Third Conference of the European Society for Fuzzy Logic and Technology

EUSFLAT 2003

PROCEEDINGS

September 10 - 12, 2003

Zittau, Germany

Publishers: European Society for Fuzzy Logic and Technology (EUSFLAT)

University of Applied Sciences at Zittau/Goerlitz
Theodor-Koerner-Allee 16
02763 Zittau, Germany

Edited by: Dr.rer.nat.habil. Michael Wagenknecht
Prof. Dr.-Ing.habil. Rainer Hampel

Press deadline: August 15, 2003

Nominal Charge: 30.- €

Printed in Germany

ISBN: 3-9808089-4-7

ACKOWLEDGEMENTS

The organizers of the 3rd EUSFLAT Conference would like to thank all institutions and people without them this conference could not be held. Special thanks are due to the following firms and organizations for their help, financial support and sponsorship:

- Deutsche Forschungsgemeinschaft (DFG)
- Deutscher Akademischer Austauschdienst (DAAD)
- Vattenfall Europe Generation AG & Co. KG
- Springer-Verlag
- Shaker Verlag GmbH
- Vieweg & Sohn Verlagsgesellschaft mbH
- R. Oldenbourg Verlag
- RWE NUKEM Gruppe
- IPM staff
- Saechsisches Ministerium für Wissenschaft und Kunst

"Fuzzy Scheduling"

"Fuzzy Logics"

"Fuzziness and Probability"

Expert and induced knowledge for Intelligent motion analysis of ground robots

Serge Guillaume, Luis Magdalena

Dept. Applied Mathematics
Universidad Politécnica de Madrid
Madrid 28040
Spain
llayos@mat.upm.es

Miguel Ángel Sotelo, Luis Miguel Bergasa, Ramón Flores, Manuel Ocaña

Department of Electronics.
University of Alcalá
Alcalá de Henares 28871
Spain
michael@depeca.uah.es

Abstract

The paper describes the knowledge extraction process in the application of a fuzzy system, using expert and induced knowledge, to the detection of motion problems in ground robots.

Keywords: Robot, Expert, Induced, Knowledge, Fuzzy

Introduction

The application described in this paper is developed in the framework of ADVOCATE II project [1].

The overall objectives of that project are:

- To construct an open, modular, and generic software architecture for autonomous robotic systems diagnosis and control.
- To develop or improve a set of intelligent diagnosis modules fully compatible with this architecture and tested in operational applications.
- To carry out practical tests and demonstrations on a set of operational prototypes in order to prove operationality and efficiency of this solution in several application fields, and particularly for Autonomous Underwater Vehicles (AUVs) and Autonomous Ground Vehicles (AGVs).

In what concerns this paper, ADVOCATE II defines an architecture build for the purpose of combining different intelligent techniques for diagnosis, recovery and re-planning into autonomous vehicles. The global objective of the architecture is to enhance the level of reliability and efficiency of an autonomous robotic system.

Three different vehicles are considered in the project: two underwater and one ground robots. For each one of these robots different failures have been analysed. In this paper we will concentrate on the ground robot BART (Basic Agent for Robotic Tasks), see Figure 1.

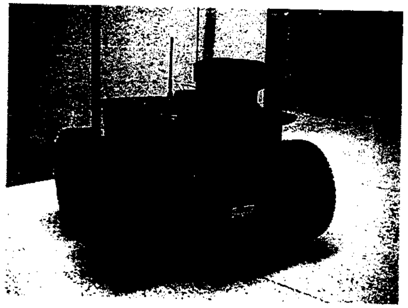


Figure 1. Basic agent for robotic tasks (BART)

BART is a ground vehicle commanded in teleoperated mode (a Remotely Operated Vehicle, ROV, indeed) 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.

The operator is in charge of global vehicle navigation by remotely commanding its actuators according to the images that are continuously transmitted through a wireless ethernet link from the vehicle to the base station. Information concerning proximity sensors (the vehicle is equipped with a ring of ultrasound sensors) is also transmitted for monitoring. The operator guides the vehicle to key positions throughout the building, such as corridor entrances. Once located at those points, the vehicle starts an autonomous local navigation manoeuvre to carry out the reconnaissance and surveillance of the corridor, in a narrow environment where complete teleoperation is complex to realize. The manoeuvre includes navigation through the corridor (in

and out) and detection of open doors for inspection. The Vehicle Piloting Module can also override the operator actuation in teleoperated mode for safety reasons so as to avoid collisions.

At any time, the operator can interrupt the mission and return the vehicle to the base station by regaining complete teleoperation command. An emergency stop command can also be issued upon severe failure conditions so as to preserve vehicle safety. During the execution of the mission, several problems or failures can occur yielding the vehicle to abnormal behaviour with respect to the expected one, and thus, impeding proper finalisation. ADVOCATE II will be used to do diagnosis and/or recovery actions for those failures using the modular and intelligent diagnosis system developed in this project. These are provided in the following sections.

Different problems have been considered in the case of BART: AGV trapped, limited capability of motion, Actuator blocked or tangled, Motor stopped, Energy too low for complete mission.

Problem description

We will now concentrate on the motion problems, that include the situations where the AGV is trapped, has a limited capability of motion, or the actuators are blocked or tangled. These problems are related to different situations as the collision with an obstacle (usually not detected by sensory means) that can even be dragged, or the entanglement of vehicle wheels.

On these situations, the diagnosis system will work on the basis of several symptoms that can be considered to determine the problem. These symptoms are mainly related to a deviation of the motion from the expected motion according to model, a second symptom could be related to a divergence in between the commanded and the obtained revolutions of the engine, and finally, the deviation in consumption from what should be produced.

The diagnosis module will try to determine these motion problems according to the described symptoms. Once defined the problem, a recovery action will be defined to produce a special manoeuvre to get rid of obstacle, and subsequently to resume the mission.

To analyse the problem, several experiments have been done with different kinds of obstacles (and different configurations of the obstacle-robot situation) to be able to generate an overall view of the problem. The objective of the experiments is twofold: to create experience (then producing expert rules to define the knowledge about the problem) and to generate data to be used for additional knowledge induction.

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 behavior. From our experience, expert reasoning uses linguistic terms and is based on prototypes.

We choose the fuzzy logic formalism [2] for its well known linguistic concept modeling ability. The fuzzy rule expression is close to expert natural language. On the other hand, as they are universal approximators, fuzzy inference systems can be used for knowledge induction processes. The cooperation between expert knowledge and induced knowledge highly depends on induced rule interpretability and will be considered in the section devoted to Experimentations.

As the objective is not the design of a complete expert system, the process of expert knowledge extraction can be kept at a 'high' abstraction level. The process can be limited at 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 first step is then to define the number and nature of variables that are involved in the diagnosis process according to the domain expert experience. Considering the problem of detecting abnormal dynamics due to obstacles dragging or even stalling, as will be described in the following section, the next variables are proposed after appropriate preprocessing provided by the vehicle piloting module.

- Difference_of_battery_voltage. It provides a differencial measurement of the decrease suffered by the battery voltage when colliding against an obstacle. This decrement is directly linked to the vehicle consumption, that should increase upon collision, subsequently producing the battery voltage to go slightly down (see figure 2).
- Linear and angular velocities. Both commanded and measured velocities are included.
  - measured_linear_velocity.
  - commanded_linear_velocity.
  - measured_angular_velocity.
  - commanded_angular_velocity.

As will be graphically demonstrated in the next section (figure 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 provide a diagnosis on it.

- Depth and width of velocity undershoot (undershoot_depth and undershoot_width) produced upon collision. These variables are crucial 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. The depth and width of the velocity undershoot are tightly related to the commanded vehicle velocity as deep peaks occur at low velocity while small ones take place at high speed, as can be seen in figure 2.
- Range_measurements. A ring of 16 ultrasound based sensors is used to provide range measurements around the robot.
- Derivative_of_range_measurements. The derivative of range measurements is useful to provide information concerning robot movement with respect to its environment as described below.

The vehicle piloting module provides a coarse alarm whenever there is a suspiction that a collision has occurred and it has not been detected by the ultrasound ring. Upon receiving the alarm, the objective of the fuzzy logic module is to provide some of the following diagnosis based on the previous variables.

- Normal: it means that a false alarm was launched by the vehicle piloting module, 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 commanded velocity and keeps on moving by dragging the obstacle on its way.
- Vehicle_stalled: in this case the obstacle is heavy enough to avoid the vehicle from moving. Upon these circumstances, there are two different possibilities: on one hand, the velocity controller pushes the actuators until the vehicle wheels start to slip without producing any vehicle movement. On the other hand, the vehicle wheels could get trapped by the obstacle. In the latter case no slippery movement would take place yielding to a situation denoted as actuator stall. Within the ADVOCATE architecture there is already an intelligent module devoted to provide diagnosis on actuator stall, and thus, the fuzzy logic module will only take care of providing diagnosis on vehicle stall on slippery circumstances.
- Test_needed: it may happen that no accurate diagnosis can be issued basing on the information provided by the variables involved in the process. In this case, a test is recommended in order to gather further data so as to launch a more precise diagnose. The tests are mainly oriented to discriminate whether the vehicle is moving or not. Basing on this, slippery situations on the vehicle wheels can be easily deduced. The procedures started during a test can involve some vehicle movements to check its manoeuvrability and a new variable, based on computer vision processing could be added in some cases. Likewise, the tests usually imply computationally expensive processes based on optical flow calculations intended to accurately compute egomotion. This is not a problem for real time performance, as a test will only be carried out upon detecting an abnormal situation during the execution of a mission.

The expert rules are expected to use a reduced number of variables (per rule), highlighting the role of the most important variables. As the final rule base will result of the merging of expert rules and induced rules, the expert rules are not supposed to form an autonomous operative rule base. Defaults, such as incompleteness, are considered as normal at this step.

To gain further knowledge on how expert rules are stated, let's consider the following analysis. Concerning range measurements provided by the ultrasound based sensors, some useful information could be extracted in order to determine whether the vehicle is moving or not. The next options are possible.

- Range measurements are different from null (something is detected within the detection range) and its derivative 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.
- Range measurements are null (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 either issued.
- Range measurements are different from null (something is detected within the detection range) and its derivative is zero. This means that the vehicle is not moving.

According to the three previous possibilities an expert rule could be constructed by following the next reasoning. If range measurements are different from null and its derivative is zero then the environment surrounding the robot is no changing. If on these circumstances the vehicle odometry system measures a velocity different from zero, it can be deduced that the vehicle is stalled and its wheels are slipping. The rule can be formalized as follows:

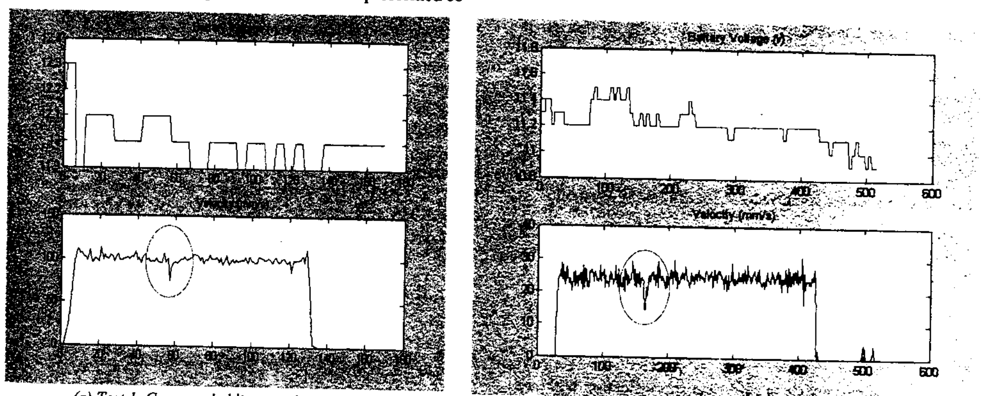
IF *Range_measurement is not_null* **AND**
Derivative_of_range_measurement is zero **AND**
Measured_linear_velocity is not_zero
THEN *Vehicle_stalled*

Another example of expert rule can be produced for the collision and drag case, according to the situation illustrated by figure 2. The rule is provided as follows:

IF *Undershoot_depth* is *high* **AND**
Measured_velocity is *low* **AND**
Difference_of_battey_voltage is *medium*
THEN *Vehicle_drags_obstacle*

Experimentations

In complement with expert knowledge, data are likely to give a good image of variable interaction. For this purpose, some real experiments have been performed so as to collect data concerning the vehicle battery voltage and linear velocity. 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 can not 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 actuators torque so as to rapidly attain 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 slightly down. An illustrative example of this behavior, for two different commanded linear velocity, is shown in Figure 2, where the vehicle battery voltage and linear velocity are depicted for a real collision-and-drag case. The effect of this situation is reflected in measured velocity of the vehicle when colliding and dragging the obstacle. This effect is an undershoot in velocity that is marked in the figures. The undershoot in left figure has a width of 40 ms and a depth of 25% of velocity, while in the right one has a width of 110 ms and a depth of 44%. These values will be considered in the next section to derive the fuzzy partitions corresponding to *undershoot_depth*.



(a) Test 1. Commanded linear velocity: 100 mm/s. (b) Test 2. Commanded linear velocity: 25 mm/s

Figure 2. Vehicle Battery Voltage and Linear Velocity during a collision-and-drag case.

But this is not the only motion problem that can occur during BART operation. In fact, the vehicle could get completely stalled upon collision, depending on the obstacle weight and roughness. Similar experiments have been carried out under these circumstances intended to gather as much real data as possible. These data will help produce the induced complementary knowledge necessary to provide accurate diagnosis for BART motion problems.

Knowledge extraction is made up two different steps. First, 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 Figure 2, after being filtered in order to remove noise, derivative must be obtained to build the variable *Derivative_of_linear_velocity*. The second step is the rule induction one. As rule base interpretability is of prime concern, several constraint are used to ensure that induced rules are interpretable rules [3,4]. Three conditions are to be met:

- Readable fuzzy partitions (reasonable number of fuzzy sets, distinguishability, significant overlapping level, completeness)
- A small number of rules
- Incomplete rules for large systems

The induction method includes fuzzy partitioning: the first step of the procedure generates a hierarchy of fuzzy partitions of different sizes for each input variable.

Figure 3 shows the result of deriving a fuzzy partition for the variable *Undershoot_depth* by working on the linear velocity of the vehicle (once filtered by a Kalman Filter) using HFP partitioning method [5]. After analysis by the experts, the partition with four membership functions has been considered as the more appropriate to be used. In addition the expert has introduced one additional fuzzy set (Figure 4) to represent *undershoot depth **null*** considering that this case was not contained in the experimental data. A similar process is carried out for the rest of variables.

The second step generates several FIS of increasing complexity. At each iteration one fuzzy set is added on a few input variables, chosen to best improve accuracy, using the previously found hierarchies of fuzzy partitions. Only the most influential rules are kept for each system. The trade-off between complexity and accuracy is user controlled.

The final step consists in rule base simplification with the objective to obtain incomplete rules. Between global variable removal and selection done at rule level, an intermediate selection level allows for evaluating the influence of a given variable within a context defined by the other inputs, and represented by a group of rules. The simplification stage tolerates some loss of accuracy while being guided by new indices complementary to the usual numerical performance index. According to the degree of simplification several FIS are designed.

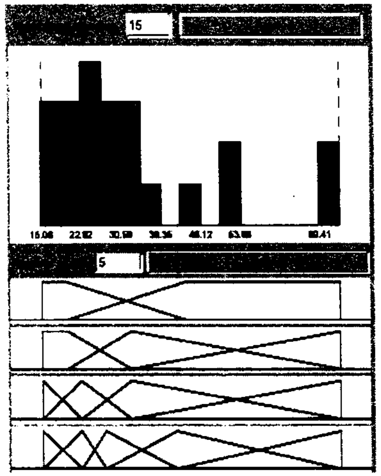


Figure 3. Extraction of fuzzy partition with different granularity for the variable *Undershoot_depth*.

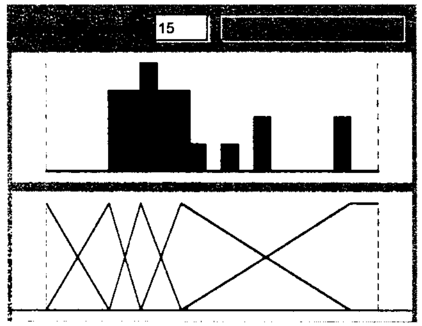


Figure 4. Expert selected fuzzy partition.

Integration

The procedure is made up of two hierarchical steps. Firstly, a common fuzzy input space is designed according to both the data and expert knowledge. The compatibility of the two types of partitions (expert and

induced) is checked according three criteria: range of interest, granularity and semantic interpretation.

Secondly, induced and expert rules have to be merged into a new rule base. Thanks to the common universe resulting from the first step, rule comparison can be made at the linguistic level only. The possible conflict situations are managed and the most important rule base features, consistency, redundancy and completeness, are studied. An analysis of the possible conflicts and their management is described in [6].

Conclusions and future work

The cooperation of expert knowledge and data in system design remains an open problem, especially when the goal is to get a system which is both accurate and interpretable.

The two kinds of knowledge, expert knowledge and data, convey complementary information. The objective is to extract their specific contribution to make the cooperation benefit for the system to be designed.

The distribution of data is used to design strong fuzzy partitions for each separate variable under expert control. This type of partitioning ensures each fuzzy set can be attached a linguistic label. Several quality criteria are available to characterize the partitions, but the final semantic agreement is given by the expert: the fuzzy set centers must correspond to possible prototypes of the corresponding labels. Then, rules defined by these linguistic labels can be written by the expert and induced from data. The second level of integration aims to merge the two rule bases into new one checking the fundamental properties of completeness, consistency and redundancy.

A portable software, which implements the whole process, is currently under design in the framework of the ADVOCATE II project. It will be available soon through the Internet.

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 Electronik GmbH, Ifremer, HUGIN Expert A/S, INNOVA S.p.A, and E-MOTIVE. The European Commission (IST-2001-34508) supports this work.

References

1. M. A. Sotelo et al. "ADVOCATE II: ADVanced On-board diagnosis and Control of Autonomous sysTEms II". EUROCAST2003, 9th International Workshop on Computer Aided Systems Theory. Las Palmas de Gran Canaria, 273-276, Spain, February 2003.
2. Zadeh L.A., "Fuzzy sets", Information and Control, Vol. 8, pages 338-353. 1965.
3. Guillaume S., "Designing fuzzy inference systems from data: an interpretability-oriented review", IEEE Transactions on Fuzzy Systems, Vol. 9 (3), pages 426-443. 2001.
4. J. Casillas, O. Cordón, F. Herrera and L. Magdalena (editors), Interpretability Issues in Fuzzy Modeling. Vol. 128 of Studies in Fuzziness and Soft Computing. Springer, 2003.
5. S.Guillaume and B. Charnomordic, "A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data", in Interpretability Issues in Fuzzy Modeling, Casillas, Cordón, Herrera and Magdalena (editors), pages 148-175. Springer, 2003.
6. S. Guillaume and L. Magdalena, "Typologie des conflicts dans les systèmes de Mamdani", Actes de LFA'03, France. Cépaduès Editions.