COMPLEXITY PROBLEM OF POTENTIAL BASED GUIDING

M. Visontai * Sz. Szabó * A.R. Várkonyi-Kóczy * P. Baranyi **
G. Samu * L. Kiss *

* *Department of Measurement and Information Systems, Budapest University of Technology and Economics*
** *Department of Telecommunications and Telematics, Budapest University of Technology and Economics*

Abstract: The development of techniques for autonomous mobile robot navigation has been in focus for several decades. The main objective of this paper is to present a neuro-fuzzy algorithm. This algorithm can approximate the basic guiding models. In this paper the approximation of the vector field model is shown. The vector field is an extension of the potential based method. With the help of this neuro-fuzzy technique we can reduce the calculation complexity of the guiding algorithm. *Copyright ©2000 IFAC*

Keywords: autonomous robots, guiding model, neuro-fuzzy systems

1. INTRODUCTION

In the recent decades the control of mobile robots has been in the spotlight. Several works have been published in the theme of controlling robots, which interact with a dynamic, uncertain environment (Brooks, 1991; Arkin, 1989; Lammens *et al.*, 1993). This paper has two objectives. One is to select a widespread and promising technique from the literature which is applicable for basic guiding model. The second is to find a simple engineering tool, namely an algorithm, which is able to approximate all kinds of basic guiding styles and to reduce their computational complexity.

The potential field based guiding is a widely used guiding model (Latombe, 1991). It is a classical technique in robot motion planning due to its simplicity and feasability for real-time control (Bronstein and Koren, 1989). Many robot navigation and guiding control algorithms, especially for obstacle avoidance, are developed based on the potential field method (Bronstein and Koren, 1991), or its variants and extensions (Kubota *et al.*, 1999). The fundamental concept behind the potential based model is that virtual repulsive and attractive forces are acting on the robot. Steering direction and velocity is then calculated based on the resultant virtual force acting on the robot. Despite the advantages of PBG, its guiding style modeling property is somewhat limited. For example, when a robot is traveling in a narrow corridor, the path generated by the potential method oscillates around the centreline of the corridor. This paper proposes an extension to PBG - the vector field based guiding (VFB) model - that eliminates the mentioned disadvantage.

As a second aim this paper proposes a simplified general neuro-fuzzy algorithm capable of approximating both the PBG and VFB models, which can easily be implemented. The ability of using the knowledge (learnt in previous situations) in unpredictable environments is an essential component of many recent publications (Kubota *et al.*, 1999; Fukuda and Kobayashi, 1999; Piaggio and Zaccaria, 1997). This general form ensures that the robot can learn the guiding style of other robots. They can learn even from human guiding.

The advantage of applying fuzzy logic or neural networks lies in their ability to imitate and implement the actions of expert operators without the need of accurate mathematical models. The drawback, however, is that there is no standardized framework regarding the design, optimality, reducibility, and definition of the concept. These algorithms, either generated by expert operators, or by some learning or identification schemes, may contain redundant, weakly contributing, or unnecessary components. In order to achieve a good approximation, some approaches may overestimate the number of neural networks' hidden units and layers. This large neural network causes problems in computational complexity. Fuzzy techniques are popular engineering tools, however, they still suffer from exponential calculation-time and storage-space complexity which has been demonstrated through the uniform complexity expression of (Kóczy and Hirota, 1990), namely that the number of rules grows exponentially with the number of antecedent fuzzy terms and the number of input variables (Kóczy and Hirota, 1997).

This paper proposes a simplified neuro-fuzzy technique for guiding models. A calculational complexity reduction technique is also proposed to compress the trained neuro-fuzzy models. A neural network or fuzzy rule base design is, therefore, required to consider two important contradictory objectives. One is to achieve a good approximation, while the other is to reduce the calculational complexity.

2. ARTIFICIAL POTENTIAL FIELD BASED GUIDING

This section introduces the potential field based guiding model. The main rule of PBG is to repulse (or attract) the robot (Arkin, 1989), from (or to) the obstacles. The objects and the target generate imaginary forces that act on the robot. These virtual forces are then summarized, resulting in the desired moving direction. The virtual vectors must be calculated for each location as quickly as possible to result in a smooth and reactive guiding.

The magnitude of the repulsive forces is usually inversely proportional to the distance between the obstacle and the vehicle, e.g. (Bronstein and Koren, 1989):

$$|y| = \frac{1}{(distance)^a} \qquad (1)$$

where a is constant, generally greater than one. As a result the potential functions are usually strongly nonlinear.

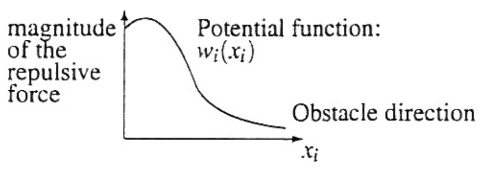

Fig. 1. Potential function

In many applications the same formula is used to compute all the repulsive forces (Bronstein and Koren, 1991), resulting in a symmetrical potential field. In order to achieve different guiding styles asymmetrical potential fields could be used, as well. A simple guiding example is the basic rule of traffic. The vehicles should keep to the right: therefore they must keep close to the obstacles on the right side while staying far from the objects on the left.

In most cases barriers cause repulsive forces, however there are some examples where attractive object forces are desired. For instance when the robot must run close to a wall (which is the usual situation when the robots are used in corridors), this obstacle must attract the robot when the robot is too far from it. Potential functions having negative sections can achieve this kind of styles.

3. POTENTIAL FIELD MODEL

Let us assume that the robot's field of view is divided into scanned directions. The sensor system provides the distances (x_i) between the robot and the obstacle in each scanned direction. The scanned directions are virtually processed from three- and/or two-dimensional pictures generated by either onboard cameras or sensors.

For each scanned direction, the potential function has to be evaluated in order to acquire the virtual forces. Let us suppose that n scanned directions are present. For each scanned line the vectors are calculated as:

$$\bar{y}_z = \bar{e}_z w_z(x_z) \qquad (2)$$

where $i = 1..n$, x_z is the measured distance in the scanned direction $\bar{e}_z$ and $w_z(x_z)$ is the z-th potential function. The virtual forces ($\bar{y}_z$) point into the opposite direction of $\bar{e}_z$.

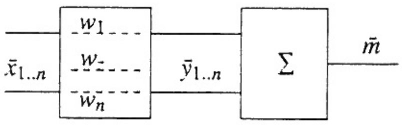

Fig. 2. Potential Based Guiding control

The algorithm should preserve low computational complexity in order to maintain the applicabality in real-time applications. Thus, some approximation technique is highly desired. Considering that the potential functions are usually strongly non-linear, soft-computing methods are to be used to approximate the proposed method.

4. DEFINITION OF BASIC ROBOT GUIDING

The task of the guiding is to define the moving direction according to the chosen style and the scanned area. Various moving styles can be implemented, as an example, let us consider two extreme styles.

discussed potential field based guiding to a more general model which can define the arbitrary moving directions.

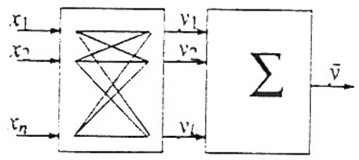

Fig. 5. Vector Field Based model

Evidently, if one scanned direction has contribution not only to itself, but to some other scanned directions, as well, other moving directions can easily be defined.

The generalization of the potential based guiding results in a structure where all inputs are connected to all outputs. Consequently, the key difference from the PBG is that all inputs of the first block have contribution to all outputs connected to the evaluation unit (Fig. 5). Obviously, the potential based model is a special case of the generalized model:

$$w_{j,i} = 0, \text{ for each } i \neq j.$$

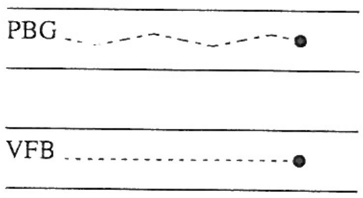

Fig. 6. Path by the PBG and the VFB in a narrow corridor

As shown in Fig. 6 the vector field based guiding eliminates the oscillatory and unstable motion of the PBG, because it can simply assign the steering direction to be parallel to the wall of the corridor.

8. NEURAL NETWORK APPROXIMATION OF THE VFB

This section introduces a generalized neural network that forms the vector field described in the above section. Fig. 5 represents two layers of a generalized feedforward neural network which is general in the sense that the neurons have no transfer functions, but it has various weighting functions ($f_{z,i}(x_i) = w_{z,i}(x_i)$) instead of weights among the layers. The output of one neuron yields:

$$\bar{y}_z = \bar{e}_z \sum_{i=1}^{n} w_{z,i}(x_i) \tag{5}$$

The second block applies the third, linear layer, which sums up the output of the neurons of the second layer, generating the algorithm's output:

$$\bar{y} = \sum_{z=1}^{k} \bar{y}_z = \sum_{z=1}^{k} (\bar{e}_z \sum_{i=1}^{n} w_{z,i}(x_i)) \tag{6}$$

In some cases it is advantageous if some inputs have a more significant effect during the guiding. For instance scanned directions (sensors) at the front are usually more important than the ones at the back. It is possible to set the significance of the vector $\bar{y}_z$ by multiplying it with a constant value c_z. Obviously this multiplication gives an offset to all weight functions connected to the z-th output. Figure 7 shows a hybrid neural architecture constructed with a non-linear and a linear layer.

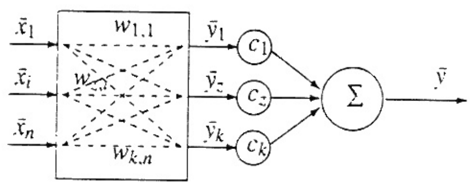

Fig. 7. Hybrid network

The output of the controller is:

$$\bar{y} = \sum_{z=1}^{k} c_z \cdot \bar{e}_z \cdot \sum_{i=1}^{n} w_{i,j}(x_i) \tag{7}$$

Let us approximate the weighting functions by the product-sum-gravity fuzzy algorithm yielding a neuro-fuzzy system (Brown and Harris, 1994). We get the final form of the output as:

$$\bar{y} = \sum_{z=1}^{k} c_z \sum_{i=1}^{n} \sum_{j=1}^{m} \mu_{A_{i,j}}(x_i) \bar{b}_{z,i,j} \tag{8}$$

9. COMPUTATIONAL COMPLEXITY OF THE GENERAL NEURAL NETWORK

Let us characterize the computational effort of (8) by the number of product operations and omit the calculation effort of the add operation.

Lemma 1. The computation requirement of (8) is:

$$P_c = n \cdot k \cdot m + P_\mu \tag{9}$$

where P_μ indicates the calculation complexity of the membership functions.

Lemma 2. If (8) can be transformed into the form of

$$\bar{y} = \sum_{z=1}^{k^r} c_z \sum_{i=1}^{n} \sum_{j=1}^{m^r} \mu_{A^r_{i,j}}(x_i) b^r_{z,i,j} \tag{10}$$

where $k^r \leq k-1$, $m^r \leq m$ (r stands for reduced), then the computation requirement of (10) is less than (8).

Theorem. The equation (8) can always be transformed into the form of (10).

The main guiding rule of a plane carrying dangerous material is to keep "as far from the mountains as possible". Remaining in secret while seeking a mouse leads to the opposite behaviour for a cat, namely "get as close to the objects as possible". A simple combination of these pseudo-rules can characterize the main rule of a traffic system: "keep close to the right side". Although, the basic robot guiding has no problem solving ability. For example, finding a path in a labyrinth needs additional intelligence.

5. FUZZY APPROXIMATION OF THE POTENTIAL FIELD

In this section we introduce the product-sum-gravity (PSG) based fuzzy technique in order to approximate the PBG model. The proposed fuzzy rule base is specialized in the sense that the number of rules does not grow exponentially by the number of inputs. This specialization is offered by the fact that the potential surface is a three-dimensional surface independent of the number of scanned lines, hence, the number of inputs.

According to Fig. 2, the inputs of the fuzzy rule base are distances, the algorithm results in values y_z that are multiplied by the unique vector $\bar{e}_z$ pointed to the opposite of the scanned directions.

Algorithm 1: Product-sum-gravity

Antecedents: The antecedent fuzzy sets $A_{i,j} : \mu_{A_{i,j}}(x_i), i = 1..n, j = 1..m, x_i \in X_i$ are defined on each input universe X_i in Ruspini-partition.

Consequents: Let us define consequent vectors as B_j : $\mu_{B_j}(y) = \delta(b_j)$, where Y_j is the j-th output universe.

Observation: Let the input value x_i be fuzzificated into the singleton observation fuzzy set A_i^* such as: $\mu_{A_i^*}(x_i) = \delta(x_i)$

Characterization of the rules: The number of rules obtained by all combinations of the antecedents grows exponentially with the number of antecedents and inputs (Kóczy and Hirota, 1997). The potential function, however, is a 3-dimensional surface. This implies that the co-relation of the inputs can be omitted. Therefore let us define the rules such as: IF $A_{i,j}$ THEN $B_{i,j}$ Consequently the number of rules is only $n \cdot m$.

Inference method: The fuzzy inference is based on product-sum-gravity (Mizumoto, 1990).

product: This step yields in the contribution of the consequent sets to the outputs according to the degree of matching among the observation and antecedent sets. The product has a role in the case of co-related rules. Therefore the contribution is: $\mu_{A_{i,j}}(x_i)$.

sum-gravity: This step follows the center of gravity defuzzification technique. All consequent sets are weighted by their corresponding contribution and summarized to produce the output vector $\bar{y}_i$:

$$\bar{y}_i = \frac{\sum_j \mu_{A_{i,j}} b_{i,j}}{\sum_j \mu_{A_{i,j}}} \qquad (3)$$

The antecedent sets are defined in Ruspini-partition, which implies that the denominator of (3) equals to 1, consequently:

$$\bar{y}_i = \sum_j \mu_{A_{i,j}} b_{i,j} \qquad (4)$$

6. LIMITATIONS OF THE POTENTIAL FIELD METHOD

The key idea of the potential field method is to repulse (or attract) the robot from (or to) objects (depending on the scanned direction's potential function). Obviously in many cases this rule is not effective enough. For instance if the vehicle must avoid an obstacle which appeared in front of it, better results would be obtained if the guiding model showed the direction in which the robot should bypass the object (instead of just repulsing it).

The potential based guiding also allows the robot to get in a "stuck" position. As shown in Fig. 3 the helicopter is surrounded by objects that are placed symmetrically and the distances are equal. In this case the evaluation will result in the zero vector, leaving the helicopter in stuck position. This requires other types of solutions, for example: choosing one direction at random, or choosing the nearest one to the target.

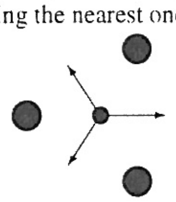

Fig. 3. Stuck position

Another problem is presented in fig. 4. The robot has to run parallel with a long wall. The required vectors (or their sum) must be parallel with the wall. However the potential field method is not able to generate such vectors.

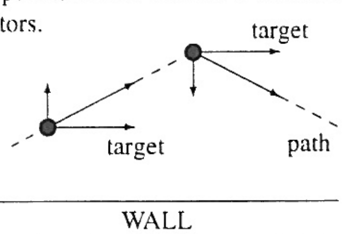

Fig. 4. Guiding fluctuation of the PBG model

7. GENERALIZING THE POTENTIAL FIELD TO VECTOR FIELD

Considering the previous section, it would be desirable to define arbitrary moving directions in every point of the potential function on each scanned line. For this purpose this section extends the previously

The transformation can be done with the use of SVD - Singular Value Decomposition (Yam *et al.*, 1999).

10. EXAMPLES

In order to show the effectiveness of the specialized fuzzy logic technique proposed for PBG we implemented various PBG-based techniques chosen from the literature and used them as "teacher" robots. A simple learning technique was implemented in a "student" robot. This learning technique is based on the general neuro-fuzzy approximation, the strongly nonlinear weighting functions are approximated by linear ones. We concluded that the trained "student" robot showed the same guiding style and found the path with small difference as the teacher did, even in case of a new set of obstacles.

In order to save computation complexity we used simplified triangular fuzzy sets defined in Ruspini-partition. We set seven sets at each dimension that implies a rough approximation, however the guiding styles of the "teacher" and the "student" robot had no remarkable difference. As an other example we controlled the "teacher" robot manually.

Let us present only three extreme results. The basic guiding styles of the manual control were: 1) keep as close as possible to the left side; 2) keep to the right; 3) get as far from the objects as possible.

Figure 8 shows the guiding of the three trained "student" robots among a new set of objects. We concluded that the robot was able to pick up the human guiding styles.

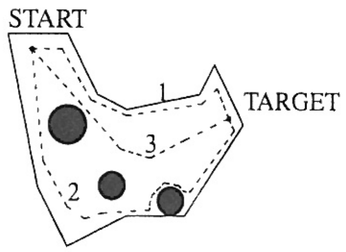

Fig. 8. Extreme guiding styles learned from humans

11. CONCLUSION

This paper proposed an extension of the potential based guiding model. The extended model eliminates the strongly alternating behavior of PBG. A simplified neuro-fuzzy algorithm is also presented to approximate the new model. Both neural network and fuzzy logic techniques and the robot guiding have problems with calculation complexity in many cases. This paper proposes a complexity reduction technique applicable to neuro-fuzzy algorithms. Some examples were presented to demonstrate the effectiveness of the algorithms.

12. REFERENCES

Arkin, R.C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotics Research* **8**, 92–112.

Bronstein, J. and Y. Koren (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics* **19**, 1179–1187.

Bronstein, J. and Y. Koren (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Jour. Robotics and Automation* **7**, 278–288.

Brooks, R.A. (1991). Intelligence without reason. *Massachusetts Institute of Technology, A.I. Memo No. 1293.*

Brown, M. and C. Harris (1994). *Neurofuzzy Adaptive Modeling and Control*. Prentice-Hall, New York.

Fukuda, T. and F. Kobayashi (1999). Adaptation, learning and evolution for intelligent systems. *IEEE Int. Fuzzy Systems Conf. Proc.* **3**, 1203–1210.

Kóczy, L.T. and K. Hirota (1990). Fuzzy interference by compact rules. *Proc. of Int. Conf. on FL & NN.*

Kóczy, L.T. and K. Hirota (1997). Size reduction by interpolation in fuzzy rule bases. *IEEE Tr. S.M.C.* **SMC-27**, 14–25.

Kubota, N., T. Morioka, F. Kojima and T. Fukuda (1999). Sensory network for fuzzy controller of a mobile robot. *IEEE Int. Fuzzy Systems Conf. Proc.* **2**, 986–991.

Lammens, J., H. Hexmoor and S. Shapiro (1993). Of elephants and men. *In Proceedings of the NATO-ASI on the Biology and Technology of Intelligent Autonomous Agents.*

Latombe, J. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

Mizumoto, M. (1990). Fuzzy controls by product-sum-gravity method. In: *Advancement of Fuzzy Theory and Systems in China and Japan* (Liu and Mizomoto, Eds.). Chap. 1.1–1.4. International Academic Publishers.

Piaggio, M. and R. Zaccaria (1997). An autonomous system for vehicle navigating in a partially or totally unknown enviroment. *Proc. Int. Workshop on Mechatronical Computer Systems for Perception and Action, MCPA.*

Yam, Y., P. Baranyi and C.T. Yang (1999). Reduction of fuzzy rule base via singular value decomposition. *IEEE Trans. on Fuzzy Systems* **7**, 120–132.