Proceedings of the XI DINAME, 28th February-4th March, 2005 - Ouro Preto - MG - Brazil
Edited by D.A. Rade and V. Steffen Jr. 

# IMPLEMENTATION OF A SENSOR SYSTEM FOR OBSTACLE AVOIDANCE AT A FOUR-LEGGED ROBOT

**Morgado de Gois, Jorge Audrin**
Institute for Mechatronics and System Dynamics - University Duisburg-Essen
47048 Duisburg, Germany
morgado@mechatronik.uni-duisburg.de

**Hiller, Manfred**
Institute for Mechatronics and System Dynamics - University Duisburg-Essen
47048 Duisburg, Germany
hiller@mechatronik.uni-duisburg.de

***Abstract.*** *This work presents a sensor system develop for the robot ALDURO, in order to allow it to detect and avoid obstacles when moving in unstructured terrains. The robot is a 4-legged walking-machine, steered by an operator, which gives simple instructions thourgh a joystick to be processed by a control computer, converting them into appropriate autonomous leg movements. To efective do that the robot has to automatically recognize the obstacles, locate and avoid them. A system based on ultra-sonic sensors was developed to carry this task on and, because of intrinsic problems with such sensors concerning to angular precision, a fuzzy inverse model of them was used to include the uncertainties about the measurements. To provide a map of the robot's surroundings the use of TSK system as data fusion tool is proposed, in order to attend the peculiarities of ALDURO. The sensors are assembled in an* $I^2C$ *net, which communicates with the main controller (a PC) by the serial port, interfaced by a micro-controller.*

***Keywords****: fuzzy logic, four-legged robot, data fusion*

## 1. Introduction

At the Department of Mechatronics of the University Duisburg-Essen is under development the Anthropomorphically Legged and Wheeled Duisburg Robot (ALDURO), a large-scale four-legged hydraulic-driven walking machine with an onboard operator (Fig.1). The machine weights 1600kg, is 3.5m high, equipped with four identical hydraulically actuated legs and supposed to operate in very rugged terrains (Müller, 2002). Operations on smoother terrains enable to replace the feet of the rear legs by wheels, to enlarge speed and stability.

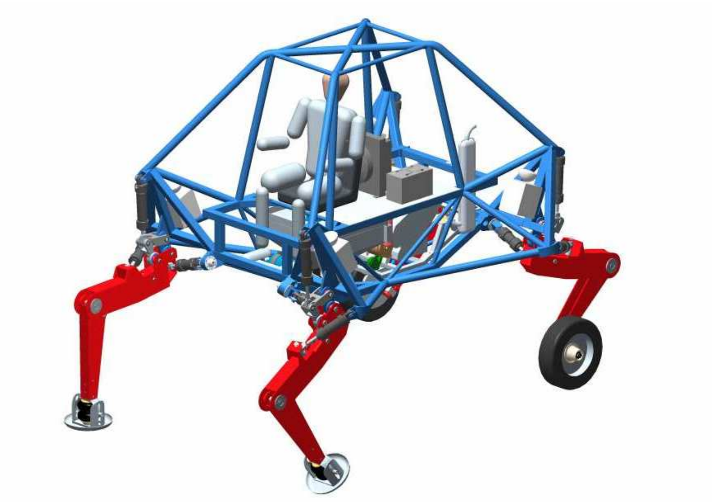

Figure 1. The robot ALDURO

As an anthropomorphic leg, each one has four degrees of freedom, what comprehends a total of sixteen degrees of freedom to control (Müller, Schneider and Hiller, 1998). The operator would not easily achieve to control the machine's movement and anyway it would prevent him of executing another task. To avoid that, the movement generation was automated, enabling the control of the robot by a simple joystick, in this way the operator has just to dictate the desired movement direction and the robot is supposed to execute itself the leg control.

But to achieve real walking and not only stable gaits on plan surfaces is necessary that the robot has information about ground and surround, especially when it controls most of the action. Hereby the collision avoidance task follows as a natural need, which must be implemented accordingly to ALDURO's reality: large dimensions, slow and spatial movements, unstructured environment, and no need of a long term path planning (because of the onboard operator).

Considering such characteristics, it was decided to implement a reactive navigation system, using a local map, based on data from ultrasonic sensors. Such sensors are quite precise with respect to range measures, but suffer of intrinsically poor angular resolution (Risse, 2002), what conversely brings an advantage: they cover a whole volume at each measurement. Because of such inaccuracy, the inverse sensor model plays a relevant role to interpret each measure based on the sensor characteristics. The so formed information has to be added in an appropriated way to a base of knowledge (the local map) through a data fusion process, which provides the necessary input data for the navigation task. In order to consider the uncertainties about the measurements and because of its relative simplicity, a fuzzy approach is used and an *TSK* inference system is developed, especially to save memory, what becomes a problem when modeling a 3D world.

## 2. Inverse Sensor Model

Sensor measurements are prone to errors, which are mainly due to the fact that the true physical sensing device usually does not operate in practice as it was modeled. To interpret the measurements from the transducer, a model of the sensor is needed; but the true physical operation of the transducer is most often too complex to model. Therefore, the interpretation of the sensor measurements often differs from the true physical value of the parameter that is measured. If this measurement error is somewhat random, it can also be interpreted as an uncertainty in the sensor measurement: since the sensor is prone to errors, it is uncertain about the true value of the measure.

Differently from other sensing medias to measure the distance to a target (like laser beam); ultrasonic sensors have quite large emission beams, which makes a priori impossible to know the exact direction. Actually, conjugated use of many ultrasonic sensors or analysis of the shape of the returning echo allows a good approximation of the direction to the detected object. However, such shape analysis requires high quality analogue sensors and a posterior, complex and time-consuming data processing. Besides, conjugated use of sensors requires the possibility to operate receivers and senders separately, what is not always possible, especially with cheap range finder units, where the whole detection is treated as a single operation.

Here a fuzzy inverse sensor model is proposed in order to describe the uncertainty degree about a measure. It is very similar to the one proposed by Oriolo, Ulivi and Venditelli (1999), but the inverse sensor model has to be fitted to the data fusion process that will be used here. Gambino, Oriolo and Ulivi (1996) have worked with occupancy grids, where actually two sensor models are used, the one used here corresponds to the sensor model of the occupancy map, modified to cover a 3D workspace.

### 2.1. Fuzzy coordinates

A set of three coordinates is necessary to completely localize a point with respect to a frame fixed to the sensor. These numbers are presumed to represent uncertainties of these coordinates too, based on the many sensor characteristics (beam shape, gain, precision…) and the measured distance. Considering a reference frame fixed to the sensor unit, with its *X*-axis along the sensor emission axis, a set of polar coordinates can be stated as $\{\alpha, \beta, r\}$, where $\alpha$ represents the rotation around the *Z*-axis, followed by a rotation of angle $\beta$ around the X-axis and $r$ is the distance to the origin as shown in Fig.2.

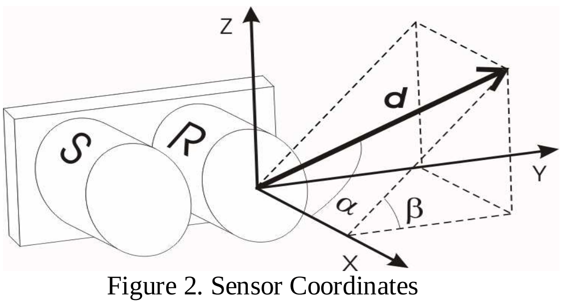

Figure 2. Sensor Coordinates

Through a fuzzification process (Morgado de Gois, Germann and Hiller, 2003), a measure $d$ of the ultrasonic range finder is to be converted in a group of fuzzy numbers $\{A, B, R\}$, which are actually fuzzy sets and thus represent the detected point's coordinates and the uncertainty inherent to them. But a question arises: what would be the predicate to be attended in defining such sets? A good one could be just: 'possible value', the set of the possible values for the target coordinates, given a sensor read value. Then the fuzzy coordinate sets are so stated:

- For a given measure $d$, if the sensor range inaccuracy is given by a multiplicative factor $\varepsilon$, the actual distance lays in the interval $[d\text{-}\varepsilon.d;\ d+\varepsilon.d]$, respecting the ranging limits, i.e., the range domain $[0;\ r_{max}]$. Considering a normal distribution of possible values, a Gaussian membership function $\mu$ is employed.

- In Morgado de Gois and Hiller (2004), is shown the correspondence between the gain factor of the sensor receiver and the possibility of a successful reading for an object on the same direction, or conversely, the grade in which a value (the point location) can be taken as a 'possible value'. As this gain function can be approximated with very good precision by a Gaussian curve, such curve is adopted as membership function for $\alpha$.
- For the angle $\beta$ the membership function is taken as unitary at its whole domain [0; 2π] because of the simple interpretation of the predicate: any value there is as possible as any other.

In this way, based on a measured distance *d*, the uncertainty about the location of a detected point is stated in the present sensor workspace, for each coordinate, by the fuzzy numbers in Eq. (1). But it is necessary to have a single membership grade for each point $s = (\alpha, \beta, r)$ and not one for each coordinate. So in Eq. (2) the Cartesian product is used to find the membership function of the point *s*, stating a new fuzzy set *S*. The classical product operation is used as *T*-norm (the corresponding operator for intersection), what makes the calculation somewhat easier.

$$\begin{aligned} A &= \{ (\alpha, \mu_A(\alpha)) / \alpha \in [0\,; \pi/2] \} \\ B &= \{ (\beta, \mu_B(\beta)) / \beta \in [0\,; r_{\max}] \} \\ R &= \{ (r, \mu_R(r)) / r \in [0\,; 2\pi] \} \end{aligned} \tag{1}$$

$$S = A \times B \times R \Leftrightarrow \mu_S = \mu_A \cap \mu_B \cap \mu_R = \mu_A \cdot \mu_B \cdot \mu_R \tag{2}$$

**3. Map Building**

Given the description of the part of the environment contained in the sensor workspace by the fuzzy set *S*, several of these sets together, through the appropriate inference, are expected to lead to a more exact representation of the obstacles and thus of the surroundings of the robot. It is important to note that, for the ALDURO application, the objective is not to locate an object at once, but to support the robot navigation. To be used, the information from each individual sensor (or from the same sensor at different moments) has to be aggregated to the knowledge base (the map). In order to do that is necessary first to transform the obtained measure and its membership functions to the Cartesian reference system of the local map.

Through the use of the Principle of Extension and coordinates transformation it is possible to obtain *S* in the local map's reference system. Now, by to the construction of an occupancy grid as proposed by Moravec and Elfes (1985), these sets would be aggregated to 3D models of the environment. Here a simplified solution is proposed, where the whole spatial information is not necessary: a 2D map will be generated by a Takagi-Sugeno-Kang (TSK) Fuzzy inference system (Morgado de Gois and Hiller, 2004). The process is quite simple and tries simply to approximate the ground of the environment by a convex surface. That does not reflect the exact reality in most of cases, but describes the necessary features for ALDUROS's navigation. That is so because of the robot's dimensions, then it is not going to walk under objects and a minimal safety distance has to be kept. The process consists of:

- The *XY*-plane (where the robot is considered to walk) is partitioned in many cells;
- A local map around the robot and fixed to its body is constructed. As the robot moves the map is upgraded by simple shift operations, eliminating the cells which fall out of it and introducing new ones;
- The fuzzy set *S* is projected on the map using the *max* operator over the membership grades. Such function is quite complicated to obtain analytically, but considering that for each possible value of *d* the support of *R* is much smaller than the corresponding length of *S*, we neglect the influence of *R*. Then, the support of S is reduced to a surface and is possible to obtain analytically its projection;
- Membership functions are associated to the partitions on the map. As the inputs are fuzzy sets, singletons $\delta_i$ are used at the partitions, what makes the whole calculation a lot easier and faster;
- In such kind of system (Takagi and Sugeno, 1985; Sugeno and Kang, 1988), the activation values $w_i$ are used to make a weighted average of the output functions $f_i$ (trough the so called fuzzy rules). Here it was employed one rule for each partition *i* of the map-cell;
- Linear functions (zero order or first order) are used as output functions.

Basically the fusion process can be seen in Eq.(3). At the end, the parameters (*a*, *b*, *c*, *d*) of the output functions are optimized in order to minimize the error between the estimated height $\hat{z}_k$ of the point $(x_k, y_k)$ and the real one $z_k$, actually this optimization leads to the most possible height.

$$
\begin{aligned}
&\mu_{S_{xy}}(x,y) = \sup_{Z}\left(\mu_S(x,y,z)\right) \\
&w_i = \mu_{S_{xy}} \cap \delta_i \\
&\hat{z}_k = \sum_i w_i \cdot f_i(x_k, y_k) \therefore f = \begin{cases} f = d \leftrightarrow \text{zero order} \\ f = a\cdot x + b\cdot y + c \leftrightarrow 1^{st} \text{ order} \end{cases}
\end{aligned} \tag{3}
$$

**4. Simulations**

Here is shown a comparison between results from fusion carried on by occupancy grid and by TSK system. Two kinds of linear functions were tested as output functions: constants and planes. The first one has one parameter per rule while the latter, three parameters, what makes the simulations much slower, since the dimensions of the matrices employed at the calculation are directly proportional to the number of parameters per rule. To the adjustment of these parameters was used recursive Least Squares method, and the number of samples processed at each cycle was adjusted at both models to fit the waiting time of the sensor device (considering the use of many sensors), in order to obtain the fastest processing. To execute the simulations:

- Number of partitions on the map was varied;
- Different input terrains (randomly generated) were used;
- Different number of samples (simulated measurements).

In Fig.3 are shown results obtained from a representative simulation. The contour plots in Fig.3 represent the input terrain (3.a), the occupancy map from occupancy grid projected onto the *XY*-plane by the *max* operator (3.b), the map constructed through *TSK* system with constants as output (3.c) and with planes as output functions (3.d).

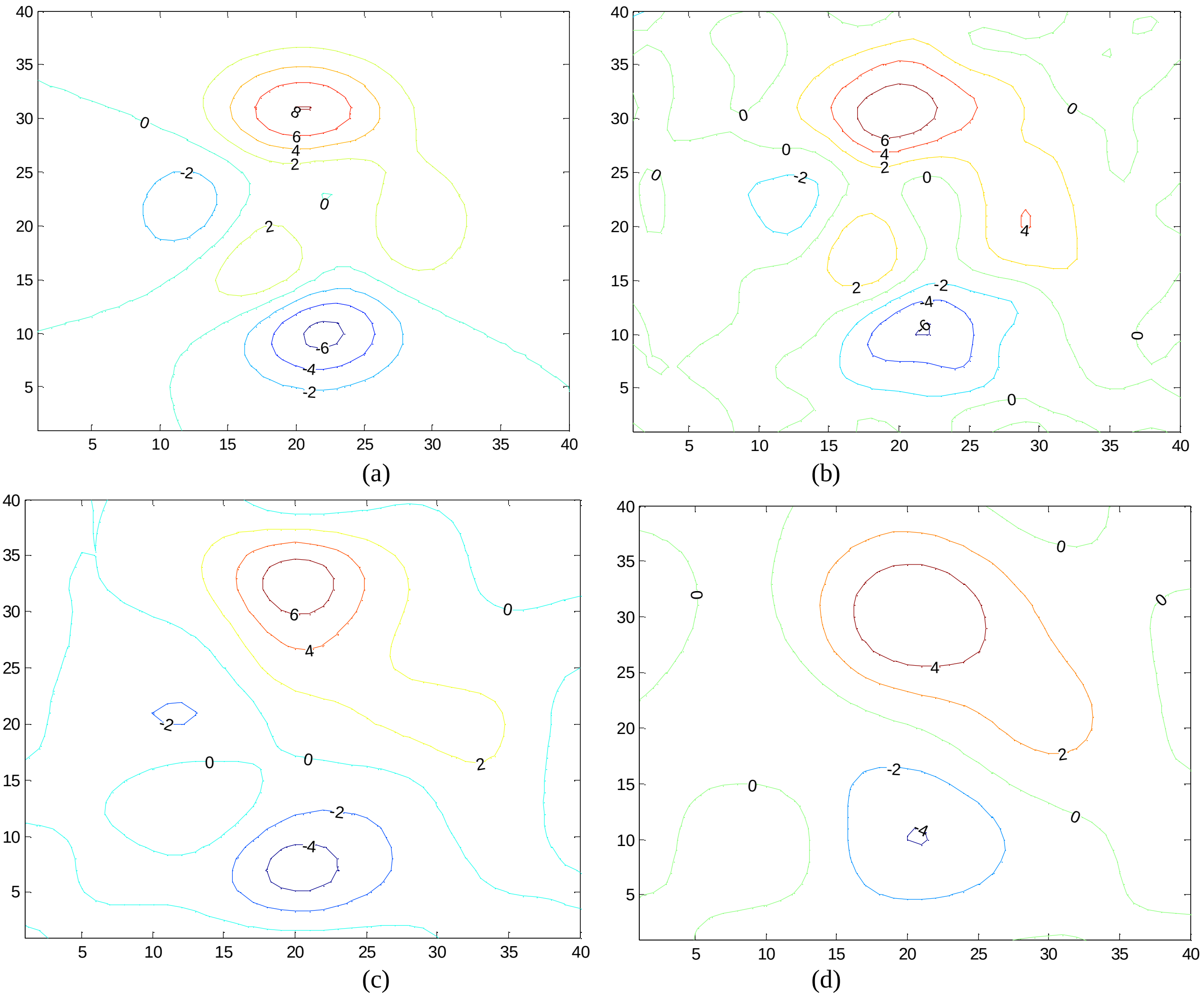

Figure 3. (a) Input Terrain; Maps obtained by (b) TSK with planes as output functions; (c) TSK with constants as outputs and (d) Occupancy Grid.

As expected the TSK system seems to be a little more sensitive to different combinations of terrain and number of partitions than the occupancy grid, but the general performance holds. Comparing the best results of both to a given terrain (considering different number of partitions), the results of TSK are closer to the real terrain than the ones obtained by occupancy grid. TSK model gives a smoother result than occupancy grid if just a few samples are available. This advantage is more remarkable when planes are used as output functions.

**5. Planing**

Navigation techniques can be basically classified within two major groups: reactive and deliberative navigation. The proposed solution for ALDURO is a mix of both. Actually, the incremental building of a dynamic map of the environment (a typical deliberative task) is employed together to the formulation of local plans (typically reactive).

There are many different techniques to carry out reactive motion planning, and for this particular application some peculiarities are important when choosing it. In this case the legs of the robot make full spatial movements and in addition, each leg has to fulfill the kinematical constraints imposed by the body movement and the joints. The controller developed here is based on the concept of general perception (Braunstingl, Sanz and Ezkerra, 1995; Braunstingl, 1996) applied to each shank. The reason to consider the shanks is because they are the most exposed area of the robot, the most probable area to undergo a collision, then the perception vector for a shank is calculated (with respect to a reference point **c** on it) considering each point on the local map.

Two behaviors are implemented by means of variations of the original technique. In order to do that, the points on the map are divided in two groups: low points (possible for the robot to overcome with a simple step) and the high ones. For the points of the group *high* the perception vector $\mathbf{P}_H$ is calculated just like originally proposed for 2D movement, as shown in Eq. (4).

$$
\begin{aligned}
l_k &= \left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} \right\| \\
\mathbf{p}_k^H &= \frac{l_{\max}^H - l_k}{l_{\max}^H} \\
\mathbf{p}^H &= \left|\mathbf{p}_k^H\right|_{\max} \cdot \frac{\sum \mathbf{p}_k^H}{\left|\sum \mathbf{p}_k^H\right|}
\end{aligned}
\qquad (4)
$$

Originally $l_{max}$ is $r_{max}$, here it was set as suitable value to pass to the local map. The behavior correspondent to the group *low* is a little more complicated, because it aims a vertical movement to overcome small obstacles on the way. The heights of the map points are taken in account, generating the perception vector $\mathbf{P}_L$ like in Eq.(5), but this behavior is active just when the foot hangs on the air in movement.

$$
\begin{aligned}
Q &= \left\{ \mathbf{q} \in local\ map \,/\, \left| \mathbf{q} - \mathbf{c} - \mathbf{v}_c \cdot \Delta t \right| = l_{\max}^L \right\} \\
l_k &= \left\| \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \right\| \Leftrightarrow (x_k, y_k, z_k) \in Q \\
\mathbf{p}_k^L &= \frac{l_{\max}^L - l_k}{l_{\max}^L} \\
\mathbf{p}^L &= \left|\mathbf{p}_k^L\right|_{\max} \cdot \frac{\sum \mathbf{p}_k^L}{\left|\sum \mathbf{p}_k^L\right|}
\end{aligned}
\qquad (5)
$$

A fuzzy inference system which has the heights of the points as inputs with two partitions (*low* and *high*) and two rules, each one corresponding to Eq.(4) and Eq.(5), generates the controller outputs. Such outputs are used as corrections to the movement dictated by the step generator. If the correction is too big, it is transmitted to the movement of the robot body through the kinematical constraints between the body and the legs. The overall result can be seen in Fig.4, where a simulation with starting point at (1; 5) and a desired final point (9; 1) is shown. The robot performs the walking without problems, reaching the target point and avoiding the possible obstacles on its way. The use of a local map has sure increased the processing time in comparison to normal general perception navigation, but the improvement in information quality by the inclusion of the uncertainties of the sensors make that necessary. The result of such improvement is realizable by the performance of the robot at the many simulations executed with many different unstructured terrains.

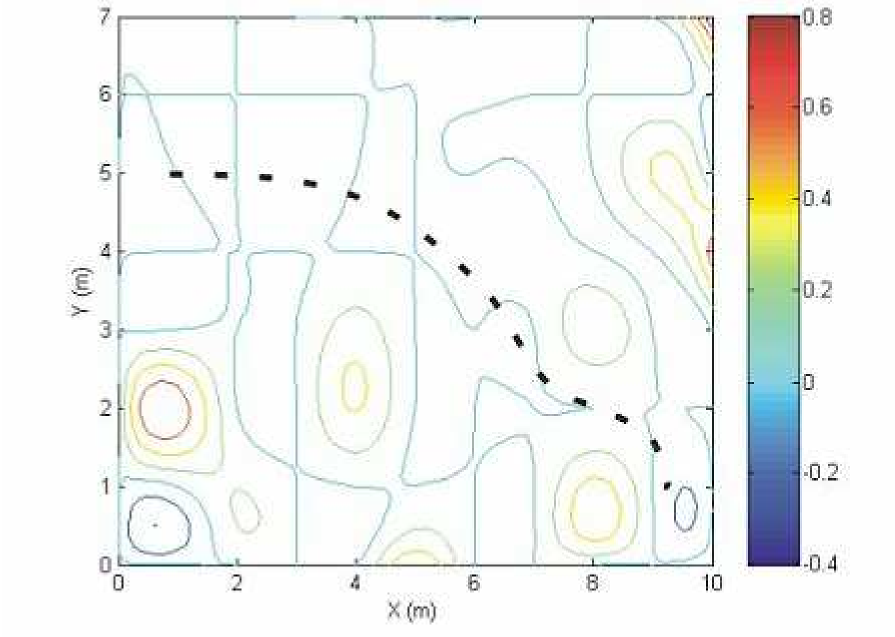

Figure 4. Robot path during simulation

## 6. Implementation

In this implementation the sensing and controlling tasks are structured in a modular partly hierarchical, partly parallel controller architecture. The main controller in ALDURO is a PC, running the modular software environment *MCA2* (Modular Control Architecture Version 2). The encapsulated modules are arranged in a hierarchical structure and consist of defined inputs and outputs. Each module carries out a certain task and communicates with the connected modules. The whole work presented here is implemented in 2 modules: map building and obstacle avoidance.

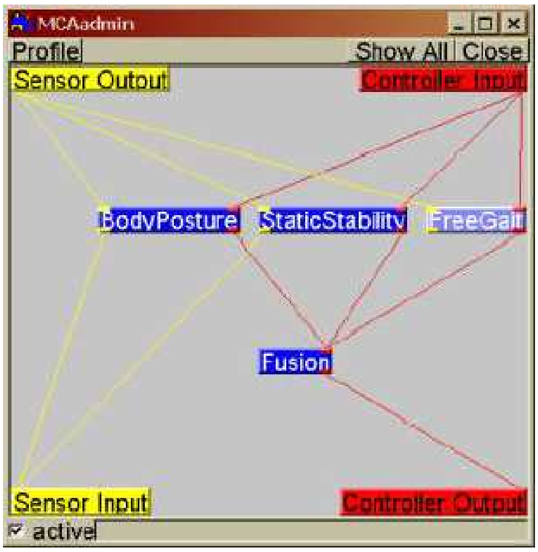

Figure 5. *MCA2* module for ALDURO's control

The used sonar units are SRF08 by Devantech, which provide digital output, having a nominal range of 6m, which is actually much larger. The sensors are connected to an *OOPic* microcontroller through an $I^2C$ network, what enables fast and reliable communication, but this realization presents the drawback that these sensors accept not more than 16 different addresses, making necessary the use of further microcontrollers if we want to use more sensors. That is easily achieved appending them to the $I^2C$ network, which is connected to PC through the serial port.

## 7. References


Braunstingl, R., Sanz, P. and Ezkerra, J.M., 1995, "Fuzzy logic wall following of a mobile robot based on the concept of general perception", ICAR '95 - 7th International Conference on Advanced Robotics, Saint Feliu de Guixols, Spain, pp. 367 - 376.

Braunstingl, R., 1996, “Improving reactive navigation using perception tracking”, in: Jamshidi, M., Pin, F. and Dauchez, P., 1996, “Robotics and Manufacturing, Recent Trends in Research and Applications”, Vol. 6, New York, USA, pp. 25-30.

Gambino, F., Oriolo, G. and Ulivi, G., 1996, “A comparison of three uncertainty calculus techniques for ultrasonic map building”, Proceedings of the 1996 SPIE Int. Symp. on Aerospace/Defense Sensing and Control, Orlando, USA, pp. 249--260.

Moravec, H.P. and Elfes, A., 1985, “High Resolution Maps from Wide Angle Sonar”, Proceedings of the 1985’s IEEE International Conference on Robotics and Automation.

Morgado de Gois, J.A., Germann, D. and Hiller, M., 2003, “Sensor-based ground detection in unstructured terrain for the walking-machine ALDURO”, Proceedings of the 6th International Conference on Climbing and Walking Robots CLAWAR, Catania, Italy, pp. 911-918.

Morgado de Gois, J.A. and Hiller, M., 2004, “Ultrasound sensor system with fuzzy data processing”, Proceedings of the 7th International Conference on Climbing and Walking Robots CLAWAR, Madrid, Spain.

Müller, J., Schneider, M. and Hiller, M., 1998, “Modelling and simulation of the large-scale hydraulically driven ALDURO”, Proceedings of the EUROMECH Colloquium, Munich, p. 375, and Proceedings of the Workshop Autonomous Walking, Magdeburg, Germany.

Müller; J., 2002, “Entwicklung einer virtueller Prototypen des hydraulisch angetriebenen Schreitfahrwerks ALDURO”, Fortschr.-Ber. VDI Reihe 1 Nr.356, VDI Verlag, Düsseldorf, Germany.

Oriolo, G., Ulivi, G. and Venditelli, M., 1999, “Real-time map building and navigation for autonomous robots in unknown environments”, IEEE Transactions on Systems, Man and Cybernetics, pp. 100-137.

Risse, W., 2002, “Sensorgestützte Bewegungssteuerung kinematisch redundanter Roboter”, Bericht aus der Robotik, Shaker Verlag, Aachen, Germany.

Sugeno, M. and Kang, G.T., 1988, “Structure identification of Fuzzy model”, Fuzzy sets and Systems, pp. 15-33.

Takagi, T. and Sugeno, M., 1985, “Fuzy identification of systems and applications to modelling and control”, IEEE Transactions on Systems, Man and Cybernetics, pp. 116-132.