Integrating and Modelling Sensors: An Approach to Low-Cost Robotics Systems using Multilayer Perceptrons

ALVARO SANCHEZ-MIRALLES, MIGUEL ANGEL SANZ-BOBI
Instituto de Investigación Tecnológica
Universidad Pontificia Comillas
Alberto Aguilera, 23. 28015 Madrid
SPAIN

Abstract: - In this article we deal with the problem of modelling the sensory behaviour of sensors in a low cost robotics system, modelling the uncertainty of the taken measurements and fusing different measurements, in order to improve the quality of the information the robotics system needs. It is achieved thanks to a sensory model based on multilayer perceptrons, trained off-line. This model not only improves the quality of the information, but also grants independence between hardware and software of the robotics system. Some experiments have been made, using the mobile robot UMI, in order to demonstrate the utility of this model. These experiments show how sensory model corrects ultrasonic reflection effect and sliding of wheels.

Key-Words: - Multilayer Perceptron, Sensor Fusion, Sensory Model, Low-Cost System.

1 Introduction
Robots must operate in a world which is inherently uncertain. This uncertainty arises in the perception and modeling of environments [1]. If robot systems are ever to achieve a degree of intelligence and autonomy, they must be capable of using many different sensors in an active and dynamic manner; to resolve single sensor ambiguity, to discover and interpret their environment [2][3]. This is more difficult to achieve it in a low cost system where sensors are very limited, both in number and quality, and it is very important to squeeze all the information they provide.

Sensor fusion may be treated in deliberative systems as a means of constructing a global world model, such as in [6] using Kalman filters, or may be treated in reactive systems [2].

In this article, a sensory model is proposed in order to model the sensory behaviour of sensors in a low cost multi-sensor robotics system. It could be used both in deliberative systems and in reactive systems. In the first section a global view of this model is introduced placing it in a robotics system. Then we will describe how the model achieves its objectives using an application example. And finally some experiments with a mobile robot are shown.

2 Sensory model
The main objective of the sensory model is to model the sensory behavior of sensors and to grant independence between hardware and software of the robotics system.

This implies two tasks:
- To model the uncertainty of taken measurements.
- To fuse the measurements of several sensor in order to improve the quality of them.

This is very important for environment modeling [4], since the estimated map of the environment depend on sensors measurements.

In the fig. 1 two robotics systems are shown. Each one has both its own sensors and its own sensory model which are different. But in both cases the robot brain has the same structure and uses the same type of variables. The sensory model could be called virtual sensor because it is the only source of information for the robot. Therefore the outputs of the virtual sensor are called virtual measurements.

It consists of several sub-models. Each one has a set of sensor measurements as inputs, and gives a virtual measurement as output of better quality than each of the input measurements. Since there is sensory model, the brain of the robot only uses the virtual measurements, instead of the sensor measurements, as inputs variables. A similar idea is used in the Saphira architecture [5].
3 Description

Suppose a multi-sensory robotics system with a set of sensors \( S_N = \{s_i\}_{i=1}^N \). It could be extracted from them a set of features \( C_1 = \{c_i\}_{i=1}^I \); for example, distance is the feature of ultrasonic rangers, coordinates of the robot are the features of GPS. Given a set of features it could be defined a set of variables \( V_k = \{v_i\}_{i=1}^K \) so that each variable is calculated using some features, \( v_i = f_j(Cv)/Cv \subset C_1 \).

In real world, these functional relationships are not lineal and even discontinue. Therefore we propose to use neural networks, such as multilayer perceptrons (MLP), for modeling those relationships. These networks are trained off-line just when you are calibrating sensors.

As it can be seen in fig 2, the sensory model has a set of features as inputs, and a set of variables as outputs and consists of sub-models. Each sub-model is a multilayer perceptron.

Sensors could be classified in two groups: exteroceptive sensors (such as ultrasonic ranger, laser …) and positioning sensors (such as compass, odometric system…). In the same way sub-models could be classified in two groups: exteroceptive sub-models which deal with exteroceptive sensors, and positioning sub-models which deal with positioning sensors.

3.1 Application example

Next, an application example of the sensory model is described. It is based on the mobile robot called UMI. It is a circular robot with differential cinematic. It has two wheels of 2.5 centimeters wide and consists of 8 ultrasonic rangers (SRF04), 5 infrared rangers (GP2D12), one compass (VECTOR 2X) and an odometric system (one HP encoder in each wheel of the robot); disposed and named as it is shown in fig. 3.

Fig. 3. Layout of the robot

3.1.1 Exteroceptive model

From the eight ultrasonic and the five IR rangers \( S_13 = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, i_1, i_2, i_3, i_4, i_5\} \), we could extract thirteen features \( C_13 = \{du_1, du_2, du_3, du_4, du_5, du_6, du_7, du_8, di_1, di_2, di_3, di_4, di_5\} \). They represent the distance from the sensor to the nearest obstacle, measured in the direction of the sensor. We could define the variables “enhanced distance” to the nearest object measured in the direction –90°, -38°,
0°, 38° and 90°, \( V_5 = \{v_{-90}, v_{-38}, v_0, v_{38}, v_{90}\} \). They are calculated through three sensory sub-models (three multilayer perceptrons), using subsets from \( C_{13} \) as follows:

\[
\begin{align*}
v_{-90} &= MLP_{-90}(du_5, di_1) \\
v_{-38} &= MLP_{-38}(du_7, du_6, du_4) \\
v_0 &= MLP_0(du_4, du_5, du_3) \\
v_{38} &= MLP_{38}(du_2, du_3, du_4) \\
v_{90} &= MLP_{90}(du_1, di_3)
\end{align*}
\]

There are sub-models which fuse features from ultrasonic sensors only, such as MLP_{-38} and MLP_{38}, and other sub-models which fuse features mixed from ultrasonic and IR sensors, such as MLP_{-90}, MLP_0 and MLP_{90}. All of them try to correct some limitations of the sensors such as measurement angle, reflections, measurement error and errors due to the incidence angle of the ultrasonic wave.

Starting from these variables, it is very easy to calculate the coordinates of obstacles points.

3.1.2 Positioning model

From the encoders and the compass \( S_e = \{e_1, e_2, c\} \), we could deduce three features \( C_3 = \{em_1, em_2, cm\} \) and, in this case, they are sensor measurements. We could define variables which represent the increment in the position of the robot \( V_3 = \{\Delta x_r, \Delta y_r, \Delta \theta_r\} \). They are calculated through three sensory sub-models, using subsets from \( C_3 \) as follows:

\[
\begin{align*}
\Delta x_r &= MLP_{3x}(em_1, em_2) \\
\Delta y_r &= MLP_{3y}(em_1, em_2) \\
\Delta \theta_r &= MLP_{3\theta}(em_1, em_2)
\end{align*}
\]

These neural networks model the diameter of each wheel and the diameter of the robot. These could be obvious but it isn’t. The effective diameter of each wheel is slightly different from the one we could measure (depend on the distribution of the load in the robot, and on the material of the wheels), but anyway it is constant. The effective diameter of the robot varies dynamically depending on the displacement of each wheel, since the wheel has a specific width (as much thinner the wheel is less varies the effective diameter of the robot). This is due to wheels are rigid solids and each point of the wheel in contact with the ground can not have the same speed (except in a pure translation), so the instantaneous rotation center IRC of each wheel is different depending on the displacement of the wheels (the effective diameter of the robot is the difference between the IRC of each wheel). This is one of the causes that make wheels to slide, so this model corrects it.

A variable could be a feature (a feature of a sub-model) and so an input of a sub-model. For example, another sub-model that could be included is:

\[
\theta_r(t) = MLP(cm, \theta_r(t - 1) + \Delta \theta_r)
\]

It uses the feature extracted from the compass (it is an absolute measurement) and the variable \( \Delta \theta_r \) (it is an incremental measurement).

4 Experimental results

Some experiments have been done using the robot and based on the sensory model described in the application example of previous section. We have used an environment for network training and another environment for sensory model validation; both of them are shown in fig. 4 and fig. 5 respectively. We will study two cases. In the first case we will validate positioning model, and in the second case the exteroceptive model will be validated. For easier understanding the advantages of the sensory model, in both cases we compare the predicted environment with and without sensory model.

Fig.4. Artificial environment (training)
4.1 Positioning model

Thanks to positioning models, the predicted position of the robot could be calculated, if the initial position of the robot is known. In order to demonstrate the utility of this model, it will be compared with the results obtained using the measurements of the wheel encoders after calibrate them. In next figures, in red blue color are represented some points of obstacles of the environment, and in green color are represented the estimated points of obstacles of the environment.

Fig. 6. Estimated environment using the measurements of the calibrated sensors of the mobile robot.

Fig. 7. Estimated environment applying positioning models MLPx, MLPy and MPLθ.

In fig. 7 is showed the estimated environment after applying the positioning models MLPx, MLPy and MPLθ, as it was previously commented. It can be seen how the models reduce the error due to the slippery of wheels.

Finally, in Fig. 8 is showed the estimated environment after applying all positioning models, MLPx, MLPy, MPLθ and MPL. In this case the compass is very important, because it improves orientation errors. There is still error in the predicted position of the mobile robot, but it is small and it could be corrected using relocalization algorithms based on previous knowledge of the environment.

Fig. 8. Estimated environment applying MLPx, MLPy, MPLθ and MPL.}

4.2 Exteroceptive model

Next sub-model MLPz is described in detail, after applying positioning models. As a result of this sub-model the “enhanced distance” to the nearest object
measured in the direction -38º is calculated. In order to demonstrate the utility of this sub-model, it will be compared with the results obtained using the measurements of sensor u6, after a little of processing for eliminating some spurious measurements.

![Fig. 9. Training of MPL-38](image)

![Fig. 10. Validation of MPL-38](image)

In figure 9 and 10, in red blue color are represented the theoretical points obtained with ideal sensors, in green color are represented the points obtained with measurements of sensor u6, and finally in red color are represented the points obtained using sub-model MLP-38. As it can be seen the error produced by reflection is eliminated and it is reduced the mean square error from 3.1 cm² to 0.18 cm² in the validation environment.

Finally, in fig. 11 is showed the final predicted environment after applying sensory models.

![Fig. 11. Estimated environment applying positioning models and exteroceptive models.](image)

5 Conclusion

In low cost multi-sensor systems, it is very important to take advantage of the overall sensory information, fusing measurements and modeling sensors. In this article has been presented a new method to do it using neural networks. Besides, this method is able to achieve independence between hardware and software of robotics systems, making easy to develop a multi-agent architecture involving several different robots.

Finally, some of the experiments are shown to demonstrate the utility of the sensory model. It improves ultrasonic reflections, slippery of wheels and encoder errors.

References:


