Fuzzy-4D/RCS for Unmanned Aerial Vehicles

Miguel A. Olivares-Méndez and Pascual Campoy and Iván Mondragón and Carol Martínez

Abstract This paper presents an improvement of the cognitive architecture, 4D/RCS, developed by the NIST. This improvement consist of the insertion of Fuzzy Logic cells (FLCs), in different parts and hierarchy levels of the architecture, and the adaptation of this architecture for Unmanned Aerial Vehicles (UAVs). This advance provides an improvement in the functionality of the system based on the uses of the *Miguel Olivares' Fuzzy Software* for the definition of the FLCs and its adaptative learning algorithm. These adaptative-FLCs contribute with the reduction of the uncertainty in the data sensor adquisition, a more adaptative behavior of the system to the real world and the reduction of the computational cost in the decision making.

1 Introduction

The autonomous decision making based on the data sensor acquisition is one of the most pursued topics in robotics. There so much approach developed in order to solve this problem, and commonly it is based on cognitive techniques and the use of ontologies. One of the earliest was the ACT architecture [6]. ACT grew out of research on human memory. ACT has evolved into ACT*, and then in, ACT-R, being used in an Advanced Decision Architectures Collaborative Technology Alliance. The Soar architecture [11] grew out of research on human problem solving, and has been used for problem solving, language understanding, computational linguistics, theorem proving, and cognitive modeling. Other cognitive architectures include Prodigy [12], ICARUS [22], IMPRINT (Improved Performance Research Integration Tool) [7], EPIC (Executive-Process Interactive Control) [10], and 4D/RCS (Real-time Control Systems) [19] [3] [1].

Miguel A. Olivares-Méndez and et al.
Universidad Politécnica de Madrid, Jose Gutierrez Abascal 2, 28006, Madrid, Spain,
e-mail: miguelangel.olivares@upm.es

In this work is used the 4D/RCS. This architecture is a control system inspired by a theory of cerebellar function. 4D/RCS models the brain as a hierarchy of goal-directed sensory-interactive intelligent control processes that theoretically could be implemented by neural nets, finite state automata, cost-guided search, or production rules. 4D/RCS is similar to other cognitive architectures in that it represents procedural information in terms of production rules, and represents declarative information in abstract data structures such as frames, classes, and semantic nets. 4D/RCS differs from other cognitive architectures in that it also includes signals, images, and maps in its knowledge database, and maintains a tight real-time coupling between metric and symbolic data structures in its world model. 4D/RCS is also different in: a) its focus on task decomposition as the fundamental organizing principle; b) its level of specificity in the assignment of duties and responsibilities to agents and units in the behavior generating hierarchy; and c) its emphasis on controlling real machines in real-world environments. The criticized points of this architecture are the lacks of connection with the real world and the high computational cost to plan an action **ref**. To these points are focussed the improvement that is presented in this paper, based on the inclusion of Fuzzy Logic Cells (FLCs) in different parts of a 4D/RCS architecture. This FLCs are implemented using the *Miguel Olivares' Fuzzy Software* (MOFS) [21] [8] which allow the possibilities to the FLCs to adapt to the environment by its adaptative learning algorithm, and reduce the computational cost by the Fuzzy Logic nature, and, also, reduce the uncertainty of the sensor data acquisition. Being one of the advantages of the Fuzzy logic in contrast to the conventional control, the possibility to include the different data sensor like inputs of a multiple inputs and single/multiple outputs, making the fuzzy logic a good solution to manage a sensor fusion and a considerable size of data from sensors.

This paper is divided in Section 2, which explains the 4D/RCS architecture in brief, the Section 3 shows the *Miguel Olivares' Fuzzy Software* by the possibilities that it bring to create the FLCs and the explanation of how the adaptative learning algorithm works. Section 4 presents the explanation of how the Fuzzy Logic Cells are include inside the 4D/RCS architecture. Some behavior definitions for a UAV using the presented improvement is shown in the Section 5.

2 Background of the 4D/RCS

The 4D/RCS integrates the NIST Real-time Control System (RCS) [4] architecture with the German (Universitat der Bundeswehr Munchen) VaMoRs 4-D approach to dynamic machine vision [9]. It incorporates many concepts developed under the U.S. Department of Defense Demo I, Demo II, and Demo III programs [3][23], which demonstrated increasing levels of robotic vehicle autonomy. The theory embodied in 4D/RCS borrows heavily from cognitive psychology, semiotics, neuroscience, and artificial intelligence. It incorporates concepts and techniques from control theory, operations research, game theory, pattern recognition, image understanding, automata theory, and cybernetics from the application domain perspective. The 4D/RCS architecture consists of a multi-resolution hierarchy of feedback control loops between sensing and acting that integrate reactive behavior with perception2, world modeling, and planning - and forming a hybrid deliberative/reactive system [17] [18]. A review of projects that have used RCS and a description of how RCS relates to other intelligent system architectures are contained in [4] [5].

4D/RCS evolved from the bottom up as a real-time intelligent control system for real machines operating on real objects in the real world (robust real-time control). The first version of RCS was developed as a sensory-interactive goal-directed controller for a laboratory robot. The fundamental element is the control loop with a goal, a transition function, a feedback loop, and an action output such as a force, velocity, or position. Over the years, RCS has evolved into an intelligent controller for industrial robots, machine tools, intelligent manufacturing systems, automated general mail facilities, automated stamp distribution systems, automated mining equipment, unmanned underwater vehicles, and unmanned ground vehicles [20] [2] [1]. The most recent version of RCS (4D/RCS) embeds elements of Dickmanns 4-D approach to machine vision within the 4D/RCS control architecture. 4D/RCS consists of a multi-layered multi-resolutional hierarchy of computational nodes each containing elements of Sensory Processing (SP), World Modeling (WM), Value Judgement (VJ), Behavior Generation (BG), and a knowledge database (KD), as shown in Figure 1. Throughout the hierarchy, interaction between SP, WM, VJ, BG, and KD give rise to perception, cognition, and reasoning. At low levels, representations of space and time are short-range and high-resolution. At high levels, distance and time are long-range and low-resolution. This enables high-precision fast-action response at low levels, while long-range plans and abstract concepts are being simultaneously formulated at high levels. The hierarchical approach also helps to manage computational complexity.

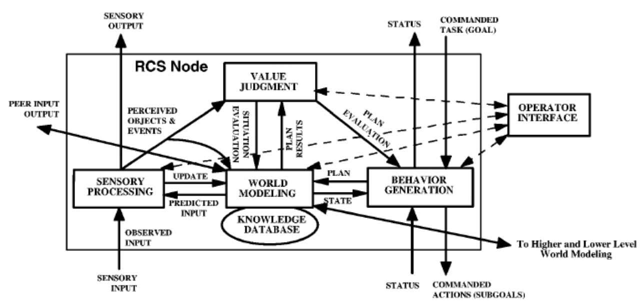

Fig. 1 Node of the 4D/RCS cognitive architecture

4D/RCS closes feedback loops at every level, through every node. SP processes focus attention (i.e., window regions of space or time), group (i.e., segment regions into entities), compute entity attributes, estimate entity state, and assign entities to classes at every level. WM processes maintain a rich and dynamic database of information about the world in the form of images, maps, entities, events, and relationships at every level. Other WM processes use that information to generate estimates and predictions that support perception, reasoning, and planning at every level. VJ processes assign worth and importance to objects and events, compute confidence levels for variables in the knowledge database, and evaluate the anticipated results of hypothesized plans.

3 Fuzzy Logic Cells

This section explains the definition and implementation of the Fuzzy Logic Cells and the Adaptive Learning Algorithm using the *Miguel Olivares' Fuzzy Software* (MOFS).

3.1 Fuzzy Controllers

For this work it used the software MOFS, developed in previous works [14], [13], [16] and [15]. This software was independently designed, defining one class for each part of the fuzzy-logic environment (variables, rules, membership functions, and defuzzification modes), in order to facilitate future updates and easy interaction with the system, as is shown in Fig. 2. There are different classes depending on the system that want to create; being possible to define the number of inputs and outputs that is prefered. Furthermore, it allows to make parts of the system work in serial or parallel mode. Also, is possible to define the type of the membership functions, the fuzzification and defuzzification methods. For this work is used for the inference process (in the defuzzification) an adaptation of the minimum and product classic method with the consideration of the weights assigned at each fuzzy-rule. The idea about the weights will be explained in the next subsection. For the defuzzification part itself, it is used the Height Method, but with the adaptation of weighting of the fuzzy-rules (Eq. 1a, 1b), where w^l is the rule weight and W is the maximum weight possible.

$$y = \frac{\sum_{l=1}^{M} \bar{y}^l \, min\left(\mu_{B'}(\bar{y}^l)\right) \frac{w^l}{W}}{\sum_{l=1}^{M} min\left(\mu_{B'}(\bar{y}^l)\right) \frac{w^l}{W}} \tag{1a}$$

$$y = \frac{\sum_{l=1}^{M} \bar{y}^l \prod\left(\mu_{B'}(\bar{y}^l)\right) \frac{w^l}{W}}{\sum_{l=1}^{M} \prod\left(\mu_{B'}(\bar{y}^l)\right) \frac{w^l}{W}} \tag{1b}$$

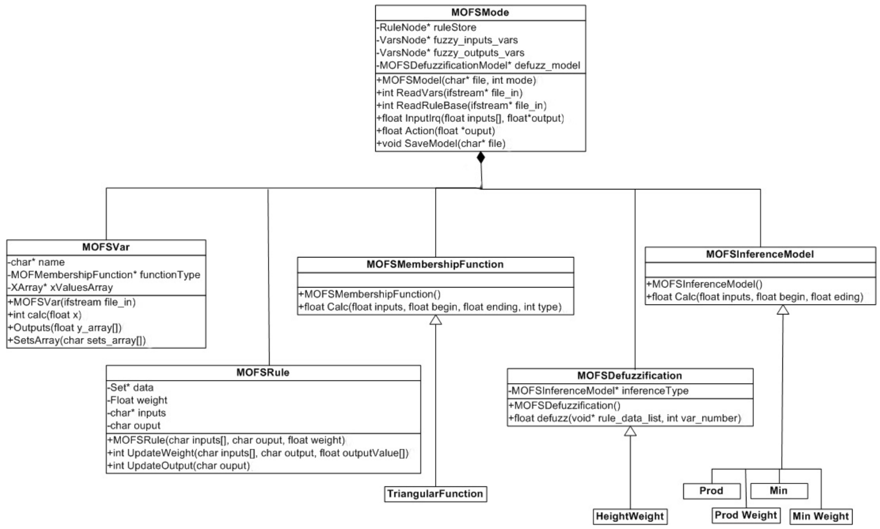

Fig. 2 Software definition.

3.2 Adaptative learning algorithm

The MOFS-Learning algorithm is based on the idea of the synapse-weight, where a weight variable for each rule is defined and represents the contribution of each rule in the system output. The default value of the weights is 0.3 and the maximum value is 3. The decision of the user, knowledge database in simulated or real missions will make the weight of the rules increase or decrease depending on the situation and in accordance with the system output. For each situation $2^{number\ of\ inputs}$ rules are selected, given the fact that it is used a simple overlap in the variables.

To develop a robust system it is important to start with a base of knowledge an to accomplish some real tests in order to train the system by a supervised commands and solutions. In the training period the system compares each rule output and with the user decision, which had been fuzzificated by using the same process. When the rule output and the user or the simulated decision are different the system will decrease the weight of the rule, and in the case of a negative rule weight the system will change the output of the rule by using the set of user decisions that has the highest value. In the case where the rule output is like one of the two sets of the user decision the system will increase the weight by a quarter of the set membership value. So with this method the system will adapt its behavior to the users' behavior.

In the execution phase, when the user or simulation are not making any decision the system continues learning, based on the rules that have higher weights, which represents the users behavior. The way in which the system compares the output of the selected rules for the situation with the output of the system, is the same way as in the training phase.

4 Fuzzy-4D/RCS system definition

Based on the architecture of the 4D/RCS and the Fuzzy Logic Cells developed using the MOFS, a new architecture of Fuzzy-4D/RCS is presented in this section. This improvement consists of the inclusion of Fuzzy Logic Cells (FLC) in different parts of the 4D/RCS architecture. The affected modules are SP, WM, VJ and BG. The number of the FLC working depend on the situation, being possible to act from three to one.

The uses of those FLCs improve the 4D/RCS system by the next advantages:

- The adaptative learning algorithm implemented inside the MOFS, explained in the previous section, improve the learning of the system to face new situations and to consolidate old ones by the simulations in the virtual environment and in the real world, improving one of the parts of the 4D/RCS that is being criticized by the lacks of connection with the real world.
- The MOFS system allows to adapt different rules bases, in order to it can act in different ways to similar situations, but with different level of emergency, for example, increasing the velocity in the basic movements to accomplish faster one mission.
- Reduce the uncertainty of the sensor data acquisition. It is possible by the use of FLCs for analyze this kind of data. The Fuzzy logic allows to reduce the uncertainty for a better recognition of the situations, and objects, and among others.
- The use of the FLCs for take decisions improve the behavior of the system and reduce the time of computation, being the high computational cost of the 4D/RCS architecture another of the most criticized point Balakirsky.

The function of the FLC depends on the module and the hierarchical level where each one is include. Next, the explanation of the FLCs functions inside the cognitive architecture is presented:

Sensory Processing. The principal function of this module is to keep in contact with the environment through the different sensor installed in the aircraft. In this module, the FLC acts to reduce the uncertainty of the sensor measures, by the reduction of false-positives and the noise, at the first level. For the next levels, different FLC would be train for the recognition of the characteristic parts of some objects, for, in the upper levels, the detection of full objects, maps or parts of complex objects by using fuzzy logic for clustering or patter recognition with the information that flow through the bottom up architecture.

World Modeling. The reduced uncertainty information and the objects recognition information from the Sensory Processing modules is passed to the Behavior modules. This information is compared with the knowledge database to detects the different situations that the aircraft is facing. Commonly the situations that the aircraft can be involved are not well defined, in the fact that the information could not belong to just one situation. For this reason the information and its belongings to the different situations are measure using the fuzzy logic.

Value Judgement. Value judgment evaluates perceived and planned situations, thereby enabling behavior generation to select goals and set priorities. For improve the computation task of the important (for attention), and what is rewarding or punishing (for learning), another FLC is introduce in this module, using the developed learning system of the MOFS, explained in the previous section. The FLC-Value judgement assigns priorities and computes the level of resources to be allocated to tasks. It assigns values to recognized objects and events, and computes confidence factors for observed, estimated, and predicted attributes and states. The union between the FLC-Value judgement and the World Modeling, simulates situations for modified the rules-base of the FLC of the task knowledge for improve the decisions of the Behavior Generation module.

Behavior Generation. Behavior generation plans and executes tasks in order to accomplish successful mission goals. The introduced FLC in the Behavior generation module uses task knowledge, skills, and abilities along with knowledge in the world model to plan and control appropriate behavior in the pursuit of goals, by uses fuzzy rules. Furthermore the FLC modified it base of rules by learning about the successful accomplished tasks.

The explained FUZZY-4D/RCS node architecture is shown in fig 3.

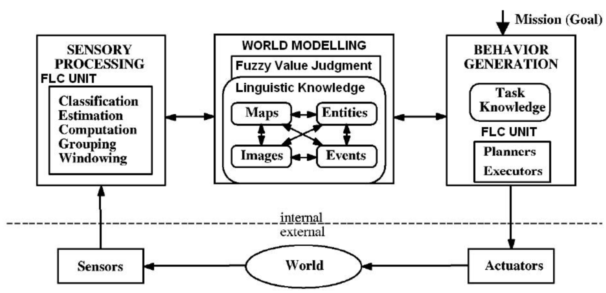

Fig. 3 Internal structure of 4D/FUZZY-RCS node

5 Fuzzy-4D/RCS behavior definition for a UAV

The defined behaviors are located in the ”Subsystem Level” of the hierarchy of the architecture. These are bottom-up communication, up-bottom and at the same level, as shown below, in the explanation of these behavior. The FLCs not only help in the acquisition external data, processing and creation of these plans, also integrated within the control and management of the UAV. The hierarchical structure of the architecture 4D/FUZZYRCS with different behaviors can be defined seen in Figure 4.

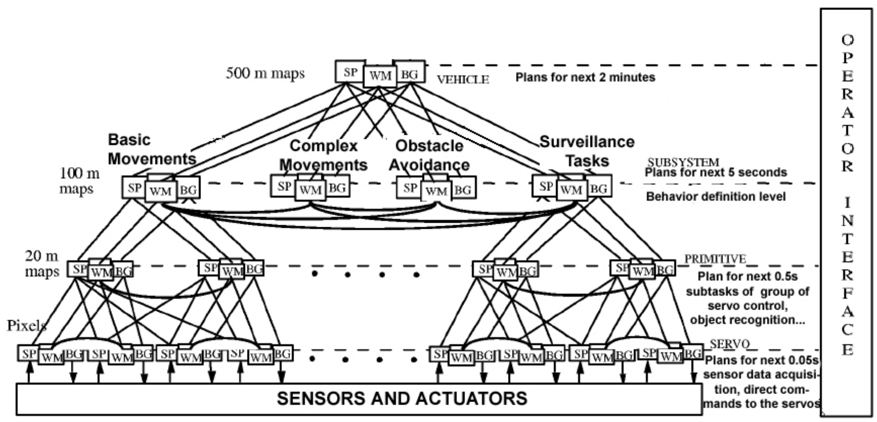

Fig. 4 Hierarchical architecture of the FUZZY-4D/RCS

5.1 Basic Movement

The main performance is the basic movement, since it is the most often used by other modules and the indispensable task for the UAV is in the air. According to the structure our architecture, this behavior is controlled at the level Lowest ”servo level” control of the 3 servos charge manage the main rotor plate, the servo manager acceleration of the engine and tail rotor servo (yaw), total 5 servos. At the next level, ”Primitive level” would defined the tasks that make possible the basic side movements, and elevation, and descend of the aircraft.

5.2 Complex Movements

The complex movements include different behaviors defined separately in the ”Subsystem Level” as well. These are: takeoff, landing, hovering, tracking moving base station and the action to establish a surveillance route. These complex movements need the feedback signal of different sensors, such as GPS, gyroscopes, cameras, among others. Furthermore, the different behaviors communicate on the same level with the behavior of basic movements.

5.3 Obstacle Avoidance

This behavior is also very importance for the control of the aircraft. This need the feedback of the data issued by multiple sensors and acts in communication with the motion basic, and also in direct communication with the sub-tasks defined for this behavior. This behavior based on the use of vision cameras, and LADAR for data acquisition and management within the Fuzzy Logic Cells.

5.4 Detection Tasks

It also defined some detection systems. These are defined separately, taking into account many sub-common tasks. The defined detection behaviors are: detection of fires, for it would use the heat camera, which after detecting a heat in a wood or other surface, send location data to the central station. Other detection system is for people in danger situations, it uses the cameras, which, through image processing algorithms and the situations analyzes, can detect people in this kind of situations. Once the person is found the UAV must be capable to send a rescue kit with a parachute. It is usable both on land and on water, rescue kits using different according to the medium.

6 Conclusions and Future Works

This paper presents an improvement of the cognitive architecture 4D/RCS. The implementation of the Fuzzy Logic Cells (FLCs) using the MOFS, are used for this aim. The work mode depends on the part of the node and the level of the hierarchical or the architecture that it is includes. The principal objectives of this improvement is to have a better interaction with the real world, using the learning algorithm developed inside the MOFS, with better adaptation to new situations. The reduction of the uncertainty of the data sensor acquisition, and better object and situation recognition by the Fuzzy Logic technique. Obtain a computational reduction of the decision making task, using a FLC working like Fuzzy controller.

For future works, a real tests are proposed in order to evaluate this approach using a real UAV.

Acknowledgements This work is the product of several research stages at the Computer Vision Group Universidad Politécnica de Madrid. Has been sponsored by the Spanish Science and Technology Ministry under grants CICYT DPI2004-06624, CICYT DPI2000-1561-C02-02 and MICYT DPI2007-66156.

References

1. Albus, J., Bostelman, R., Hong, T., Chang, T., Shackleford, W., Shneier, M.: Integrating learning into a hierarchical vehicle control system. Integrated Computer-Aided Engineering pp. 121 – 139
2. Albus, J., Bostelman, R.V., Hong, T., Shackleford, W.P., Shneier, M.O.: The lagr project - integrating learning into the 4d/rcs control hierarchy. International Conference on Informatics in Control, Automation and Robotics, Setubal, Portugal pp. 1 – 5
3. Albus, J., Huang, H., Messina, E., Murphy, K., Juberts, M., Lacaze, A., Balakirsky, S., Schneier, M., Hong, T., Scott, H., Proctor, F., Shackleford, W., Michaloski, J., Kramer, A.W.T., Dagalakis, N., Rippey, W., Stouffer, K., Legowik, S., Evans, J., Bostelman, R., Norcross, R., Jacoff, A., Szabo, S., Falco, J., Bunch, R., Gilsinn, J., Chang, T., Tsai, T., Meystel, A., Barbera, A., Fitzgerald, M.L., Giorno, M., Finkelstein, R.: 4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems Version 2.0. NISTIR (2002)
4. Albus, J.S., Meystel, A.M.: A reference model architecture for design and implementation of intelligent control in large and complex systems,. International Journal of Intelligent Control and Systems pp. 15 – 30 (1996)
5. Albus, J.S., Meystel, A.M.: Engineering of Mind: An Introduction to the Science of Intelligent Systems. John Wiley and Sons (2001)
6. Anderson, J.R.: The architecture of cognition (1983)
7. Anderson, J.R.: The Architecture of Cognition. L. Erlbaum Associates Inc., Hillsdale, NJ, USA (1995)
8. Balakirsky, S.: A framework for planning with incrementally created graphs (2003)
9. Dickmanns, E., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Thomanek, F., Schiehlen, J.: The seeing passenger car 'vamors-p'. In: Intelligent Vehicles '94 Symposium, Proceedings of the, pp. 68–73 (1994). DOI 10.1109/IVS.1994.639472
10. Kieras, D.E., Meyer, D.E.: An overview of the epic architecture for cognition and performance with application to human-computer interaction (1997)
11. Laird, J.E., Newell, A., Rosenbloom, P.S.: Soar: an architecture for general intelligence. Artif. Intell. **33**(1), 1–64 (1987). DOI http://dx.doi.org/10.1016/0004-3702(87)90050-6
12. Minton, S.: Quantitative results concerning the utility of explanation-based learning. Artif. Intell. **42**(2-3), 363–391 (1990)
13. Olivares, M., Campoy, P., , Correa, J., Martinez, C., Mondragon, I.: Fuzzy control system navigation using priority areas. In: Proceedings of the 8th International FLINS Conference, pp. 987–996. Madrid,Spain (2008)
14. Olivares, M., Madrigal, J.: Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments. Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on pp. 1–6 (2007). DOI 10.1109/WISP.2007.4447633
15. Olivares-Mendez, M.A., Campoy, P., Martinez, C., Mondragon, I.: A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. IEEE/RSJ International Conference on Intelligent Robots and Systems 2009. IROS09 (2009)
16. Olivares-Mendez, M.A., Campoy, P., Martinez, C., Mondragon, I.: Visual servoing using fuzzy controllers on an unmanned aerial vehicle. Eurofuse workshop 09, Preference modelling and decision analysis (2009)
17. Rasmussen, C.: Laser range-, color-, and texture-based classifiers for segmenting marginal roads. In: IEEE International Conference on Computer Vision and Pattern Recognition Technical Sketches (2001). URL http://nameless.cis.udel.edu/pubs/2001/Ras01a
18. Rasmussen, C.: Combining laser range, color, and texture cues for autonomous road following. In: In IEEE International Conference on Robotics and Automation, pp. 4320–4325 (2002)
19. Schlenoff, C., Albus, J., Messina, E., Barbera, A.J., Madhavan, R., Balakrisky, S.: Using 4d/rcs to address ai knowledge integration. AI Mag. **27**(2), 71–82 (2006)
20. Schlenoff, C., Albus, J., Messina, E., Barbera, A.J., Madhavan, R., Balakrisky, S.: Using 4d/rcs to address ai knowledge integration. AI Mag. **27**(2), 71–82 (2006)

21. Schlenoff, C., Balakirsky, S., Uschold, M., Provine, R., Smith, S.: Using ontologies to aid navigation planning in autonomous vehicles. Knowl. Eng. Rev. **18**(3), 243–255 (2003). DOI http://dx.doi.org/DOI:10.1017/S0269888904000050
22. Shapiro, D., Langley, P.: Controlling physical agents through reactive logic programming. In: AGENTS '99: Proceedings of the third annual conference on Autonomous Agents, pp. 386–387. ACM, New York, NY, USA (1999). DOI http://doi.acm.org/10.1145/301136.301252
23. Shoemaker, C., Bornstein, J.: The demo iii ugv program: a testbed for autonomous navigation research. In: Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings, pp. 644–651 (1998). DOI 10.1109/ISIC.1998.713784