Robotica (1998) volume 16, pp. 415–423. Printed in the United Kingdom

Rough level path planning method for a robot using SOFM neural network

Kimmo Pulakka* and Veli Kujanpää**

(Received in Final Form: October 10, 1997)

SUMMARY

In this paper a path planning method for off-line programming of a joint robot is described. The method can automatically choose the easiest and safest route for an industrial robot from one position to another. The method is based on the use of a Self Organised Feature Map (SOFM) neural network. By using the SOFM neural network the method can adapt to different working environments of the robot. According to test results one can conclude that the SOFM neural network is a useful tool for the path planning problem of a robot.

KEYWORDS: Path planning; SOFM neural network; Work environments; Robots

1. INTRODUCTION

Path planning is one of the main processes for an industrial robot. A path planning procedure should be easy to use and as fast as possible. The speed of procedure is the key issue for industrial productivity. If the path planning procedure is fast enough, the same robot can easily be used for many working tasks.

A traditional method for the path planning is based on the programming sequence. A robot expert sets the robot joint locations during the path from one working location to another. These sequences will take up a lot of time because working environments contain always many robot movements. On the other hand, the most simple part of path planning programming can be very boring work for programmers. These facts point to the need for an automatic path planning.

A configurational space method is one of the oldest automatic path planning methods. It gives the exact mathematical solution for the path planning problem. On the other hand, this method requires a lot of information for the solution. That's why the method needs a lot of memory and calculation time. Even if the computer technology is developing rapidly, these technical requirements are too difficult for up-to-date computers that can be bought at a realistic price. In today's world it is unrealistic to use this method for real planning problems.

The amount of the path planning data for computers must be decreased in order to make the calculation rapid enough. When the amount of the data is decreased, the needed information for a mathematical complete solution is also missed. That's why a new path planning procedure that can work with an incomplete system information must be found. Neural networks are used for applications where the application does not have all of it's working environmental information. In these cases the environment can be so complicated or difficult to measure that it is impossible to obtain all information. Neural networks are also used for path planning problems of different kinds of robot types.[1—4] There neural networks are used to imitate the characteristics of working environments or to build a function between different working situations and suitable behaviour of the robot. The main benefit of neural network based path planning methods is that they can adapt to the current working environment of the robot by the learning process of a neural network.

The basic problem in path planning is how we can move a joint robot from one working location to another. Usually, the working environment of the robot allows several different possibilities to perform a certain movement. The path planning system must know which movements are not suitable for a robot in the current working environment.

The aim of this work has been to develop a rough level path planning method for off-line programming of the robot that can automatically choose the easiest and safest route for an industrial joint robot moving from one position to another.

2. DEVELOPED PROCEDURE

Path planning can be separated into two different tasks: These tasks are rough level and fine level path planning. A path planning method must make a decision which is the suitable path for the kinematic behaviour of the robot. On the other hand, a path planning method must know how to move the joints of the robot during a certain path. In this research we have designed a rough level path planning method which can calculate ''goodness'' values for alternative paths. According to ''goodness'' values the path planning system can choose the right path to move the tool of the robot between two working locations.

2.1 Environment definition

During the environmental definition the robot expert defines all possible types of working environments of the robot. Environmental types are defined by sample environments. Each sample environment definition consists of obstacle definition, skeleton-surface definition, the stand point and two working locations of the robot.

* Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere (Finland). E-mail Kimmo.Pulakka@cs.tut.fi
** VTT Manufacturing Technology, P.O. Box 17021, FIN-53851 Lappeenranta (Finland). E-mail Veli.Kujanpaa@vtt.fi

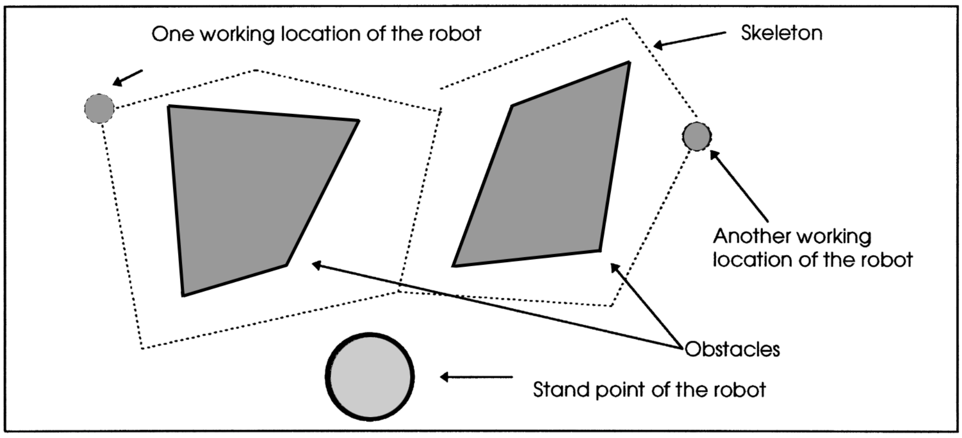

Fig. 1. Working environment elements of the robot (2D picture).

Obstacles are different objects in a working environment. A skeleton-surface describes the free space in the environment so that the surface is as far as possible from all obstacles. Two working locations of the robot show the start and end locations of the robot for the current moving task.

A skeleton-surface contains an infinite amount of points. Because of the limit of the computer memory the amount of skeleton-surface data is reduced. Therefore a skeleton-grid has been used to approximate the real skeleton-surface. A skeleton-grid contains grid points on a skeleton-surface and movements between grid points. The path planning method assumes that the tool of the robot moves across a skeleton-grid when the robot is moving between two working locations. It can be said that the path planning method can calculate "goodness" values for different skeleton-grid paths. Figure 1 shows all the elements needed to define the working environment of the robot.

The "goodness" of the sample environment is an important issue. The path planning system is based on learning by example. If the sample base for the environmental definition does not include examples needed for some types of environment, a path planning method cannot define "goodness" values for these missing types of environment routes. A good sample of environmental definition includes all types of working environments of the robot but it must not include all possible working environments.

2.2 Path description structure and neural network teaching

After the environmental definition, enough information is available to learn different kinds of paths in the working environment of the robot. Now the path planning method will produce a self-organised feature map (SOFM) neural network. After the teaching sequence, a SOFM neural network includes the description for different kinds of skeleton-grid paths. The path planning method describes paths using a certain path description structure.

The main reason for using the SOFM neural network for the path planning method is its capability to classify characteristics of the input space in a defined number of classes.[5] In the path planning method the input space consist of skeleton-grid paths described by the path description structure; SOFM classifies these paths into certain number of path classes. From the path planning point of view it is much easier to handle a certain number of path classes than the real space of skeleton-grid paths. The classification capability of a SOFM neural network is also used for the path planning method of a mobile robot.[1]

2.2.1. Path description structure. Path planning methods based on neural networks are totally depended on the empirical information obtained from the working environment of the robot. Input variables of the used neural network must be chosen so that they describe different working situations of the robot as unambiguously as possible. Usually, this requirement influences the large input variable space of the neural network system.[1,2,4] In this respect, our path planning method does not differ from the methods described in the bibliography.

The path description structure describes paths so that it is possible to separate different kinds of skeleton-grid paths. Path "goodness" is dependent on the kinematic behaviour of the actual robot. Structure variables are chosen so that they describe the current path from a robot kinematic point of view. The skeleton-grid path is divided to a defined number of information points. The characteristic of each point is described using seven variables. These variables are:

- The tool moving direction of the robot on the current skeleton-grid path
- Direction from the current robot TCP (tool centre point) to the robot standpoint location
- Distance from the current robot TCP to the robot standpoint location
- Direction from the current robot TCP to the nearest environment obstacle wall
- Distance from the current robot TCP to the nearest environment obstacle wall
- An average distance from the current robot TCP to the three nearest obstacle walls
- An average direction from the current robot TCP to the three nearest obstacle walls

A path description structure is a vector which includes $(n*7)$ embryos. $\langle n \rangle$ is the number of defined information points in a skeleton-grid path. Figure 2 shows the significance of the skeleton-grid path's information points.

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

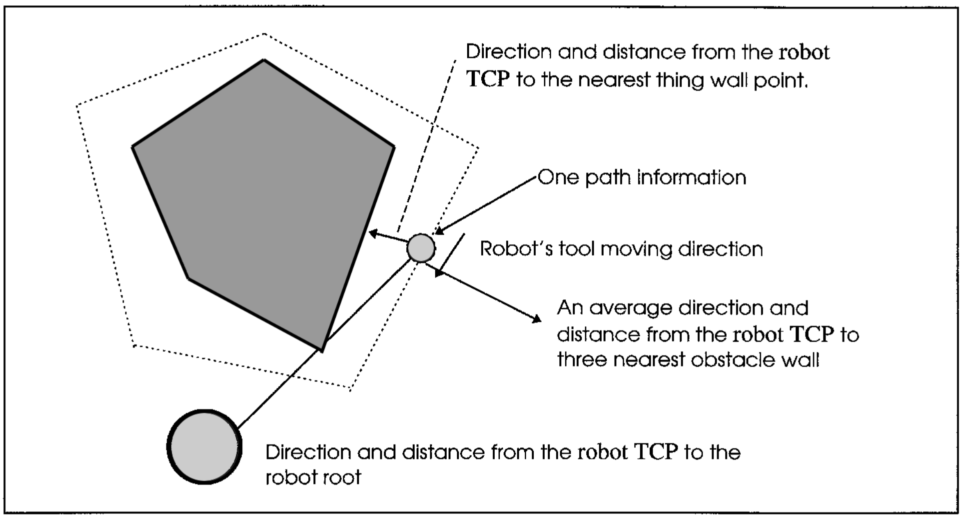

Fig. 2. Information points of the skeleton-grid path.

2.2.2. Neural network teaching. The SOFM neural network teaching process loads all obtained sample environments and defines all possible skeleton-grid paths in each sample environment from one working location to another working location. Figure 3 shows examples for different possible skeleton-grid paths between two working locations. We have used a standard SOFM learning algorithm for path teaching. This algorithm is described in reference 5. Path description structure vectors are input vectors for the SOFM neural network learning process.

After the teaching process the current SOFM neural network contains descriptions of different kinds of skeleton-grid paths. The network contains one kind of picture which describes sample paths characteristics. Each node of the SOFM neural network describes one type of skeleton path. According to the properties of the SOFM neural network the accuracy of the path is dependent on the SOFM neural network density. A dense network can describe sample paths more exactly than a sparse one.

2.3 Path goodness definition

When defining the path "goodness" the robot is driven along different skeleton-grid paths. The method defines the "goodness" of each SOFM neural network node to represent the current testing path. After that, it will increase or decrease the "goodness" value of the node according to the "goodness" of the testing path. If the testing path is satisfactory for a kinematic behaviour of the robot, the "goodness" value of the node will be increased; otherwise, it will be decreased.

After this operation, each of the classified routes described using a SOFM neural network has also a "goodness" value. The "goodness" value of the classified route is a probability value that shows how good this type of path is for the kinematic behaviour of the robot. The reliability of the "goodness" value depends on the number of sample environments that are used for the path "goodness" definition. It is important to use as large sample environment database as possible for path "goodness" definition.

2.4 Use of the path planning system

During the path planning task the system must choose the right path for the movement. The system calculates all possible skeleton-grid paths and checks their "goodness" values using the route classification information. The

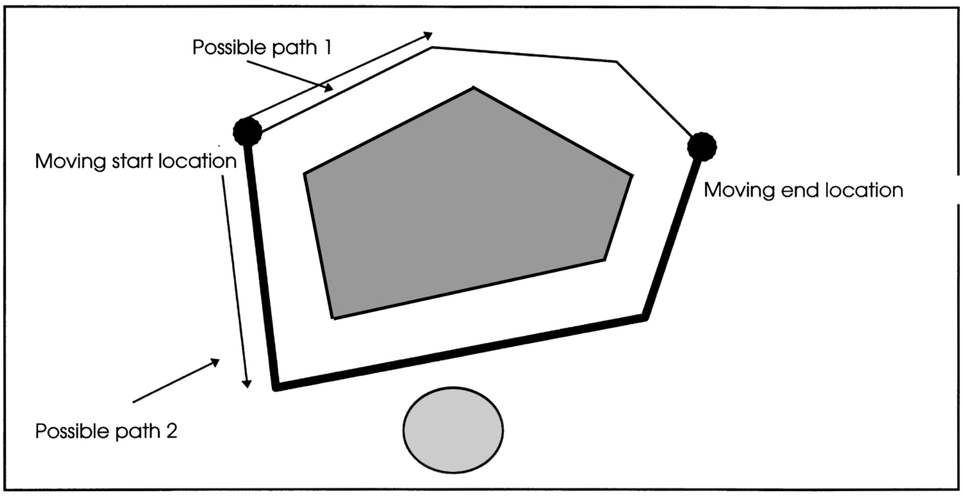

Fig. 3. An example of different skeleton-grid paths between two working locations.

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

skeleton-grid path which has the best "goodness" values is the winner. The system will use this path for the movement. Figure 4 describes the problem of choosing of the skeleton-path. In this example the skeleton path one is impossible for the robot. Therefore skeleton path two is the only possible path between these two working locations.

3. DESCRIPTION OF THE TESTING SEQUENCE AND TESTING RESULTS

The goal of the method testing was to test how useful the developed path planning method is for joint robots. On the other hand, it was important to find the main factor that affects the behaviour of the path planning system.

Tests were made in a two-dimensional space because the first tests for path planning method were easier to do and test results were also easier to analyse in a two-dimensional space than in a real three-dimensional space. Obviously, test results in the two-dimensional space must be good enough before it is reasonable to test the method in the three-dimensional space. Tests were made using testing programs that were made especially for this testing purpose. Testing programs can handle two-dimensional working environments of the robot. Obstacles in the working environment were two-dimensional obstacle blocks and the skeleton-grid surface was described as a two-dimensional line graph. The joint robot was described as a two-dimensional line-robot that has three joints. The test robot can move according to simple movement rules that are defined in the testing program. Figure 5 shows an example of a two dimensional testing environment.

The testing sequence consisted of the algorithm steps described above in chapter two. During the first algorithm step all kinds of necessary working environments of the robot were defined. The task of the second step was to teach the SOFM neural network. After this step the SOFM neural network included pictures of different kinds of path routes. The aim of the third step was to calculate "goodness" values for classified routes. After these three steps we define testing environments and calculated the performance of the path planning method.

There were 52 different working environments defined for the algorithm steps one, two and three, and 61 different environments to define ther performance of the path planning method. Each of these environments includes several skeleton path routes. The total number of paths used for the method initialisation process were 5016 and 5928 for the performance testing process. These environments can be divided into four different classes (Fig. 6). Environments are defined so that the environment database contains examples for each of the environment classes. Examples in a certain class are not exactly the same as those in some other classes, but they have some common characteristics.

In real path planning, the working environments of the robot cannot be exactly the same as used for the method initialisation during the algorithm steps one, two and three. That's why working environments for the method performance testing differ from the environments that are used for the method initiation process. The method must give right results for the "goodness" of skeleton path routes, although routes are not the same as those used during the method initiation process.

In the path planning method described in reference 4 a randomised definition of the working environment of the robot has been used to ensure a variation of the working environment for the neural network teaching process. In this research work environments used for the simulation are defined by a system operator. From the joint robot point of view a randomised environment definition process could generate too many environments which are totally impossible for the kinematics of a joint robot. The special definition of the working environment ensures that every environment contains at least some possible routes for the robot. On the other hand, this does not restrict the variation

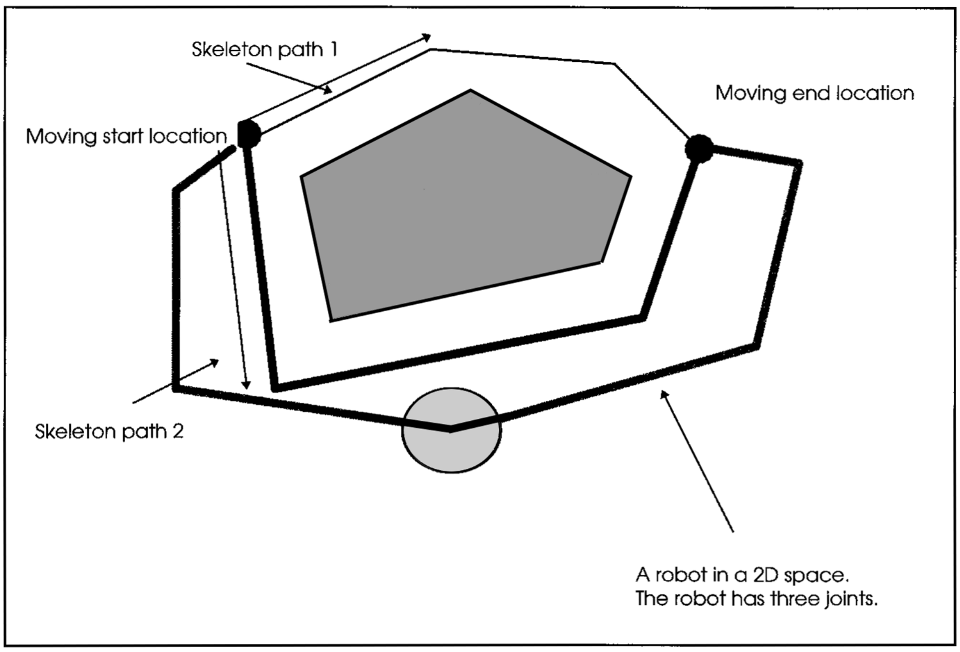

Fig. 4. Skeleton-path selection problem.

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

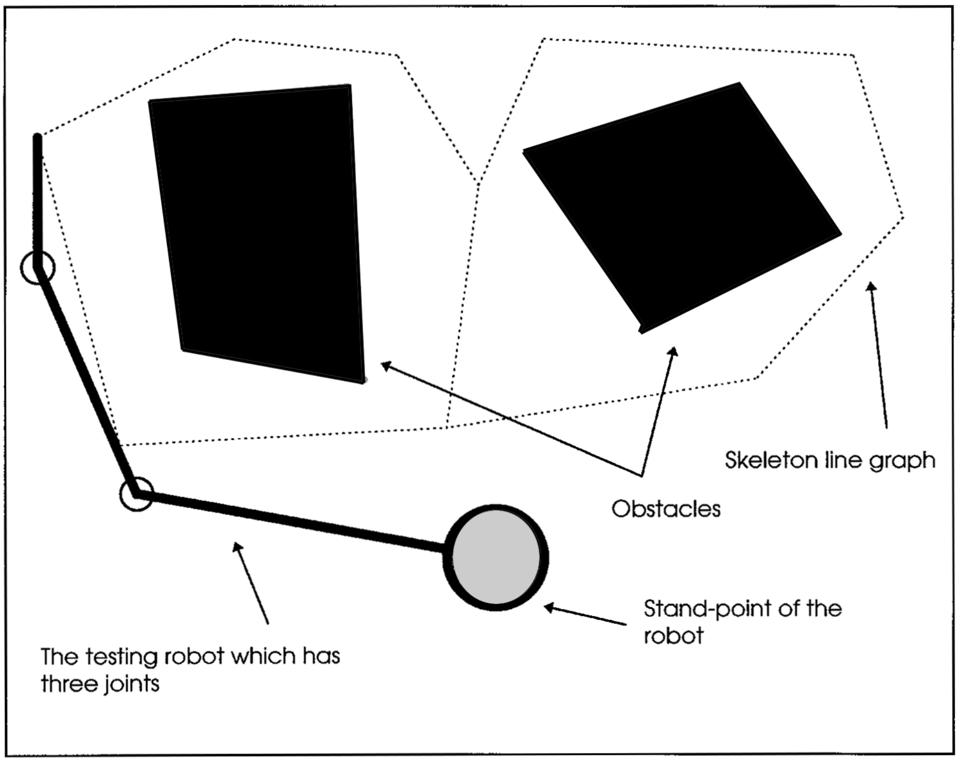

Fig. 5. Description of a two-dimensional testing environment.

of different skeleton-grid routes; working environments always differ somehow from each other.

The aim of testing the algorithms is to measure how good a predictor the path planning method is for these 61 testing environments. The path planning method gives a "goodness" probability value for each testing route. Skeleton routes are tried that are good enough for the robot kinematic behaviour. The testing program measures the real "goodness" probability for each classified skeleton route during the testing sequence. The path planning method works perfectly if "goodness" probability values are the same during the initiation and testing sequences. "Goodness" probability values are checked for 11 "goodness" probability classes. The same test is made using six different sizes of SOFM neural networks. The different size of SOFM neural networks gives different testing results. Results of the

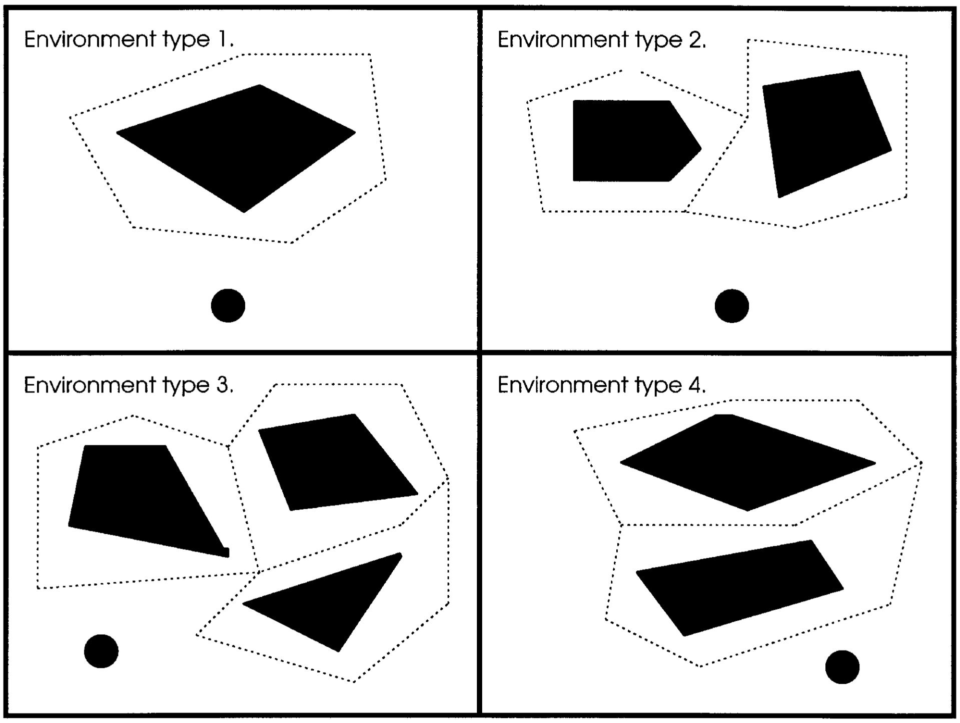

Fig. 6. Environment classes during the testing sequence.

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

testing sequence are printed in Table I; the meaning of measurement variables are described in Table II.

Blank cells in Table I mean that there are no testing routes for the current "goodness" probability class or the current path planning method that do not even contain the information for the "goodness" probability class. This means that the database used for the initialisation process of the path planning method does not include example skeleton

Table I. Test results.

Measures variables	SOFM-20	SOFM-100	SOFM-130	SOFM-170	SOFM-250	SOFM-300
L-vectors	5016	5016	5016	5016	5016	5016
SOFM size	20	100	130	170	250	300
Positive routes	3	9	11	19	17	20
Avg. I-counter value	250	50	38	29	20	16
Testing vectors	5928	5928	5928	5928	5928	5928
50–55%	0			0·3	0	
55–60		0		0	0·02	0·13
60–65%		0·05			0	0·09
65–70%		0·33	0·22		0·05	0
70–75%	0·09		0	0·01		
75–80%				0·19		0·58
80–85%			0·32			0·03
85–90%						
90–95%		0·02	0·01	0·04	0	
95–99%						
100%	0	0	0·08	0·14	0	0·13
Average error	0·03	0·08	0·13	0·11	0·09	0·16
Performance	2·91	8·28	9·57	16·91	15·47	16·8

Table II. Meanings of measurement variables

Measurement variable	Meaning of the measurement variable
L-vectors	Amount of skeleton-route vectors for the path planning initialisation process
SOFM size	The size of used SOFM neural network
Positive routes	Amount of the routes whose "goodness" is more than 50%
Avg. 1-counter value	Average value of example routes which are used for route "goodness" definition for each classified route
Testing vectors	Amount of used testing vectors
50–55%	Measured average error value for classified routes whose "goodness" probability value should be among 50–55%. *The value tells "goodness" probability value difference between the current probability class limits and measured "goodness" probability values for these routes whose probability values should be between the current probability class limits.*
55–60%	Measured average error value for classified routes whose "goodness" probability value should be between 55–60%
60–65%	Measured average error value for classified routes whose "goodness" probability value should be between 60–65%
65–70%	Measured average error value for classified routes whose "goodness" probability value should be betgween 65–70%
70–75%	Measured average error value for classified routes whose "goodness" probability value should be between 70–75%
75–80%	Measured average error value for classified routes whose "goodness" probability value should be between 75–80%
80–85%	Measured average error value for classified routes whose "goodness" probability value should be between 80–85%
85–90%	Measured average error value for classified routes whose "goodness" probability value should be between 85–90%
90–95%	Measured average error value for classified routes whose "goodness" probability value should be among 90–95%
95–99%	Measured average error value for classified routes whose "goodness" probability value should be between 95–99%
100%	Measured average error value for classified routes whose "goodness" probability value should be between 100%
Average error	Calculated average error across all found error
Performance	The calculated performance value of the path planning method. The performance value is calculated so that it also calculates the number of positive skeleton routes. Number of positive routes shows how many times it would be possible to calculate a wrong "goodness" value. **Performance = (1 − Average error) ∗ Positive routes**

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

routes for the current probability class. According to Table I it can be seen that the ‘‘goodness’’ of the performance increases when the performance value increases.

4. ANALYSIS OF THE RESULTS

According to the test results the average error made by the path planning method is 10%. Test results also show that a path planning method with a smaller SOFM neural network makes fewer errors than that with a larger SOFM neural network. On the other hand, the amount of the classified runable routes depends on the size of the used SOFM neural network. The path planning method with a large SOFM neural network can classify more suitable routes than that with a smaller SOFM neural network.

A large SOFM neural network consists of more nodes than a small one. That’s why it also describes routes learned during the initiation process more correctly than a small one. The amount of positive routes depends on the size of the SOFM neural network. The path planning method with a large SOFM neural network can find also more satisfactory routes than that with a smaller SOFM neural network. The difference between the suitabilities of route description, in different sizes of SOFM neural networks is described in Figure 7. It shows average Euclidean distances between best matching unit vectors of the SOFM neural networks and learning vectors. Learning vectors describe skeleton routes used during the initialisation process. Distances are measured after the initialisation process. The SOFM neural network with 20 nodes gives the largest average distance, and the neural network with 300 nodes gives the smallest average distance, viz. 20% less.

Test results also show that together with the increase of satisfactory classified routes the probability for big errors also increases when the size of the used SOFM neural network increases. The reason for this phenomenon is the amount of learning vectors used for the initialisation process. The same learning vector database is used for all tested path planning systems. This means that the average number of testing vectors used for each classified route decreases when the size of the used SOFM neural network increases. Hence the ‘‘goodness’’ of each classified route is defined more reliably in a small SOFM neural network than in a large one. Test results show that path planning methods with large SOFM neural networks produce big errors more frequently than those with a small SOFM.

It is also possible to observe the error made by the path planning method as a function of the amount of learning vectors used for the initialisation process. Figure 8 shows a result of this research. In Figure 8 the X-axis describes the number of example routes used for ‘‘goodness’’ calculations of classified routes, and the Y-axis describes the error between the classified ‘‘goodness’’ value and the measured one during the testing sequence. According to Figure 8, the variation and the average error decrease when the amount of learning vectors for classified routes increases.

On the other hand, there are still some errors although the number of learning samples per classified route is very high. Figure 7 shows that a planning method with a small SOFM neural network describes the working environment less carefully than that with a large SOFM neural network. It is possible that classified routes when using a small SOFM neural networks are described so imprecisely that the path planning method produces always some errors when the database used for the testing process is large and complex enough.

Although the method produces big errors in some test samples their number is smaller than that with a small error, as shown in Figure 9. These errors produced by the path planning method are divided into two error classes: There is one class for test samples when the error made by the method is equal or more than 10%, and one class for test samples when the error is less than 10%.

According to the analysis described above, it is important to find the right balance between the size of the used SOFM neural network and the number of used sample vectors for

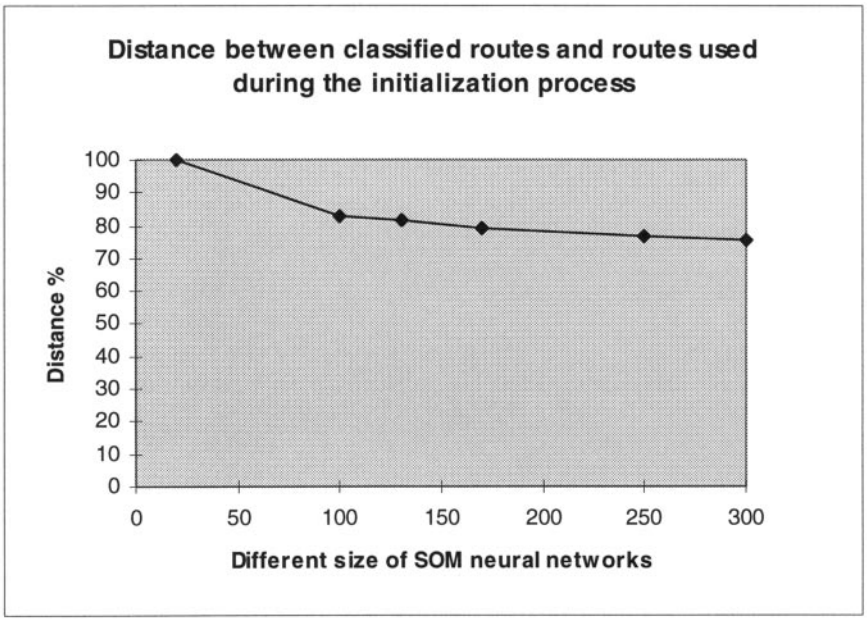

Fig. 7. Route description correctness in different sizes of SOFM neural networks.

https://doi.org/10.1017/S0263574798000447 Published online by Cambridge University Press

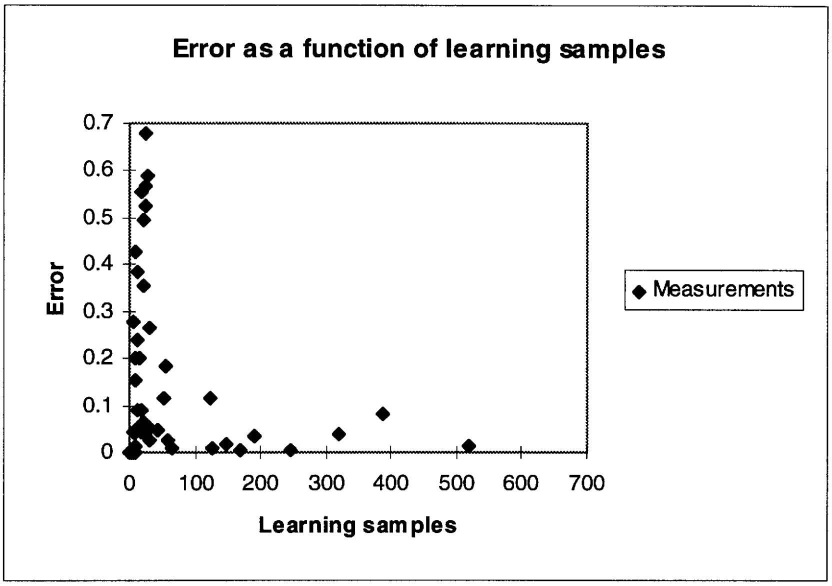

Fig. 8. Error as a function of learning vectors used in the initialisation process.

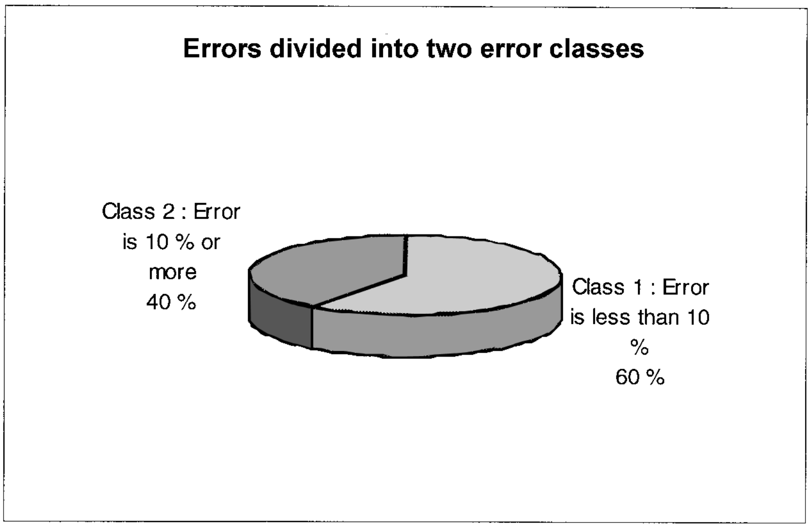

Fig. 9. Amount of test cases in two error classes.

the initialisation process. A large SOFM neural network can allocate more satisfactory routes than a small one, but a large SOFM neural network needs more sample vectors for the initialisation process for reliable results.

In the current test case at least 50 learning samples for each classified route are needed before the error made by the path planning method is stable. This means that the path planning method with a SOFM network size 100 would be the optimal choice if one of tested methods for the test case should be chosen. Methods which use larger SOFM neural networks need more sample data for reliability. The number of samples used for the method initialisation has also been mentioned in reference 4 as an important issue.

Although the method is tested in a two-dimensional space, it is at least in theory possible to use the method for real three-dimensional path planning problems. On the other hand, more research work is needed so that the method works more reliably even in a two-dimensional space. A three-dimensional working space contains much more data than a two-dimensional one, and it is much more difficult for the method. It is possible that the method needs some changes before it could be useful in a three-dimensional space. However, the path planning method described in this paper is a good basis for future development work for the path planning method based on the use of a SOFM neural network.

5. CONCLUSIONS

In this paper a rough level path planning method is described that can choose a suitable rough level path for a robot from one working point to another. The method is tested in a two-dimensional space. The average error made by the path planning method is 10%. This means that the difference between the path “goodness” prediction and its real “goodness” is 10% on the average. According to test results, the main issue for a reliable performance of the method is the number of learning vectors used for the initialisation process of the method. The size of the SOFM neural network should be chosen so that there are enough samples for the initialisation process of the path planning method. The method can work with a small sample space but the probability for big errors increases when the size of the sample space decreases. According to test results, one can say that a SOFM neural network is a useful tool for the path planning problem of robots.

References

1. A. Kurz, ‘‘Building maps for path-planning and navigation using learning classification of external sensor data’’ *Proc. of the 2nd Int. Conf. Artificial Neural Networks* (ICANN-92) (Eds., I. Alexander and J. Taylor) Sept. 4–7, Brighton, UK (1992) **Vol. 1**, pp. 587–590.
2. V. Santos, J.G.M. Goncalves and F. Vaz, ‘‘Perception maps for the local navigation of a mobile robot: a neural network approach’’ *Proc. of Int. Conf. 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, U.S.A. (1994) **Vol. 3**, pp. 2193–2198.
3. S. Faibish, G. Fruchtman and O. Dancygier, ‘‘Neural reflexive control of a mobile robot’’ *Proc. of Int. Conf. 1992 IEEE International Symposium on Intelligent Control*, Aug. 11–13, 1992, Glasgow, UK (1992) pp. 144–146.
4. C. Kozakiewicz and M. Ejiri, ‘‘Neural network approach to path planning for two dimensional robot motion’’ *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems ’91* (IROS ’91), Nov. 3–5, 1991, Osaka, Japan, 818–823 (1992) **Vol. 2**, pp. 818–823.
5. S. Haykin, *Neural Networks. A Comprehensive Foundation* (Macmillan College Publishing Company, Englewood Cliffs, NJ, U.S.A., 1994).