NEURAL NETWORK BASED MOBILE ROBOT NAVIGATION

István ENGEDY
Advisor: Gábor HORVÁTH

I. Introduction

The navigation system of a mobile robot must handle numerous well separable subtasks for proper operation. One of these tasks is motion planning, which is divided into at least two separate parts, the path planning and the movement itself. The first is the procedure, which designs the path from the current location to the end point, of which the robot will have to go through. The movement is the procedure of keeping the robot on this path. The motion could be complex [1], because the movement of the robot must meet various physical constraints, like the momentum of the car.

The motion planning could be carried out in other ways too. Using soft computing methods, the path planning and the movement can be carried out simultaneously [2]. In this paper we will present an artificial neural network based robot navigation solution, which could avoid moving obstacles.

II. Navigation method

In our motion planning solution the mobile robot is controlled by an artificial neural network (ANN). It is trained with the backpropagation through time method (BPTT) [3], which is a well known training algorithm of dynamic feedback ANNs. Its use for robot navigation has been already shown by D. Nguyen and B. Widrow [4]. The main idea behind this is to open the feedback control loop and unfold it through many iteration steps, thus making a simple feed-forward system, which can be trained with the usual backpropagation algorithm (Fig. 1. a).

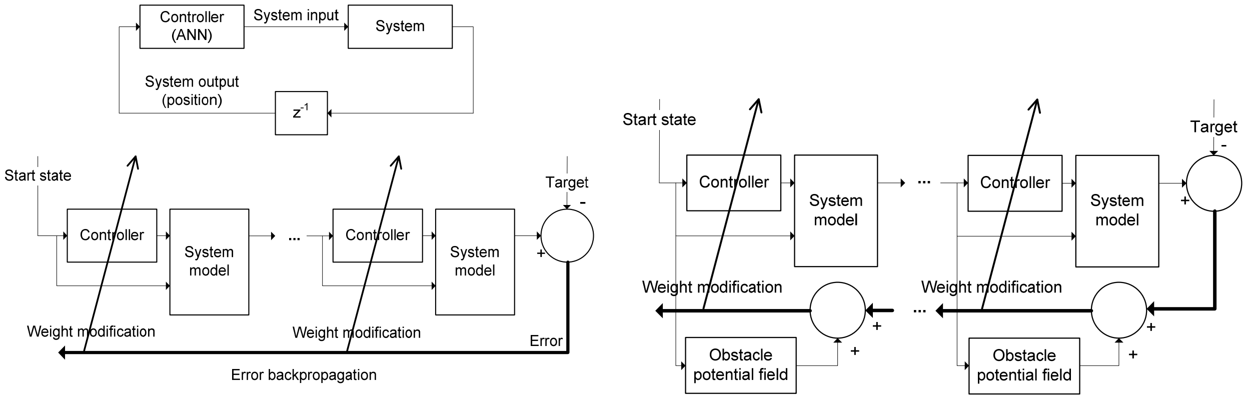

Figure 1: (a.) BPTT in use of training in a control loop (b.) Regularizing the BPTT

This method is able to navigate a mobile robot from any starting point on the working area to any target point. The path and the motion planning are done simultaneously, as the constraints of the robot are taken into account during the backpropagation through the system model. This way the controller is trained to follow a path, with the robot controlling commands calculated. This method however is not able to avoid obstacles, especially not moving obstacles.

III. Obstacles

To make the BPTT method able to handle obstacles, we have elaborated the following solution. Based on the location and the size of the obstacles, a potential function can be defined (Eq. (1) left), which is used to repel the robot away from the obstacles. This function must be used to extend the cost function of the Delta-rule. The cost function is regularized with the potential field (Eq. (1) right and Fig. 1. b), so the goal of the weight modification is not only to minimize the error at the end of the simulation chain, but also to minimize the potential of the path, to get the robot the farthest from the obstacles. This way the obstacles could be avoided. The potential field and the regularization of the cost function are defined as follows:

$$U_i(y)=\begin{cases}\dfrac{1}{\left(d_i(y)-r_i\right)^2}, d_i(y)>r_i+\varepsilon\\ \dfrac{1+r_i-d_i(y)}{\varepsilon^2}, otherwise\end{cases} \qquad C_R(t,y)=\|t-y_n\|^2+\lambda\sum_{i=0}^{n}\sum_{j=0}^{k}U_j(y_i) \qquad (1)$$

where $U_i(y)$ is the potential field of the i^{th} obstacle, $d_i(y)$ is the distance from the centre of the obstacle, r_i is the radius of the obstacle, y is the position of the robot, and ε is a small positive constant. $C_R(t,y)$ is the regularized cost function, t is the position of the target, n is the number of iteration steps during BPTT, and k is the number of obstacles.

To use this method to navigate among moving or previously unknown obstacles, further modifications must be made. Till this point, we did not specify, whether the ANN is to be trained offline or online. Offline training could be used, if the obstacles were previously known and static. In such cases the training could be carried out from multiple starting points. In case of moving, or previously unknown obstacles, the use of online training seems to be the only option, despite its trivial drawback: the increased need of computational power.

Online training brings the ability to adapt to changing environment, e.g. avoid moving or to previously unknown obstacles [5]. It has also many advantages. There is no need for training the ANN from multiple starting points, only the current location of the robot should be used, which makes the training much faster. Using the online training makes the method to an anytime algorithm between reasonable limits, because the navigation result is degraded only in quality with the decreasing time limit until the ANN training becomes insufficient. On the other hand this makes the algorithm well-scalable; using a faster CPU can increase the quality of the result of the algorithm.

IV. Conclusion

In this paper we have shown how the classical BPTT training approach can be extended using regularization, to take additional constraint into consideration, like obstacles. Simulations and real robot experiments have proved that using real-time online training, this method is able to handle moving obstacles as well.

Acknowledgement

The authors gratefully acknowledge the support of the Hungarian Fund for Scientific Research (OTKA), Grant #73496.

References

[1] J.-C. Latombe, *Robot Motion Planning,* Kluwer, 1991.

[2] Pratihar D.K., Algorithmic and soft computing approaches to robot motion planning, *Machine Intelligence and Robotic Control*, 5(1):1-16, 2003

[3] M. Minsky and S. Papert. *Perceptrons, An Introduction to Computational Geometry,* MIT Press, 1969

[4] D. Nguyen and B. Widrow, “The Truck Backer-Upper: An Example of Self-Learning in Neural Networks,” in *Proc. of the International Joint Conference on Neural Networks*, pp. 2:357-362, 1989.

[5] I. Engedy and G. Horváth, “Artificial Neural Network based Mobile Robot Navigation,” in *Proc. of the IEEE International Symposium on Intelligent Signal Processing*, pp. 241-246 ,Budapest, Hungary, Aug. 2009