowledge base.
5. Evaluate quality of temporal knowledge base.
6. If (new quality isn't worse than old quality) Then (changes in knowledge base are saved) Else (changes are discarded).
7. Go to 2.

In order to merge two rules, it is needed to analyze the premise part of the rules. There is no problem with the conclusion because it is the same in the merger rule and in the original rules. However, for each input the premises of original rules have to be compared. If both premises are the same, then this is the merger rule premise. But, if premises are different, then a new premise has to be built as merger of both.

The simplification process produces a generalization, a simplified rule may cover an input space which was not initially managed, for instance because there are no available training data in this zone. Figure 8 shows how the merger rule R12 covers a larger input space than the one covered by R1 and R2.

Merging rules changes rule base configuration, but it does not modify the fuzzy partitions in data base. Nevertheless, as a result of *Merge RB* process, composite labels could appear in the premise part of the rules. The fuzzy partition modification is made in the Data Base Reduction process as explained in Section 4.3.1.

4.3.3 Improving Integration

The integration of expert and induced knowledge has been designed with the aim of completing, through a data driven process, the knowledge provided by the expert. In that sense, the induction process should concentrate on those areas of the problem that were not appropriately described by the expert. As a consequence, it seems

Table 4 Induced rules quality over training set (84 cases)

Method	Rules	Error cases (%)	Ambiguity cases (%)	Unclassified cases (%)
1 Expert	2	3.57	0	59.53
2 WM	70	1.19	0	0
3 WM + Simplification	9	1.19	0	0
4 Expert + WM	72	1.19	2.38	0
5 Expert + WM + Simplification	12	1.19	1.19	0
6 Expert + Data selection + WM	47	1.19	4.76	0
7 Expert + Data selection + WM + Simplification	9	1.19	4.76	0

to be interesting to reduce the training set, by removing those cases that represent situations clearly described by the expert knowledge, before the induction process.

The first aim of this reduction of the training data set is focused on reducing the required effort for both the induction and the simplification stages of the process, but as will be shown by the results, there is a side effect that at the end is the most important one. The data selection process improves the quality of the obtained knowledge base both in interpretability and accuracy. Interpretability is improved through the reduction of the rule base, while the number of correctly classified cases is larger (accuracy improvement).

4.4 Results

Tables 4 and 5 summarize the main results for the training (84 cases) and test (83 cases) sets respectively.

The first line illustrates the performance of the expert rule base. As we know, only a third part of the samples are managed.

Lines 2 and 3 give the results gained using the induced rule base, without any cooperation with expert rules. These results highlight the poor level of generalization of the induced rules. The number of uncovered samples rises up to 45.78% for the test set. The simplification process helps in building more general rules: While the number of rules decreases from 70 to 9, the number of unmanaged samples drops to 7.23%. We can argue that the simplification process somehow produces a generalization.

Table 5 Induced rules quality over test set (83 cases)

Method	Rules	Error cases (%)	Ambiguity cases (%)	Unclassified cases (%)
1 Expert	2	0	0	68.67
2 WM	70	0	0	45.78
3 WM + Simplification	9	2.4	0	7.23
4 Expert + WM	72	0	0	32.53
5 Expert + WM + Simplification	12	3.61	0	8.43
6 Expert + Data selection + WM	47	0	2.4	24.1
7 Expert + Data selection + WM + Simplification	9	1.2	3.61	0

Table 6 Input Variables after Simplification

Variable	Labels	Linguistic terms
Undershoot_width	6	null, very small, small, medium, medium large OR large, very large
Undershoot_depth	4	null, very small, small OR medium OR medium large OR large, very large
Decrease_of_battery_voltage	5	zero, very small, small, medium, large
Range_sonar	2	zero, NOT(zero)
Derivative_of_range_sonar	2	zero, NOT(zero)
Commanded_linear_velocity	4	zero, very low, low, medium OR high OR very high
Measured_linear_velocity	4	zero, very low, low OR medium OR high, very high

The first way of cooperation, lines 4 and 5, deals with the expert rules and the rules induced from the whole training set. The resulting rule base is likely to be redundant, even inconsistent, as illustrated by the presence of ambiguous classification cases. The remarks already made about induced rule generalization hold in this way of cooperation. The number of unclassified items drops from 32.53 to 8.43% when the number of rules becomes 12 (72 in the initial base).

The last lines investigate a new way of cooperation: Examples covered by expert rules are removed from the training set before induction. The removed samples are 16 of the stalled case (over 17) and 12 (over 45) of the drag case. The induction process yields a smaller number of rules: 45 instead of 70. The generalization ability of complete induced rules is still poor: 24.1% of items in the test set remain unclassified. After simplification, with 7 induced rules and 2 expert rules, all the samples are managed by the system. The final result gives one error case (1.2%) and three ambiguous cases (3.61%) dealing with the test set. This last simplified rule base uses the linguistic terms given in Table 6. The nine final rules are available in the [Appendix](#).

The analysis of the results provides some information about the knowledge extraction approach that has been considered. The first idea is that the integrated knowledge base with expert and induced knowledge outperforms the results of both, the expert and the induced knowledge base. That point seems to be the obvious result of adding rules with the subsequent effect of improving accuracy.

But there are two other less obvious ideas since they represent the fact that in some cases the reduction of the rule base (improving interpretability) produces an accuracy improvement. In that sense, the second situation to be considered is the fact that in all the three options for generating the knowledge base (induction, expert plus induction, and expert plus induction after data selection), the effect of simplification is a simultaneous improvement of interpretability and accuracy. It is really impressive how simplification process can lead to 24.1% more of the unclassified cases being classified (see lines 6 and 7 in Table 5). The knowledge gain is due to the simplification process affects to the whole knowledge base (fuzzy rules, variables, and membership functions). It produces a generalization at both levels, variables and rules. As a result, the simplified knowledge base may cover an input space which was not initially managed, for instance because there are no available

training data in this zone. Finally, the third conclusion is that, in the two cases of integration of expert and induced knowledge that have been considered (with and without simplification), the data selection process has produced an improvement of both interpretability and accuracy.

Finally, and what seems to be the most interesting question from the point of view of learning, the results obtained with simultaneous data selection and simplification show a similar performance on training and test sets (even considering the reduced size of the training data set), i.e., the generalization ability of the approach is quite good.

5 Conclusions

Some ground robot motion problems are considered in this paper and especially the detection, using robot motion parameters, of non visible obstacles using the usual onboard sensorial capacities of the robot. This detection is of prime importance for autonomous operation of ground robots in real environments. The designed system aims to provide diagnosis as well as recovery actions in these circumstances. Although the diagnosis system developed in this work is ideal to be applied indoors, it can also be used in outdoor environments whenever the terrain is not uneven.

As the obstacle characteristics are, obviously, unknown, the global system, i.e. robot and obstacle, cannot be accurately modelled from a quantitative point of view and only qualitative (or approximate) reasoning can be applied. As demonstrated throughout the paper, some linguistic relationships can be established between the obstacle characteristics and their influence in the robot motion variables upon collision. Fuzzy logic and fuzzy inference systems are used to model those relationships. To complete expert knowledge, experiments have been carried out to characterize typical behaviours. Resultant data were used for building new induced fuzzy rules.

This paper illustrates that the two kinds of knowledge, expert knowledge and data, convey complementary information. Fuzzy inference systems offer a common representation framework, as these two types of knowledge can be modelled using fuzzy linguistic rules. The cooperation of expert knowledge and data in system modelling remains an open problem, especially when the goal is to obtain an accurate and interpretable system. This work proposes a cooperation framework, focused on rule base integration and simplification under expert control. The results show that the proposed approach leads not only to a good balance between accuracy and interpretability but also to a simultaneous improvement of both. The final knowledge base is more compact and transparent, but also more accurate than the initial one.

All results presented in this paper were reached using KBCT[2], a free software tool (distributed under the terms of the GNU General Public License) for generating or refining fuzzy knowledge bases.

Acknowledgements We would like to acknowledge the partners of the ADVOCATE II project. The consortium consists of the following partners: University of Alcalá, GETRONICS DECAN, Universidad Politécnica de Madrid, STN-ATLAS Elektronik GmbH, Ifremer, HUGIN Expert A/S, INNOVA S.p.A, and E-MOTIVE. The European Commission (IST-2001-34508) supports this work.

Appendix

Expert Rules

IF *Undershoot_depth* is very large **AND**
Range_sonar is not(zero) **AND**
Derivative_of_range_sonar is zero **AND**
Commanded_linear_velocity is not(zero) **AND**
Measured_linear_velocity is zero

THEN *Vehicle_stalled*

IF *Undershoot_width* is (small OR medium) **AND**
Undershoot_depth is (very small OR small OR medium OR
medium large OR large) **AND**
Decrease_of_battery_voltage is small

THEN *Vehicle_drags_obstacle*

Induced Rules after Simplification

IF *Undershoot_width* is NOT(null) **AND**
Undershoot_depth is NOT(null) **AND**
Range_sonar is NOT(zero) **AND**
Commanded_linear_velocity is NOT(zero) **AND**
Measured_linear_velocity is (very low OR low OR medium OR high)

THEN *Vehicle_drags_obstacle*

IF *Undershoot_width* is NOT(null) **AND**
Decrease_of_battery_voltage is NOT(large) **AND**
Range_sonar is NOT(zero) **AND**
Commanded_linear_velocity is (low OR medium OR high OR very high) **AND**
Measured_linear_velocity is (low OR medium OR high OR very high)

THEN *Vehicle_drags_obstacle*

IF *Undershoot_width* is very large **AND**
Undershoot_depth is NOT(null) **AND**
Decrease_of_battery_voltage is large **AND**
Range_sonar is NOT(zero) **AND**
Derivative_of_range_sonar is zero **AND**
Commanded_linear_velocity is very low **AND**
Measured_linear_velocity is zero

THEN *Vehicle_drags_obstacle*

IF *Undershoot_width* is NOT(null) **AND**
Undershoot_depth is very large **AND**
Decrease_of_battery_voltage is very small **AND**
Range_sonar is NOT(zero) **AND**
Derivative_of_range_sonar is zero **AND**
Commanded_linear_velocity is very low **AND**
Measured_linear_velocity is zero

THEN *Vehicle_drags_obstacle*

IF	<i>Undershoot_width</i>	is <i>very large</i>	AND
	<i>Undershoot_depth</i>	is <i>very large</i>	AND
	<i>Decrease_of_battery_voltage</i>	is <i>large</i>	AND
	<i>Range_sonar</i>	is <i>NOT(zero)</i>	AND
	<i>Derivative_of_range_sonar</i>	is <i>NOT(zero)</i>	AND
	<i>Commanded_linear_velocity</i>	is <i>low</i>	AND
	<i>Measured_linear_velocity</i>	is <i>zero</i>	
THEN	<i>Vehicle_stalled</i>		
IF	<i>Undershoot_width</i>	is <i>NOT(null)</i>	AND
	<i>Undershoot_depth</i>	is (<i>null OR very small</i>)	AND
	<i>Decrease_of_battery_voltage</i>	is (<i>small OR medium OR large</i>)	AND
	<i>Range_sonar</i>	is <i>NOT(zero)</i>	AND
	<i>Commanded_linear_velocity</i>	is (<i>zero OR very low</i>)	AND
	<i>Measured_linear_velocity</i>	is <i>very low</i>	
THEN	<i>Normal</i>		
IF	<i>Undershoot_width</i>	is <i>NOT(null)</i>	AND
	<i>Undershoot_depth</i>	is <i>null</i>	AND
	<i>Decrease_of_battery_voltage</i>	is (<i>zero OR very small OR small</i>)	AND
	<i>Range_sonar</i>	is <i>NOT(zero)</i>	AND
	<i>Commanded_linear_velocity</i>	is <i>very low</i>	AND
	<i>Measured_linear_velocity</i>	is <i>very low</i>	
THEN	<i>Normal</i>		

References

1. Aguirre, E., González, A: Fuzzy behaviors for mobile robot navigation: design, coordination and fusion. *Int. J. Approx. Reason.* **25**(3), 255–289 (2000 November)
2. Alonso, J.M., Guillaume, S., Magdalena, L.: Kbct: A knowledge management tool for fuzzy inference systems. <http://www.mat.upm.es/projects/advocate/kbct.htm>. (2003)
3. Carmona, P., Castro, J.L., Zurita, J.M.: Strategies to identify fuzzy rules directly from certainty degrees: a comparison and a proposal. *IEEE Trans. Fuzzy Syst.* **12**(5), 631–640 (2004 October)
4. Casillas, J., Cordon, O., Herrera, F., Magdalena, L.: Interpretability Issues in Fuzzy Modeling. volume 128 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg New York (2003)
5. de Oliveira, J.V.: Semantic constraints for membership functions optimization. *IEEE Trans. Syst. Man Cybernet., Part A* **29**(1), 128–138 (1999)
6. Demirli, K., Türksen, I.B.: Sonar based mobile robot localization by using fuzzy triangulation. *Robot. Autonom. Syst.* **2–3**(33), 109–123 (2000 November)
7. Duan, Z.-H., Cai, Z.-X., Yu, J.-X.: Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: a survey. In: *IEEE International Conference on Robotics and Automation*. Barcelona, Spain, (2005 April)
8. Espinosa, J., Vandewalle, J.: Constructing fuzzy models with linguistic integrity from numerical data-afrel algorithm. *IEEE Trans. Fuzzy Syst.* **8**(5), 591–600 (2000 October)
9. Fraichard, T., Garnier, P.: Fuzzy control to drive car-like vehicles. *Robot. Autonom. Syst.* **34**(1), 1–22 (2001 January)
10. Fukuda, T., Kubota, N.: An intelligent robotic system based on a fuzzy approach. *Proc. IEEE* **87**(9), 1448–1470 (1999 September)
11. Garibaldi, J.M., Musikasuwana, S., Ozen, T., John, R.I.: A case study to illustrate the use of non-convex membership functions for linguistic terms. In: *FUZZ-IEEE 2004*, vol. 3, pp. 1403–1408, Budapest, Hungary, (July 25-29, 2004)

12. Glorrenec, P.-Y.: *Algorithmes d'apprentissage pour systèmes d'inférence floue*. Editions Hermès, Paris, (1999)
13. Gómez-Bravo, F., Cuesta, F., Ollero, A.: Parallel and diagonal parking in nonholonomic autonomous vehicles. *Eng. Appl. Artif. Intel.* **14**(4), 419–434 (2001 August)
14. Guillaume, S.: Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Trans. Fuzzy Syst.* **9**(3), 426–443 (2001 June)
15. Guillaume, S., Charnomordic, B.: Generating an interpretable family of fuzzy partitions. *IEEE Trans. Fuzzy Syst.* **12**(3), 324–335 (2004 June)
16. Guillaume, S., Magdalena, L.: An or and not implementation that improves linguistic rule interpretability. In: Eleventh International Fuzzy Systems Association World Congress, vol. I, pp. 88–92, Beijing, China, (2005 July)
17. Guillaume, S., Magdalena, L.: Expert guided integration of induced knowledge into a fuzzy knowledge base. *Soft Comput. - A Fusion Foundations Methodol. Appl.* **10**(9), 773–784 (2006 July)
18. Hüllermeier, E.: Fuzzy methods in machine learning and data mining: status and prospects. *Fuzzy Set Syst* **156**, 387–406 (2005)
19. Isermann, R.: On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Trans. Syst. Man Cybernet. Part A Syst. Human.* **28**(2), 221–235 (1998 March)
20. Jetto, L., Longhi, S., Vitali, D.: Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted kalman filter. *Control Eng. Prac.* **7**(6), 763–771 (1999 June)
21. Joentgen, A., Mikenina, L., Weber, R., Zeugner, A., Zimmermann, H.J.: Automatic fault detection in gearboxes by dynamic fuzzy data analysis. *Fuzzy Set Syst.* **105**(1), 123–132 (1999)
22. Kodagoda, K., Wijesoma, W., Teoh, E.: Fuzzy speed and steering control of an agv. *IEEE Trans. Control Syst. Tech.* **10**(1), 112–120 (2002 January)
23. Maaref, H., Barret, C.: Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment. *Control Eng. Prac.* **8**(7), 757–768 (2000 July)
24. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Stud.* **7**, 1–13 (1975)
25. Mohan, N., Undeland, T.M., Robbins, W.P.: *Power Electronics: Converters, Applications, and Design*. John Wiley & Sons, Inc., (1989)
26. Ruspini, E.H.: A new approach to clustering. *Inf. Control* **15**, 22–32 (1969)
27. Soliman, A., Rizzoni, G., Kim, Y.W.: Diagnosis of an automotive emission control system using fuzzy inference. *Control Eng. Prac.* **7**(2), 209–216 (1999 February)
28. Sotelo, M.A., Bergasa, L.M., Flores, R., Ocaña, M., Doussin, M.-H., Magdalena, L., Hornfeld, W., Perrier, M., Madsen, A.L., Furlani, A., Roland, D.: ADVOCATE II: ADVanced On-board diagnosis and Control of Autonomous sysTEms II. In: Ninth International Conference on Computer Aided Systems Theory. Las Palmas de Gran Canaria, Spain, (2003 February)
29. Steinberg, A.N., Bowman, C.L., White, F.E.: Revisions to the jdl data fusion model. *Sensor Fusion: Architectures, Algorithms, and Applications, Proceedings of the SPIE 3719* (1999)
30. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. System Man Cybernet.* **15**, 116–132 (1985)
31. Wang, L.-X., Mendel, J.M.: Fuzzy basis functions, universal approximation, and orthogonal least squares learning. *IEEE Trans. Neural Net.* **3**, 807–814 (1992)
32. White, F.E.: A model for data fusion. In: First National Symposium on Sensor Fusion, (1988)
33. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**, 338–353 (1965)