Computers and Electronics in Agriculture
18 (1997) 187–204

Computers
and electronics
in agriculture

Path planning of an agricultural mobile robot by neural network and genetic algorithm

Noboru Noguchi *, Hideo Terao

Department of Agricultural Engineering, Hokkaido University, Kita 9, Nishi 9, Kita-ku, Sapporo 060, Japan

Abstract

The purpose of this study is to develop a method that is able to create a suboptimal path of an agricultural mobile robot. This work is an attempt to apply a control technique combining a neural network (NN) and a genetic algorithm (GA). A NN is applied to describe the motion of the agricultural mobile robot as a nonlinear system because it is able to identify the dynamics of complex systems with its high learning ability. To create a path using a simulator described by the NN, a GA, which is inspired by biological evolution and uses a process of variation and selection to search a solution space, is utilized as an optimization method. Using this simulator and the GA, the time series of the steer angles, which were the control input, were optimized, and consequently an optimal work path of the mobile robot was created. The technique explored here should be applicable to a wide variety of nonlinear control problems in agriculture. © 1997 Elsevier Science B.V.

Keywords: Neural network; Genetic algorithm; Mobile robot; Path planning

1. Introduction

In order to utilize agriculture-related vehicles as robots, it is necessary to create the work path in advance. Because agricultural vehicles generally run repeatedly on predetermined paths, determination of the paths affects working efficiency and energy consumption. There have been many attempts to develop an agricultural

* Corresponding author. E-mail: noguchi@agm.hokudai.ac.jp

PII S0168-1699(97)00029-X

mobile robot as the 21st century approaches. Almost all previous studies of an agricultural mobile robot dealt with hardware techniques, such as spatial position-sensing systems and steering control systems, for following a predetermined path (Choi et al., 1990; Erbach et al., 1991; Smith et al., 1985, 1987; Tillett, 1991). Currently, Noguchi et al. (1997) have developed a mobile robot system, including a positioning system, using image sensors and a mobile robot which can be controlled by positions obtained from the image sensors and the heading angles from a geomagnetic direction sensor.

On the other hand, little work has been done towards the development of work schedules for agricultural mobile robots. However, to make agricultural vehicles autonomous, the creation of work paths and task planning must be developed. Noguchi and Terao (1995) dealt with task planning for hay transport on a slope using two transporting robots and a loading robot. They developed multi-layered genetic algorithms (LGA) to plan tasks for some of the robots. The proposed method could create a work schedule under not only a single objective function, such as work or work time, but also under a multi-objective function considering both of them. The effectiveness of this method was assured by comparing the results with the schedule made by a man with experience in hay transport.

The purpose of this study is to develop a method of creating the work path for an agricultural mobile robot. However, the precise model of the mobile robot and the technique which is able to optimize it are required to achieve this purpose. This work is an attempt to apply a control technique combining a neural network (NN) and a genetic algorithm (GA). In this study, a NN was applied to describe the motion of the mobile robot as a nonlinear system, because it was able to identify the dynamics of complex systems with its high learning ability. To create a path using the simulator by the NN automatically, the technique for optimization by a GA that can be applied to a wide range of optimization problems was then explored.

2. Related work

Motion planning of a mobile robot was brought into the domain of robotics. There is much basic research which considers the problem of motion planning for a car-like robot (Laumond et al., 1990; Laumond and Jacobs, 1994; Desaulniers and Soumis, 1995; Shan and Koren, 1995). Because a mobile robot has a nonholonomic constraint whose turning radius is lower-bounded, a significant amount of interest has been attracted during the last few years. Each navigational problem asks for a minimum-cost feasible path. The cost of a given solution may depend on many factors, including distance traveled, time, or energy expended.

The most straight forward methods for planning algorithms for a mobile robot are based on graph search approaches. In the simple paradigm, the path of the robot is defined by a sequence of translations and rotations of the rigid body and a description of the robot kinematics is not required. Motion planning then entails computing a path from a source to a destination that avoids collisions between the robot and the obstacles (Latombe, 1991). Hu et al. (1993) effectively solved this basic path planning problem using a network flow formulation. However, because the developed method has the time complexity of the network flow computation, they concluded that the hierarchical approach as a heuristic speedup should be adopted. A common heuristic for reducing search complexity is the hierarchical solution method based on 2^k-trees or dissection into triangular or rectanguloid cells. Unfortunately, the hierarchical solutions approach cannot guarantee optimal solutions (Zhu and Latombe, 1991). Alternatively, physical analog methods might be employed; these commonly involve a potential field that forces the robot to move toward goal configuration while avoiding contact with obstacles (Kohno and Thushima, 1994).

On the other hand, Li and Canny (1990) and Laumond and Jacobs (1994) applied differential geometric control theory to the motion planning of a car-like robot. Laumond and Jacobs (1994) used a simplified model of the movement of a real car and proposed the three-step algorithm to obtain an optimal feasible path under the objective function of the path length. It took into account both the nonholonomic constraint that specifies the tangent direction along any feasible path for the robot and a bound on the curvature of the path. Therefore, a dynamics characteristic of a steering system and lateral slip of the robot were not considered in the kinematic model. Using the same kinematic model of a mobile robot with that by Laumond and Jacobs (1994), Desaulniers and Soumis (1995) also presented a geometric approach for finding the shortest path for a car-like mobile robot with a minimal turning radius constraint. They presented an algorithm based on a new partition of the mobile robot configuration space and gave results showing that the algorithm was effective for rapidly finding an optimal path.

The fundamental problem with almost all methods is that the construction of the kinematic model is not suitable to apply to an industrial mobile robot, such as agricultural and construction robots. All kinematic models of car-like robots previously mentioned are constructed by ignoring the lateral slip phenomena of the robot. In particular, the motion of the agricultural mobile robot is affected by a lateral slip which occurs when using a larger steering angle. Noguchi et al. (1993) evaluated and discussed the accuracy between a general physics-based dynamic model neglecting a lateral slip and a NN-based dynamic model of an agricultural mobile robot. It was made clear that the output of the NN model was more correct than that of the physics-based model. The NN model implicitly includes the nonlinear dynamic characteristics, such as longitudinal and lateral slips, due to the acquisition of the training data of the NN through the actual travels of the robot. In fact, Nguyen and Widrow (1990) addressed the problem of controlling severely nonlinear systems from the stand point of utilizing a NN to achieve a nonlinear controller. A NN controller steering a trailer truck while backing up to a loading dock was demonstrated by simulation. A simulator of the trailer truck was also constructed as a nonlinear system by a NN. The simulator built by the NN was able to identify the system's dynamic characteristics and represent nonlinearity in the emulator, such as the jackknifed phenomena. Therefore, in order to solve the path planning problem for an agricultural mobile robot, nonlinearity of the simulator is essential for accurate modeling of the kinematics. This study presents a technique which can identify the dynamic characteristics of the robot and then solve the path planning problem using the nonlinear model.

3. Simulator of the mobile robot by the neural network

3.1. Tested mobile robot

Fig. 1 shows a fabricated agricultural mobile robot. The specifications of the robot are shown in Table 1. The robot was designed so that the specifications corresponded to those of a conventional small-sized tractor. This mobile robot has a rear-wheel drive and is driven by a gasoline engine. The robot can measure the steer angle using a potentiometer and the rotation of the rear wheels using rotary encoders. It can also control by electrical means and using a computer, the steer angle, the clutch and brake systems. The robot has a maximum steer angle of about 40° and a maximum steering rate of 7°/s. Travel velocity is in the range of 0.4–1.2 m/s.

3.2. Construction of the robot simulator

Since the motion of agricultural vehicles that use larger steer angles compared with automobiles shows high nonlinearity, it cannot be modeled with high accuracy with conventional techniques, such as a linear differential equation or a simple kinematics equation (Noguchi and Terao, 1994). In this study, the NN illustrated in Fig. 2(a) was utilized in the dynamic model of the mobile robot. The network is layered, consisting of one input, one output and two hidden layers. The input and hidden layers have five units each, while the output layer has three units. The input vector z is composed of the forward speed v_x, the lateral velocity v_y and the yaw angular velocity ω. The control vector u consists of the steer angle ψ and the change in the steer angle $\Delta\psi$. The output vector represents the state of the robot one second after the present state. The forward velocity v_x and the lateral velocity v_y are expressed in terms of the robot coordinate system. Using this simulator, the velocities v_x, v_y and ω in the space fixed system and the locations by integration of these velocities could be obtained. These velocities result from a given time history of the steer angle ψ. The change in the steer angle $\Delta\psi$, which was an element of the control vector, became a constraint when controlling the robot and was the control input calculated by the GA.

The units in each input layer store the input values. The hidden layer and output layer units each carry out two calculations (Fig. 2(b)). The first calculation involves multiplying all inputs by a weight w_{ij} and then summing the result as:

$$s_j = \sum_{i=1}^{N} w_{ij}\alpha_{ij} \tag{1}$$

In the second, the output of the units (β_j) is calculated as the sigma function of s_j using:

$$\beta_j = \sigma(s_j) \tag{2}$$

where

$$\sigma(s) = [1 + \exp(-s)]^{-1} \tag{3}$$

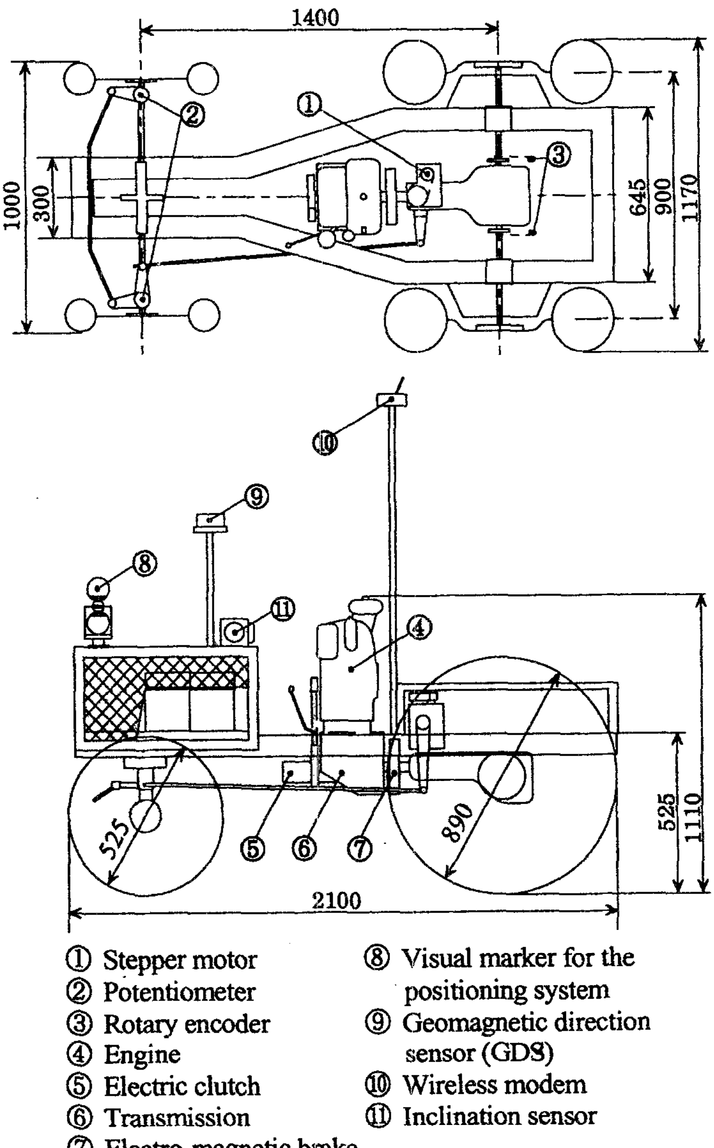

Fig. 1. Schematic diagram of the tested mobile robot.

Table 1
Specification of the tested mobile robot

Speed (m/s)	
Forward	0.40 ~ 1.21
Reverse	0.43 ~ 0.92
Mass on wheels (kg)	392
Front	130
Rear	262
Distance from centre of gravity to rear axle (mm)	466
Centre of gravity height (mm)	512
Tread (mm)	900
Wheel base (mm)	1400
Min turning radius (m)	2
Max steer angle	40°
Max steer rate	7°/s

3.3. Learning of the NN

The coordinate system defined in this paper is shown in Fig. 3. The data sets for teaching the NN were obtained by running the test robot on an asphalt surface. To measure the locations and directions of the test robot during travel at an interval of 1 s, electric solenoid valves were mounted to the head and the rear of the robot center. These valves sprayed pressurized water on the ground and the time of movement of the robot was measured with a tape. The three components of velocity which were transformed from the locations of the robot were adopted as the learning data for the NN. To measure the location of the robot, the steer angle measurements were synchronized with the timing of water spraying. The training data for the NN were obtained using a forward velocity of 0.5 m/s in various travel patterns including steady-state circle turns and some sinusoidal paths.

A back propagation algorithm (BP) was adopted as the learning method of the NN. The aim of the BP as derived by Rumelhart et al. (1986) is to find a set of weights w_{ij} such that for each input vector the output vector computed by a NN is the same as the desired output vector. Using this algorithm, the NN for the robot simulator learned by adjusting its weights so as to minimize the sum of squared errors between outputs and learning data set. Steepest descent algorithm was utilized in minimizing the error. The modification rates of weights and biases were scheduled so that they gradually decreased with respect to the degree of error convergence. The learning required 900 sweeps through a set of 180 input vectors

3.4. Performance of the robot simulator

The performance of the robot simulator is shown in Fig. 4. The figure compares three velocity components between the outputs of the robot simulator and the experimental results. A fairly low value of root mean square error was obtained when the simulator and actual robot velocities (v_x) were compared (Fig. 4(a)). As a result, the simulator and actual robot velocity (v_x) were equal. High correlation coefficients existed between the lateral velocity v_y and the actual robot velocity, and the yaw angular velocity and the actual robot velocity ω (Fig. 4(b), (c)). The r.m.s. errors of the lateral velocity, v_y and the yaw angular velocity, ω were 0.0155 m/s and 0.0156 rad/s, respectively.

The trajectories of the actual robot and the simulator are shown in Fig. 5. The sinusoidal input of steer angles, which has a period of 30 s and an amplitude of 15°, was applied to both the actual robot and the simulator. The trajectory obtained by the simulator agreed well with experimental results.

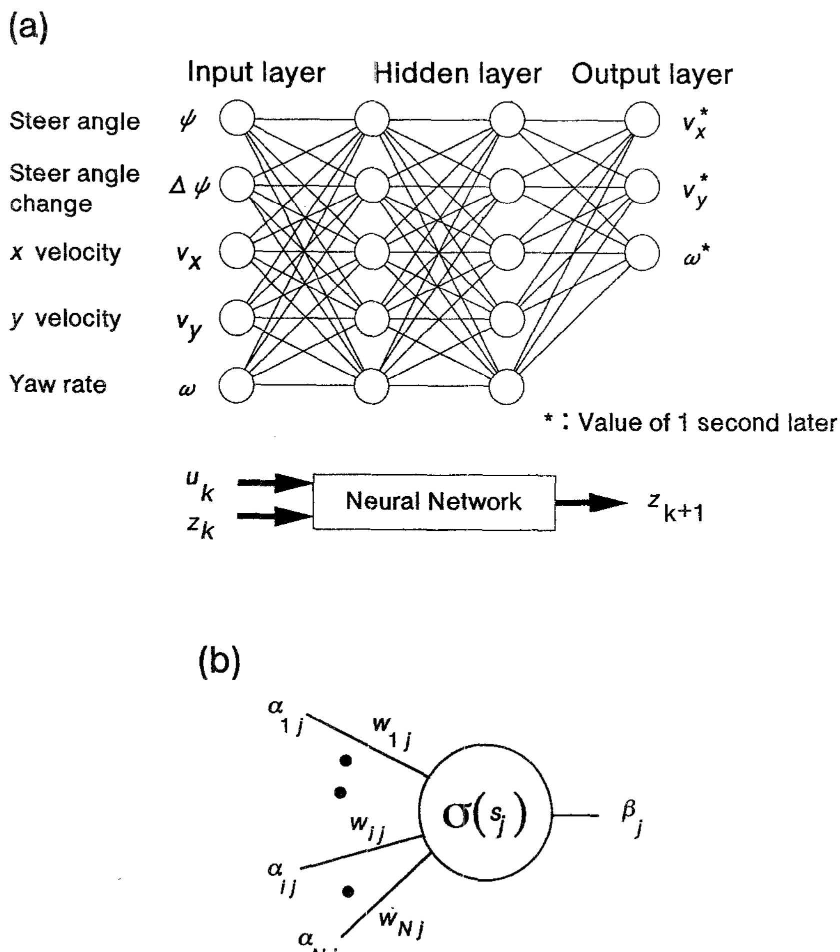

Fig. 2. Neural network structure for the motion of the mobile robot. (a) Calculation method of each unit; (b) construction of neural network.

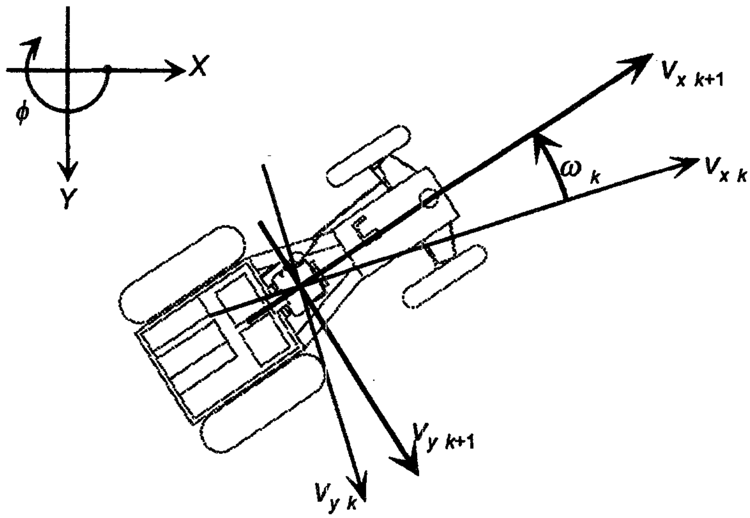

Fig. 3. x-y coordinate system for describing the robot motion.

4. Optimization of the work path by the genetic algorithm

4.1. Overview of GAs

Genetic algorithms proposed by Holland (1975) are general purpose stochastic optimization methods for search problems. Genetic algorithms are interesting because they are inspired by biological evolution and they seem applicable to a wide range of optimization problems. The data processed by the algorithm consists of a set (population) of strings which represent multiple points in a search space. A string with a finite length in which each bit is called an allele, is defined as a solution (individual) having the objective function value of a point in the search space. The function to be minimized by the algorithm is converted to a fitness value that determines the probability of the individual undergoing transitional operators. The operators are analogous to the biological terms of crossover, mutation and selection.

4.2. Optimization problem

A GA was used to create a work path satisfying the given initial and final conditions of the robot. When the control vector was defined by $u = (\Delta\psi, \psi)^{\mathrm{T}}$ and the location and yaw angle of the robot by $\chi = (x, y, \phi)^{\mathrm{T}}$, the problem of optimal control could be described by Eq. (4) to (7).

$$\text{min. } V(u) \tag{4}$$

$$\text{subject to } \Delta\psi \in \{-7, -6, -5, \cdots, 6, 7\} \tag{5}$$

$$\psi \in \{-40, -39, -38, \cdots, 40\} \tag{6}$$

$$[\chi - \chi_{\mathrm{d}}]_{t=t_{\mathrm{f}}} = 0 \tag{7}$$

where t_{f} is the final time and χ_{d} is the desired state. In this paper, the sum of the steer angle changes of Eq. (8) and the travel time of Eq. (9) were applied to the objective function.

$$V(u) = \sum_{k=0}^{t_{\mathrm{f}}} (\Delta\varphi_k)^2 \tag{8}$$

$$V(u) = t_{\mathrm{f}} \tag{9}$$

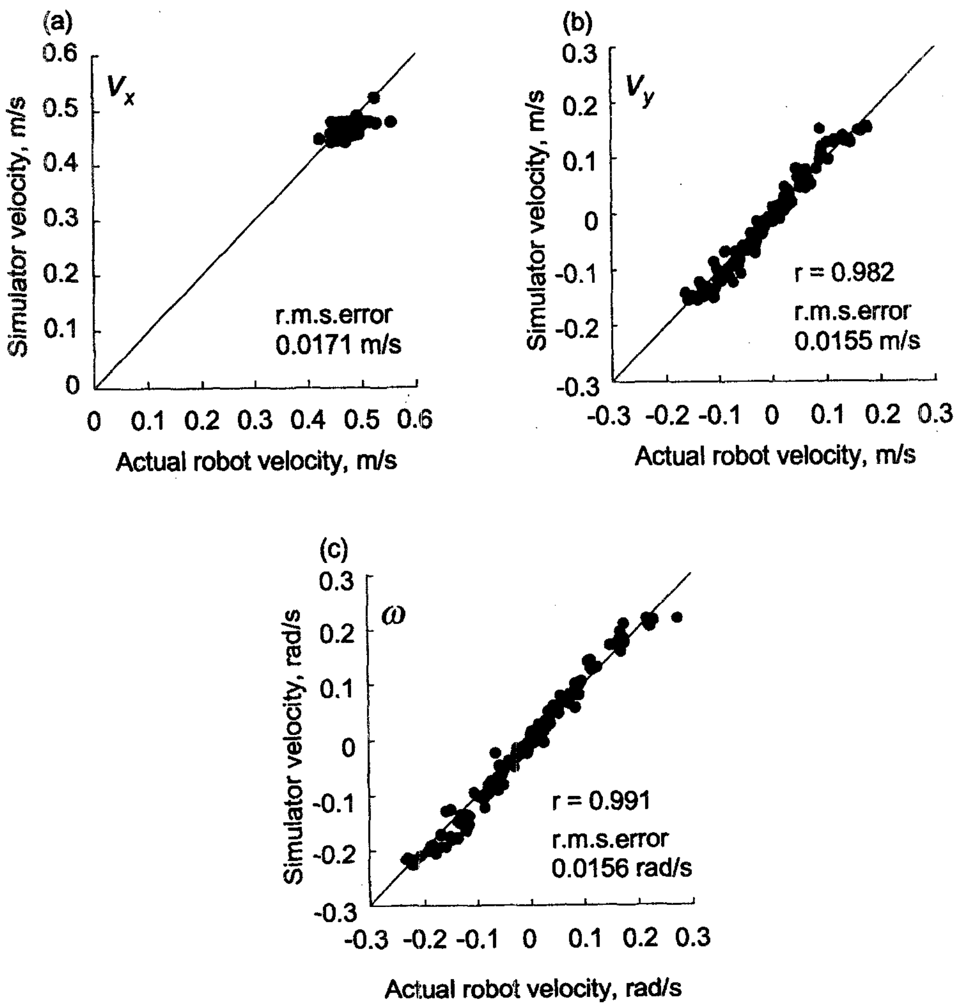

Fig. 4. Performance of robot simulator for three velocity components (v_{x}, v_{y}, ω). (a) Longitudinal velocity, v_{x}; (b) Lateral velocity, v_{y}; (c) yaw velocity, ω.

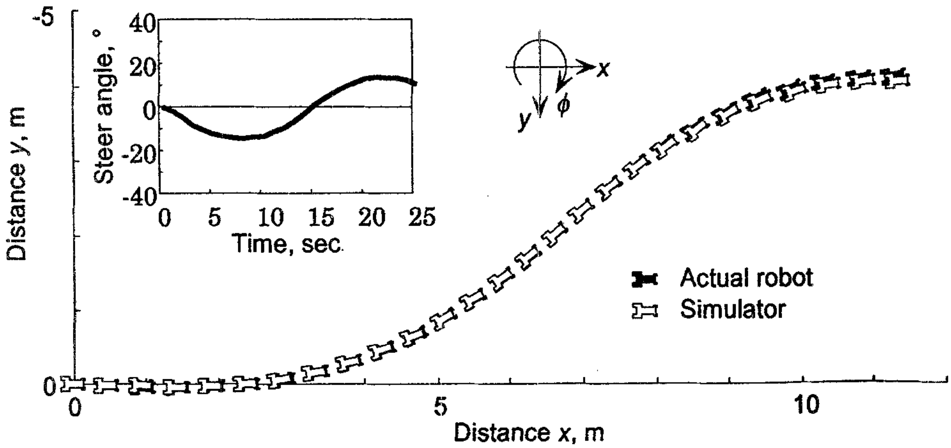

Fig. 5. Comparison between the trajectory of the robot simulator and that of the actual robot.

Constraints were represented by Eq. (5) to 7. The range from -7 to $+7°$ of $\Delta\psi$ in Eq. (5) and the range of ψ from -40 to $+40°$ in Eq. (6) were determined from the structural restrictions of the tested robot. Eq. (7) represents the equality constraints of the final positions of the robot. The state vector χ, which was composed of the location and yaw angle of the robot, was calculated from the output z of the simulator using Eq. (10):

$$\chi_k = \sum_{i=0}^{k-1} \begin{pmatrix} \cos\phi_i & -\sin\phi_i & 0 \\ \sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} z_i \tag{10}$$

A penalty method was adapted in the transformation from the constrained optimization problem to the unconstrained one, and the transformed objective function $V'(u)$ was described by:

$$V'(u) = V(u) + \lambda[\chi - \chi_\mathrm{d}]^T[\chi - \chi_\mathrm{d}]_{t=t_\mathrm{f}} \tag{11}$$

where λ is the weighting factor of the penalty method.

There are some obstacles, such as stumps and rocks in the work space of an agricultural vehicle. Investigation showed that it was also possible to determine an obstacle-avoiding path by expanding Eq. (11). Eq. (12) which penalized individuals that run over an obstacle was adopted as the objective function.

$$V'(u) = V(u) + \lambda[\chi - \chi_\mathrm{d}]^T[\chi - \chi_\mathrm{d}]_t = {}_{t_\mathrm{f}} + p \tag{12}$$

When expressing a rectangular obstacle using the coordinate of the left top corner (x_1, y_1) and that of the right bottom corner (x_2, y_2), the region of the obstacle was described as a partial set O of the vehicle location as shown in Eq. (13). The penalty p in Eq. (12) was calculated by Eq. (14).

$$O = \{\chi / x < x < x_2, \quad y_1 < y < y_2\} \tag{13}$$

$$p = \begin{cases} V_{\max} & (\chi \in O) \\ 0 & (\chi \notin O) \end{cases} \tag{14}$$

4.3. Optimization by the GA

A flowchart of path creation by a GA is shown in Fig. 6. Crossover, mutation and selection were used as transitional operators of the GA. The string coded the time series of the change in the steer angle $\Delta\psi$, shown in Fig. 7(a). The value of each allele was an integer in the range of Eq. (5). In order to create the working path during a maximum of 100 s, the string of the change in steer angle was composed of 100 alleles. When the robot reached the desired location within the maximum time, it adjusted the travel time by changing the effective length of the string. The algorithms of crossover and mutation are shown in Fig. 7(b) and (c). Crossover was a random exchange of multiple alleles between the selected mating pairs G_{p0} and G_{q0}, which created two offspring G_{p1} and G_{q1}. The number of alleles to exchange was determined as half the total number of alleles possessed by an individual. Mutation happened with a probability of 1.0 on selected individuals that

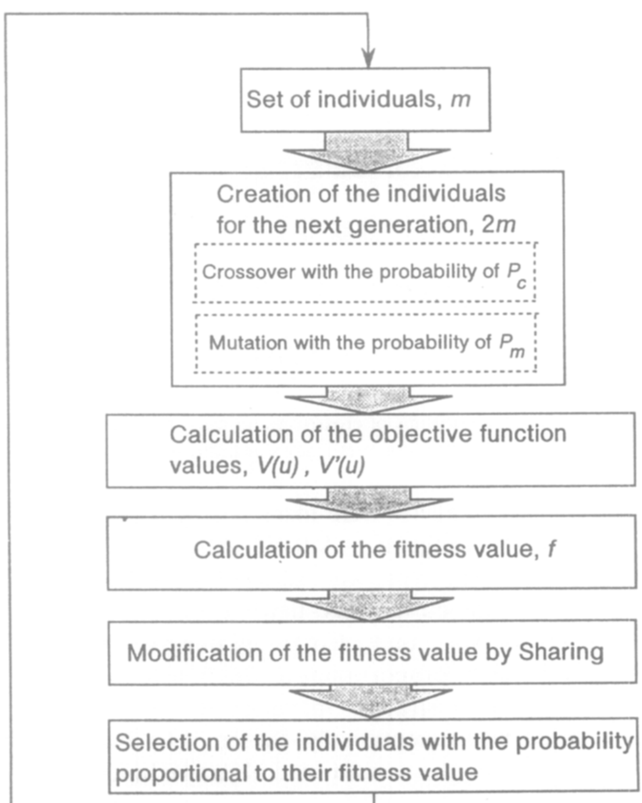

Fig. 6. Flowchart of a path creation algorithm using the GA.

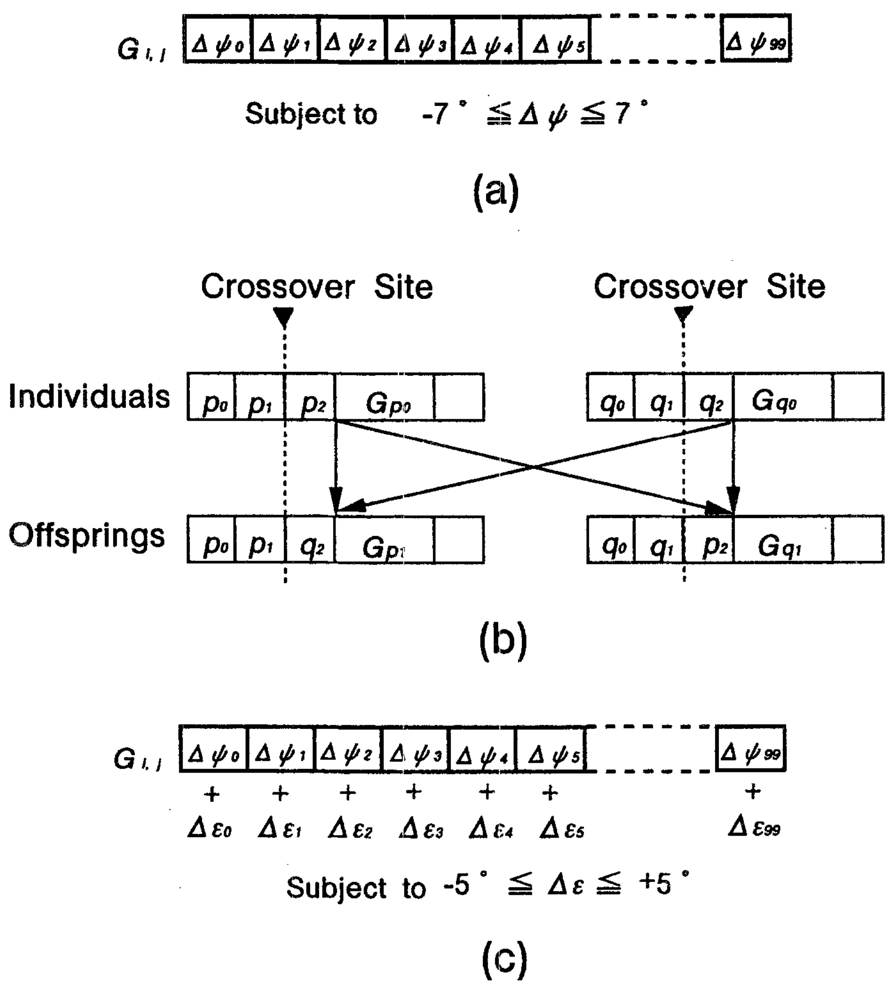

Fig. 7. Coding of individuals and algorithms of crossover and mutation on the GA. (a) Coding method of individuals; (b) algorithm of crossover operation; (c) algorithm of mutation operation.

had not crossed over. All the alleles were increased by a mutation width $\Delta\varepsilon$ which was randomly chosen in the range of -5 to 5°. After these transitional operators created new individuals, the survivors were selected from the double-sized tentative population which included the current and newly created individuals. Selection, an analog to natural selection, was conducted by spinning a simulated roulette wheel whose slots had different sizes proportional to the fitness values of the individuals and by an elitist preservation strategy in which the individual with the lowest objective function value was exceptionally chosen. The fitness value f for each individual was converted from the transformed objective function $V'(u)$ by Eq. (15):

$$f = \frac{1}{V_{\min} - V_{\mathrm{mean}}} V'(u) + \frac{V_{\min} - 2V_{\mathrm{mean}}}{V_{\min} - V_{\mathrm{mean}}} \tag{15}$$

where V_{min} is the minimum objective function value and V_{mean} is the mean of the objective function value for all individuals. In addition, to maintain the diversity of individuals in the evolution process, the fitness value was corrected using a sharing operation which was proposed by Ichikawa and Sano (1992). The size of the population for the next generation was also kept constant by choosing half of the tentative population. The paths of Fig. 9 to Fig. 11 were obtained using the GA parameters of population size $m = 100$, the number of generations $N_g = 600$, the probability of crossover $P_c = 0.7$ and the probability of mutation $P_m = 0.3$.

5. Results and discussion

Fig. 8 shows a block diagram of the work path creation method combining the NN and the GA. The time series of the steer angles ψ and the changes in the steer angles $\Delta\psi$ created by the GA were introduced to the NN and then the paths were created. The GA searched the optimal path through evolutional computation using the objective function of the created paths. Fig. 9 shows the creation process of the path by the GA. The initial state $\chi_i = (0\text{ m}, 0\text{ m}, 0°)^T$ and the desired final state $\chi_f = (20\text{ m}, 10\text{ m}, 0°)^T$ were set as the travel conditions. The paths in Fig. 9 are the individuals having the highest fitness value in the 100th, 500th, 700th and 1000th generations. The final state errors and the objective function of each created path are shown in Table 2. The sum of the steer angle changes was applied to the objective function. It was found that the quality of the path improved the generation of the GA. It was also found that the errors in the final state of the robot decreased, and as the GA generation progressed, the robot precisely approached the target state. In addition, the objective function value, which is the r.m.s. of the steer angle changes, decreased by 25% between the 100th and 1000th generations. It was clear that the developed method using NN and GA was effective for creating a path for the robot.

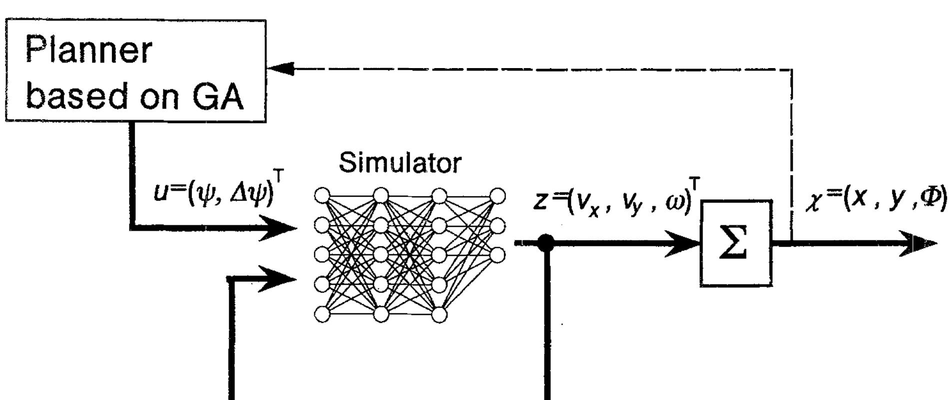

Fig. 8. Block diagram for creation of an optimal path using the NN and the planner of steer angles based on the GA.

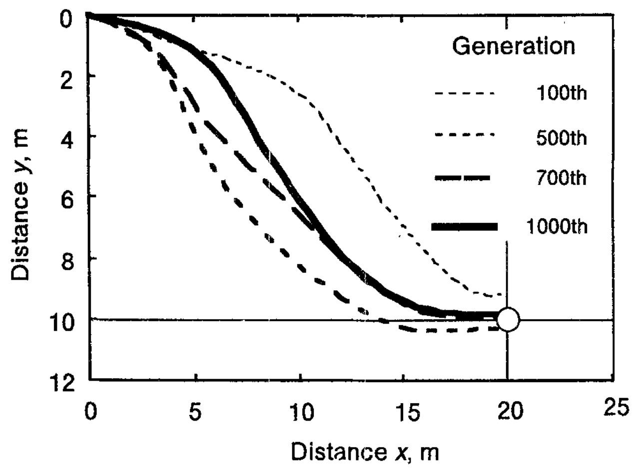

Fig. 9. Change in the created paths with progress in the generation of the GA.

Fig. 10 compares paths created under the different objective functions of Eq. (8) and Eq. (9). The sum of the steer angle changes and the travel time are adopted as the objective function. The figure also shows the total travel time and the steer angles during travel. The initial state of the robot was $\chi_i = (0\text{ m}, 0\text{ m}, 0°)^T$ and the desired final state was $\chi_f = (20\text{ m}, 20\text{ m}, 90°)^T$. The two paths almost satisfied the constraint of the final state. However, the different paths were obtained when the objective function was changed for optimization. The path which minimized the sum of the steer angle change led to the target position by a steer handling of 28% compared with that which minimized the travel time. Similarly, by minimizing the sum of the steer angle change, the travel time increased by 5 s compared to when minimizing the travel time. It was also found that the path that used steer handling resembled a circle turn, and the path created by the objective function of the travel time was similar to a straight line.

Next, to confirm the creation of an appropriate path that could avoid obstacles, the same initial and final states as those of Fig. 10 were established. The obstacle was laid in the region in which the previous two paths passed. The parameters of

Table 2
Change in the errors of the final state and the objective function value with progress of generation

		Generations			
		100th	500th	700th	1000th
Error of final state $[\chi-\chi_d]_{t=t_f}$	x (m)	−0.37	0.23	−0.15	−0.11
	y (m)	−0.79	0.32	−0.05	−0.15
	Φ (°)	1.01	−0.43	−1.26	−0.92
Objective function	r.m.s. of $\Delta\psi$ (°)	2.61	2.50	2.30	1.97

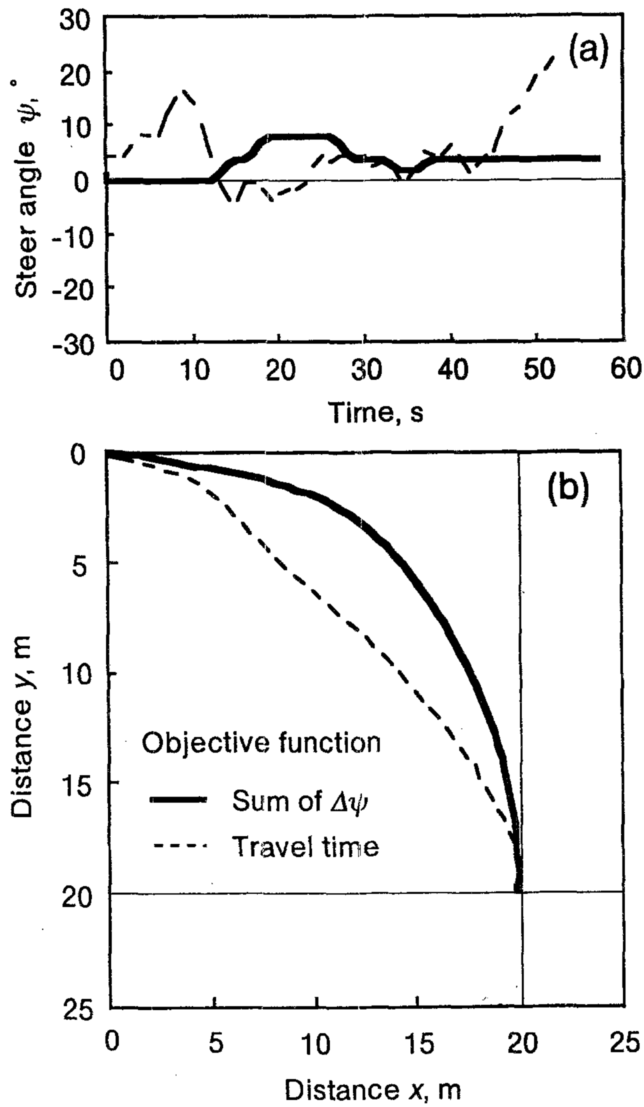

Fig. 10. Comparison of paths which were created under the different objective functions: The sum of the steer angle changes and the travel time. (a) Time series of the steer angles; (b) created paths.

the obstacle defined in Eq. (13) were $(x_1, y_1) = (14$ m, 3 m) and $(x_2, y_2) = (16$ m, 13 m). Fig. 11 shows the path and the steer angles of the individual having a minimum objective function in the 1000th generation. The appropriate path avoiding the obstacle was obtained using Eq. (12) to (14). Rapid steer angle changes were required to avoid the obstacle and achieve the desired final state.

6. Conclusions

(1) A NN was applied to a dynamic model of an agricultural mobile robot. A BP algorithm was utilized in training the NN and was accurate enough in simulating the robot path. The r.m.s. errors of the forward velocity v_x, the lateral velocity v_y and the yaw angular velocity ω were 0.0171 m/s , 0.0155 m/s and 0.0156 rad/s, respectively. In addition, the trajectory obtained by the simulator agreed well with that of the actual robot.

(2) A GA was applied as the technique for creating an optimal path. The string coded the time series of the steer angle changes of the mobile robot. In order to search the optimal path effectively, crossover, mutation and selection were used as the transitional operators of the GA.

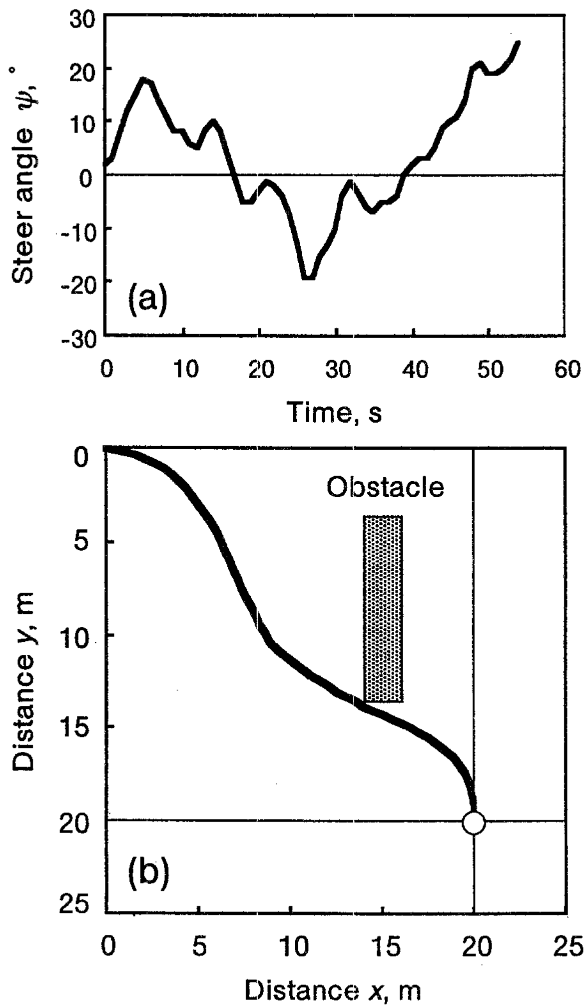

Fig. 11. Path creation that could avoid an obstacle. (a) Time series of the steer angles; (b) created path.

(3) The quality of the created path improved as the generations progressed. Both the error in the final state and the objective function value decreased as the generation proceeded. It was clear that the developed method using the NN and the GA was effective in creating a path for the agricultural mobile robot.

(4) The paths were created under different objective functions (the sum of the steer angle change and the travel time). It was found that an appropriate path considering each objective function can be created by combining the NN and the GA. In addition, it was also possible to search an obstacle-avoiding path by adding the penalty method to the combination of the NN and the GA.

References

Choi, C.H., Erbach, D.C., Smith, R.J., 1990. Navigational tractor guidance system. Trans. ASAE 33 (3), 699–706.

Desaulniers, G., Soumis, F., 1995. An efficient algorithm to find a shortest path for a car-like robot. IEEE Trans. Robotics Automat. 11, 819–829.

Erbach, D.C., Choi, C.H., Noh, K., 1991. Automated guidance for agricultural tractors. In: Proceedings 1991 ASAE Symposium on Automated Agriculture for the 21st Century, 16-17 December 1991, Chicago, IL. ASAE, pp. 182–191.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, MI.

Hu, T.C., Kahng, A.B., Robins, G., 1993. Optimal robust path planning in general environments. IEEE Trans. Robotics Automat. 9, 775–784.

Ichikawa, Y., Sano, T., 1992. Neural network application for direct feedback controller. IEEE Trans. Neural Networks 3 (2), 224–231.

Kohno, Y., Thushima, K., 1994. A Path-Planning Algorithm of Mobile Robot using Potential Locally. Proc. ROBOMEC 1994, 30 June–1 July 1994, Japan, JSAM, pp. 127–128. (in Japanese with English abstract).

Latombe, J.C., 1991. Robot Motion Planning. Kluwer Academic, Boston, MA, 651 pp.

Laumond, J.P., Taix, M., Jacobs, P., 1990. A motion planner for car-like robots based on a mixed global/local approach. Proc. IEEE Int. Workshop Intell. Robots Syst., Japan, pp. 765–773.

Laumond, J.P., Jacobs, L.P., 1994. A motion planner for nonholomic mobile robots. IEEE Trans. Robotics Automat. 10, 577–593.

Li, Z., Canny, J., 1990. Motion of two rigid bodies with rolling constraint. IEEE Trans. Robotics Automat. 6, 62–70.

Nguyen, D.H., Widrow, B., 1990. Neural networks for self-learning control systems. IEEE Contr. Syst. Mag., 18–23.

Noguchi, N., Ishii, K., Terao, H., 1993. Optimal control of agricultural vehicles by neural networks (Part 1). J. JSAM 55 (5), 83–92. (in Japanese with English abstract)

Noguchi, N., Terao, H., 1994. Creation of optimal route for agricultural vehicle and construction machinery by using genetic algorithm. Trans. SICE 30 (1), 64–71. (in Japanese with English abstract)

Noguchi, N., Terao, H., 1995. Work schedule of industrial robotics by using multi-layered genetic algorithms. Trans. SICE 31 (10), 1762–1768. (in Japanese with English abstract)

Noguchi, N., Ishii, K., Terao, H., 1997. Development of an agricultural mobile robot using a geomagnetic direction sensor and image sensors. J. Agri. Eng. Res. 67(1), in press.

Rumelhart, D., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. Parallel Distributed Processing, Vol. 1. MIT Press, Cambridge, MA.

Shan, Y., Koren, Y., 1995. Obstacle accommodation motion planning. IEEE Trans. Robotics Automat. 11, 36–49.

Smith, L.A., Schafer, R.L., Young, R.E., 1985. Control algorithms for tractor-implement guidance. Trans. ASAE 28 (2), 415–419.

Smith, L.A., Schafer, R.L., Bailey, A.C., 1987. Verification of tractor guidance algorithms. Trans. ASAE 30 (2), 305–310.

Tillett, N.D., 1991. Automation guidance sensors for agricultural field machines: A Review. J. Agri. Eng. Res. 50, 167–187.

Zhu, D., Latombe, J.C., 1991. New heuristic algorithms for efficient hierarchical path planning. IEEE Trans. Robotics Automat. 7, 9–20.