17th International Conference
ENGINEERING MECHANICS 2011
Svratka, Czech Republic, 9 – 12 May 2011

MOBILE ROBOT MOTION PLANNER VIA NEURAL NETWORK

J. Krejsa*, S. Věchet**

Abstract: *Motion planning is essential for mobile robot successful navigation. There are many algorithms for motion planning under various constraints. However, in some cases the human can still do a better job, therefore it would be advantageous to create a planner based on data gathered from the robot simulation when humans do the planning. The paper presents the method of using the neural network to transfer the previously gained knowledge into the machine learning based planner. In particular the neural network task is to mimic the planner based on finite state machine. The tests proved that neural network can successfully learn to navigate in constrained environment.*

Keywords: ***Mobile robot, motion planning, neural networks.***

1. Introduction

Navigation of mobile robots usually incorporates several modules among which the path planning and position estimation are the most important ones. There is a vast number of approaches for path planning in both constrained and unconstrained environment. Most of the planning algorithms do not take into account dynamic obstacles or when they do, low number of obstacles is expected. The usual solution is to detect the dynamic obstacle and predict its motion thus creating another constrain in the configuration space.

In our application the safe path planner controls the robot in indoor environment with dynamic obstacles. There is high number of obstacles and their motion is hard to predict. This excludes most of the planning algorithms that calculate the whole track to the goal as unpredictable movement can create constraint that suppress validity of the path proposed. Further issues to consider in the application are: the optimality of the path is not required, available computational power is limited. Therefore some kind of semireactive navigation must be used, that is fast enough but does not ignore the goal position. That lead to idea to use complex controller, gather the data and use machine learning to mimic (approximate) such controller in a fast manner. Neural networks (NN) are a natural choice of machine learning engine for such a case. In (Yang, 2003, Janglová, 2004, Sirotenko, 2006) the NN is used for mobile robot navigation based on proximity sensors values. Our proposed method also uses the proximity sensors values, but additionally incorporates the past values of planner output as NN inputs. Such approach enables NN to learn the internal state of the planner and solve the navigation problem when the nonholonomic constraints require the robot to backup.

The paper describes proposed method on the case, when motion planner based on finite state machine (FSM) with internal state was used to generate the data for feedforward neural network. The network is then trained and further used as the motion planner.

2. Methods

In order to use neural network based planner, the data must be gathered first for NN training. While our ultimate goal is to use observations of human controlling the robot, in the first attempts we used the planner currently used in our robot that is based on finite state machine with internal state. The scheme of the simulation data flow is shown in Fig. 1.

* Ing. Jiří Krejsa, PhD.: Institute of Thermomechanics AS CR, v.v.i., Brno branch, Technická 2, 616 69, Brno, CZ, email: krejsa@fme.vutbr.cz

** Ing. Stanislav Věchet, PhD.: Faculty of Mechanical Engineering, Brno University of Technology, Technická 2, 616 69, Brno, CZ, , e-mail:vechet.s@fme.vutbr.cz

The planner receives the array of ultrasound sensors proximity measurements, the estimate of the robot position (x and y coordinates and heading angle in global coordinate system) and the position of the goal. The estimate of robot position is obtained from the estimator that implements extended Kalman filter. The output from the planner consists of generalized velocities (translation and rotation) that are fed into robot simulation module in the case of simulation or to the action converter in the case of real robot, where the velocities are recalculated into main drive velocity and front wheels steering angle. Those actions are then fed into the actuator controllers. The estimator receives the action (velocities) and measurement of static beacons placed in known positions. Extended Kalman filter produces the estimate of the robot position, further used in the planner.

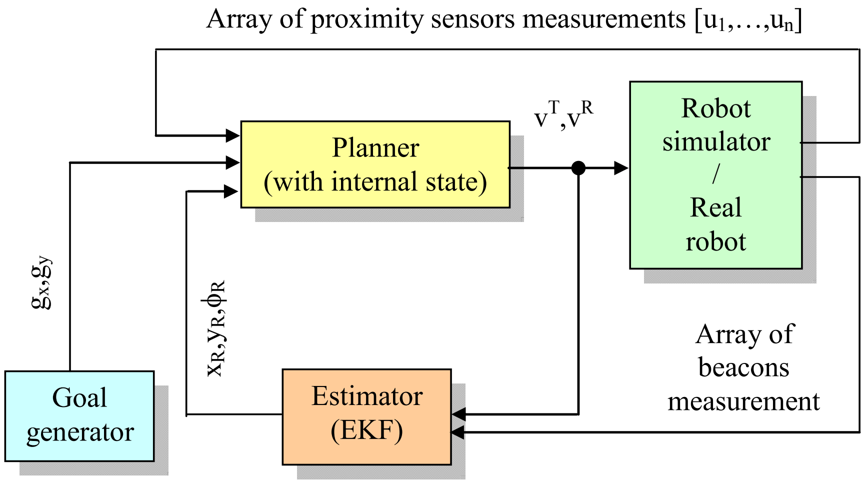

Fig. 1: Gathering data for NN training.

Neural network based planner uses feed forward NN to approximate the planner to be learned. The feed forward NN does not contain the feedback connections, however, the feedback is required as the original planner contains the internal state. Therefore the delayed outputs (generalized velocities) are used as additional inputs to the network. Furthermore the recalculation of the angle towards the goal position is also performed before the data are fed into the network. To summarize the inputs to the NN, it consists of following variables;

$$NNinp = \left[u_1, \ldots, u_n, \Delta\varphi_g, v_{k-1}^T, v_{k-1}^R, \ldots, v_{k-m}^T, v_{k-m}^R, \right]^T \qquad (1)$$

where n is number of proximity sensors, u_i is i-th proximity sensor measurement, $\Delta\varphi_g$ is the angle towards the goal, v_{k-1}^T, v_{k-1}^R are the translational and rotational velocities delayed for a single step delay and m is maximum number of delays used.

The angle towards the goal is calculated from the estimated position, the proximity measurement is in simulation disturbed with the Gaussian noise with experimentally found characteristics. Once the data are gathered and NN is trained to sufficient level of precision, it can be incorporated into the robot navigation structure. The overall scheme is shown in Fig. 2.

3. Numerical experiments details

The robot used for experiments is test wheeled robot with driven back wheels and Ackerman steering. Robot motion is modeled in discrete time steps. The motion model incorporates the noise on front wheels IRC sensors used to calculate the estimate of odometry readings. The localization of the robot is obtained via extended Kalman filter that uses the actions given by the planner, motion model and readings of beacon receiver that gives the relative angle to beacons placed in fixed known positions. Details about the position estimator can be found e.g. in (Krejsa, 2010).

The data used for training were obtained from a single experiment where the robot was driven by FSM based planner to the goals sequentially regenerated once the robot approached to the given distance to the current goal. The goals were generated randomly in a simulated room of H shape with overall size of 20x20 meters. The training run included dynamic obstacles of various sizes moving in various, but constant velocities. The translation velocity of the robot was limited to 0.6m/s and data were calculated in time step of 0.5 sec. Total of 2351 sets of data were gathered and preprocessed for the training.

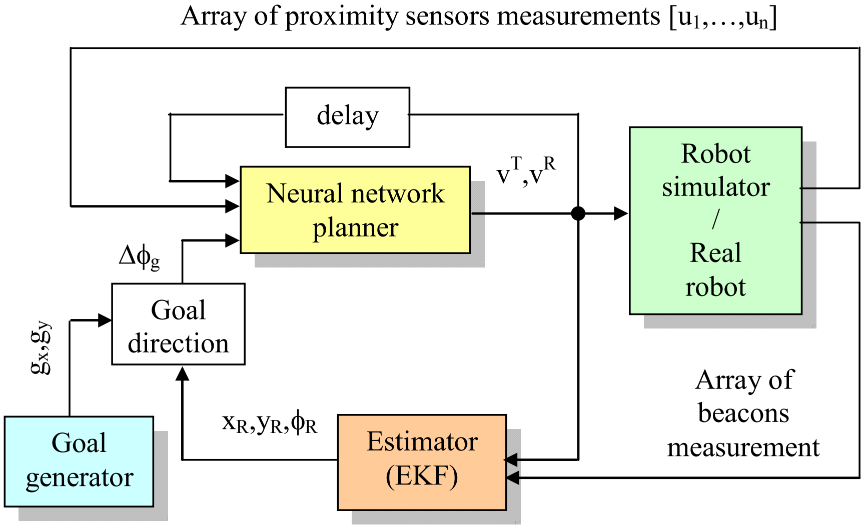

Fig. 2: NN used as motion planner.

Neural network was trained off-line with traditional Levenberg-Marquardt algorithm implemented in Matlab Neural Network Toolbox. Initially the number of past actions fed back as the input to the network (parameter *m* from (1)) was set to 1. The learning MSE stayed at about $1.26e^{-3}$ and when NN was used in navigation process it exhibited occasional unstable behavior. Therefore *m* was increased to 2, which proved to be sufficient. The learning MSE quickly dropped to $2.31e^{-4}$. The first impression of the quality of the training was obtained by testing the network on previously unseen set of test data gathered by the same process as the training data. The comparison of network output and known outputs are shown in Fig. 3.

For further testing the learned network was used in simulation of the whole navigation process. As the dynamic obstacles move in random directions, NN planner can not be directly compared with its predecessor. In order to determine successfulness of the planner the number of simulations were run and results were expressed in two simple parameters: number of steps required to reach the goal and number hits to the obstacles (both static and dynamic). Regarding both parameters the methods are directly comparable, see Tab. 1.

Tab. 1: FSM and NN planners comparison.

Planner	***hits / 1000 steps***	***average path length***
FSM based	0.272	95.4 [m]
NN based	0.261	98.2 [m]

4. Conclusions

Simulation results proved that neural network based planner is capable of learning required behavior. The main advantage of using the NN planner is the speed and low computational requirements. The drawback might be in inability to guarantee safe planning in all times, therefore certain safety measure module must be incorporated that will stop the robot when proximity sensors detect distances below safety threshold. Such system is already implemented and used in real robot.

NN planner with the feedback of previously generated actions used as additional inputs enables the planner to consider the state of the robot, enabling the use of such planner under nonholonomic constraints when backing up is necessary to meet the goal.

Regarding future work, the next step is in gathering the data from simulator when human will be used to control the robot based on the robot sensor information available. Whether the NN planner trained on such data is capable of planning in hard to solve situation (typically overcrowded space) is the question we would like to answer.

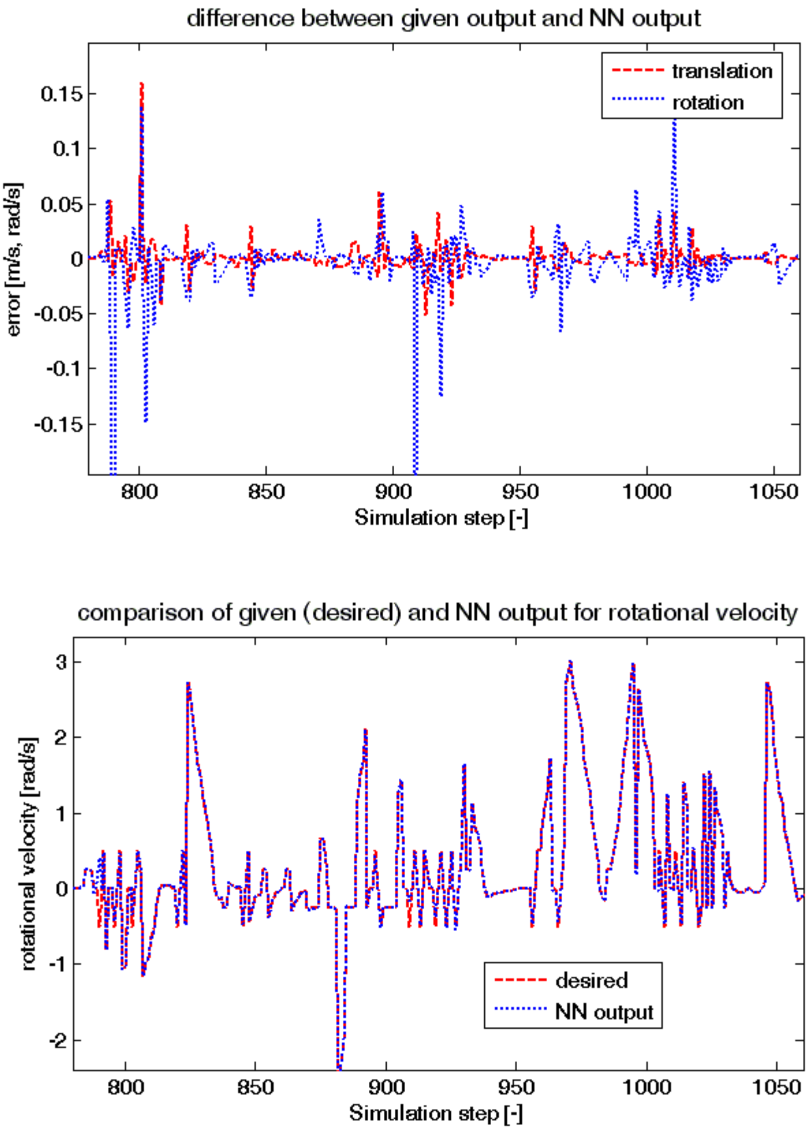

Fig. 3: Test run results.

Acknowledgement

Published results were acquired with the support of the Academy of Sciences of the Czech Republic under the research plan AV0Z20760514 and with the support of the Operational Programme Education for Competitiveness, project number CZ.1.07/2.3.00/09.0162.

References

Janglová, D. (2004) Neural Networks in Mobile Robot Motion, International Journal of Advanced Robotic Systems, vol. 1 (1), pp. 15-22.

Krejsa J., Věchet S. (2010) Odometry-free mobile robot localization using bearing only beacons, in Proceedigns of EPE-PEMC 2010 conference, Republic of Macedonia, pp. T5/40-T5/45.

Sirotenko M.Y. (2006) Applications of Convolutional Neural Networks in Mobile Robots Motion Trajectory Planning. Mobile Robots and Mechatronic Systems: Proceedings of Scientific Conference and Workshop, Moscow, pp. 174-181.

Yang S.X., Meng M.H. (2003) Real-time collision-free motion planning of a mobile robot using a Neural Dynamics-based approach, IEEE Transaction of neural networks, 2003; vol. 14 (6), pp 1541-1552.