Department of
AUTOMATIC
CONTROL
and SYSTEMS
ENGINEERING
UNIVERSITY of SHEFFIELD

Hybridisation of Neural Networks and Genetic Algorithms in an Application of Time-Optimal Control

N. Chaiyaratana and A.M.S. Zalzala

Robotics Research Group,
Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom

Research Report #713
March 1998

Tel : +44 (0)114 2225250
Fax : +44 (0)114 2731729
EMail : rrg@sheffield.ac.uk
Robotics Research Group

Hybridisation of Neural Networks and Genetic Algorithms in an Application of Time-Optimal Control

Nachol Chaiyaratana and Ali M S Zalzala

Robotics Research Group - Department of Automatic Control and Systems Engineering - University of Sheffield - Sheffield - UK

Abstract. This paper presents the use of neural network and genetic algorithms in the time-optimal control of a closed loop robotics system. Radial-basis function networks are used in conjunction with PID controllers in an independent joint position control to reduce tracking error. Genetic algorithm is then used to solve a multi-objective optimisation problem where decision variables are torque limits on each joint and 'the objective variables are trajectory time and position tracking error. This represents a task hybridisation between neural network and genetic algorithm. Two approaches with genetic algorithms are used to solve this optimisation problem: Multi-Objective Genetic Algorithm (MOGA) and genetic algorithm with weighted-sum approach.

1. Introduction

Time-optimal motion control has been one of the major research interests in robotics during the past decade. Time-optimality can lead to an overall improvement in the level of productivity from a manufacturing point of view and increase the effectiveness of task execution from an operational point of view. One particular aspect of research is the theory and application of time-optimal motion control along a specified path. An algorithm that can lead to time-optimality of this kind was firstly developed by Bobrow et al. (1985). Over the years, this algorithm has undergone a number of refinements and one of the latest modifications has been described by Shiller and Lu (1992). In summary, time-optimal motion of the end-effector of a robot arm along a specified path can be achieved when the motion is executed with either the maximum possible acceleration or deceleration along the path. This can be done when one of the actuators on the robot arm is always saturated and the other actuators adjust their torque values so that their torque limits are not violated.

Although this time-optimal motion control has been proved a useful algorithm in a number of applications, the majority of demonstrations have only been done in open-loop control. This can hardly be the case for a practical use of a motion controller in real-time implementations where closed-loop control would be a more common practice. Shiller et al. (1996) have pointed out that actuator dynamics and delays caused by an on-line feedback controller would lead to a reduction in efficiency of the algorithm when a feedback loop is also in use. Two possible methods have been used to solve this problem. The first method is based on a modification of the original time-optimal control problem into a time-energy optimal control problem which can be regarded as a Lagrangian constraint optimisation problem and can only be solved numerically (Shiller, 1996). The second method is based on the use of a simplified friction model to compensate for actuator dynamics and the implementation of a trajectory pre-shaping to account for the dynamics of the controller (Shiller et al., 1996).

In this paper, time-optimal motion control is used in a closed-loop robotics system. This is done in a similar way to that described in Shiller et al. (1996) except that actuator dynamics are not considered. However, neural network controllers are used in conjunction with standard controllers, which leads to the redundancy of the use of trajectory pre-shaping. A further multi-objective optimisation problem associated with the use of time-optimal control in a feedback system is also considered where the decision variables are torque limits on each joint, and the objective variables are trajectory time and position tracking error. Two approaches on multi-objective optimisation using genetic algorithm, namely weighted-sum approach and Multi-Objective Genetic Algorithm (MOGA) approach (Fonseca and Fleming, 1993), have been successfully used to solve this optimisation problem. This leads to a task hybridisation of neural network and genetic algorithm.

The paper is presented as follows. Time-optimal motion control as described by Shiller and Lu (1992) is briefly explained in section 2. In section 3, the control structure of the robotics system is discussed. Improvement in system performance due to the use of neural controller is also discussed in this section. Multi-objective optimisation using genetic algorithms and the multi-objective optimisation problem in this feedback robotics system are discussed in section 4. Simulation results from different approaches of multi-objective genetic algorithms are shown in section 5. Discussions on optimisation results are given in section 6. Finally, conclusions and possible further investigation are given in section 7.

2. Time-Optimal Motion Control Algorithm

The basic understanding on how to achieve time-optimal motion lies on two factors: ability to create a predefined smooth path and a well-defined description of robot dynamics. Shiller and Lu (1992) have explained that once a smooth path has been chosen, a consideration should be placed on a set of unique combinations of joint angles which makes the tip of the end-effector coincides with every point on the path. Then a function mapping of the following form is performed,

$$\theta(t) = \theta(s(t)), \qquad \theta \in R^n \tag{1}$$

where $\theta(t)$ is the joint angle at time t and $\theta(s(t))$ is the joint angle at point s on the path. Usually s is the distance from the initial position of the end-effector and is measured along the path.

Consider the dynamic equations of motion for an n-degree-of-freedom (n-dof) robot which are given by

$$\mathbf{D}(\theta)\ddot{\theta} + \mathbf{h}(\theta, \dot{\theta}) + \mathbf{c}(\theta) = \mathbf{T}(t) \tag{2}$$

where $\mathbf{D}(\theta)$ is the $n \times n$ inertial acceleration-related matrix,

$\mathbf{h}(\theta,\dot{\theta})$ is the $n \times 1$ centrifugal and Coriolis forces vector,

$\mathbf{c}(\theta)$ is the $n \times 1$ gravity loading force vector,

$\mathbf{T}(t)$ is the $n \times 1$ torque input vector,

$\theta(t)$ is the $n \times 1$ angular position vector,

$\dot{\theta}(t)$ is the $n \times 1$ angular velocity vector,

$\ddot{\theta}(t)$ is the $n \times 1$ angular acceleration vector,

n is the degree of freedom of robot model

and the dot denotes time derivatives.

By substituting expression given in (1) and its first and second time derivatives into (2), equation (2) can be simplified to

$$\mathbf{d}(s)\ddot{s}+\mathbf{h}(s)\dot{s}^2+\mathbf{c}(s)=\mathbf{T}. \tag{3}$$

The expression in (3) will be bounded by the actuator torque limits. In other words,

$$T^i_{\min} \le d_i(s)\ddot{s}+h_i(s)\dot{s}^2+c_i(s) \le T^i_{\max}, \quad i = 1, 2, \ldots, n \tag{4}$$

where $T^i_{\min}$ is the minimum limit of torque on joint i and $T^i_{\max}$ is the maximum limit of torque on joint i.

According to Shiller and Lu (1992), in order for the end-effector to be able to follow the path precisely, the tip velocity of the end-effector also has to be bounded by a certain limit. With some mathematical manipulation of equation (4), the bound on maximum possible velocity of the end-effector tip along the path is given by

$$\dot{s}_{\max}(s) = \min_{ij}\left\{\max_{T_iT_j}\left(\frac{d_j(s)(T_i-c_i(s))-d_i(s)(T_j-c_j(s))}{d_j(s)h_i(s)-d_i(s)h_j(s)}\right)\right\}, \quad i,j = 1, 2, \ldots, n. \tag{5}$$

Note that T_i can take either the value of $T^i_{\min}$ or $T^i_{\max}$, the minimum and maximum torque limits on joint i. With the use of equations (3) - (5), the control problem can be formulated as follows,

$$\min_{T\in\Omega} J = \int_0^{t_f} 1dt \tag{6}$$

where $\Omega = \left\{T \mid T^i_{\min} \le T_i \le T^i_{\max}; i = 1,2,\ldots,n\right\}$ (7)

and t_f is the free final time, subject to constraints

$$\ddot{s}_{\min}(s,\dot{s}) \le \ddot{s} \le \ddot{s}_{\max}(s,\dot{s}) \tag{8}$$

and

$$\dot{s} \le \dot{s}_{\max}(s). \tag{9}$$

Note that the constraint given in (8) is obtained by using equation (3) to map the actuator constraints in (7) to state dependent bounds on $\ddot{s}$. It can be seen that the optimisation problem is greatly simplified by parameterising the path with parameter s. The time-optimal control problem is reduced in structure from $2n$-dimensional state-space form into a two-state (s and $\dot{s}$) form. With the use of this control algorithm, open-loop torque profile and time-optimal joint position time history can be obtained. For a feedback robotics system, joint position time history is generally used as desired trajectory input to the position control loop, which is the case in the implementation presented in this paper.

3. Control Structure and the Neural Network Contribution

The control strategy that is used in this study is independent joint control. The control objective is to find a signal $\mathbf{u}(t)$ such that the overall system will be de-coupled into n linear second order systems. Freund (1982) has suggested such a control signal which take the form of

$$\mathbf{u}(t) = \mathbf{h}(\theta,\dot{\theta}) + \mathbf{c}(\theta) - \mathbf{D}(\theta)\begin{bmatrix} \alpha_{11}\dot{\theta}_1(t) + \alpha_{01}\theta_1(t) - \lambda_1 u_{ref}^{(1)}(t) \\ \vdots \\ \alpha_{1n}\dot{\theta}_n(t) + \alpha_{0n}\theta_n(t) - \lambda_n u_{ref}^{(n)}(t) \end{bmatrix} \tag{10}$$

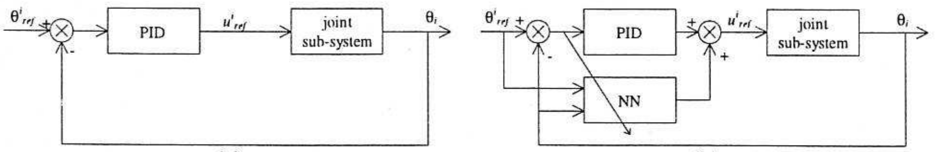
where α_{ij} and λ_i are arbitrary scalars. With the use of $\mathbf{u}(t)$ of this form where $\mathbf{u}(t)$ is input torque signal (equating $\mathbf{u}(t)$ to $\mathbf{T}(t)$), the overall dynamics of robotics system as described in equation (2) will transform into the following equation,

$$\ddot{\theta}_i(t) + \alpha_{1i}\dot{\theta}_i(t) + \alpha_{0i}\theta_i(t) = \lambda_1 u_{ref}^{(i)}(t)\,,\ i = 1, 2, \ldots, n \tag{11}$$

which indicates the de-coupled input-output relationships of the system. However, this exact de-coupling can only be achieved if the dynamics of the robot arm is exactly known. This would hardly be the case in the practical implementation. If modelling errors or disturbances exist in the control system, the input-output relationship described in equation (11) will deteriorate into

$$\ddot{\theta}(t) + \mathbf{f}(\theta,\dot{\theta}) = \mathbf{b}(\theta,\dot{\theta})\mathbf{u}_{ref}(t) \tag{12}$$

where $\mathbf{f}(\theta,\dot{\theta})$ and $\mathbf{b}(\theta,\dot{\theta})$ are unknown non-linear functions of angular position and velocity. In order to maintain the level of de-coupling between joints as high as possible, modelling errors and disturbances must be compensated for. Neural networks are known to be a good modelling tool and can be used in conjunction with any standard controller for enhancing control performance. A model-based neural network controller has been used to compensate for modelling errors and disturbances in this case. The control structure is discussed as follows.



(a) (b)

Fig. 1. (a) Independent joint controls using de-coupled control scheme.
(b) Neural network and PID controllers in de-coupled control scheme.

Consider an independent joint control scheme with the use of control signal given by equation (10) in conjunction with n PID controllers. The control loop for each joint can be shown schematically as the one in Fig. 1(a). A neural network can be used as another controller that will be used in parallel to a PID controller as shown in Fig. 1(b). This neural network will learn to compensate for modelling errors and disturbances.

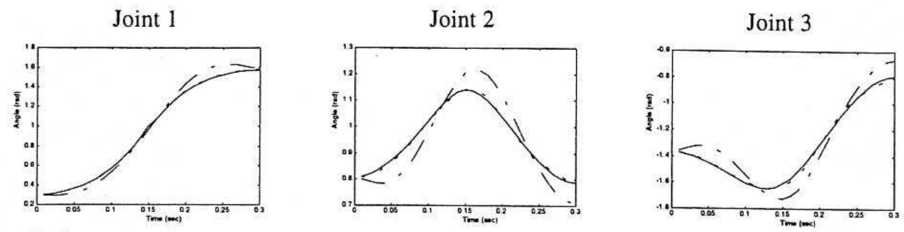
However, with the control scheme shown in Fig. 1(b), it is not possible to derive an exact desired neural network output training signal for the use in supervised learning algorithm. An alternative training signal must be acquired. In a standard supervised learning algorithm, the difference between actual neural network output and desired neural network output, the training error, is used to update the network parameters such as connection weights. In the case where neural network is placed in a closed loop like in Fig. 1(b), feedback error signal can be used to represent training error signal. This modified supervised learning scheme is called feedback error learning (Ziauddin and Zalzala, 1995).

In this study, radial-basis function network is used to assist PID controller in the position control loop. Position feedback error learning is used to train connection weights while centres of Gaussian functions are unsupervisedly trained. The training algorithm is similar to the one described by Fritzke (1994) except no new centre is incremented into the network during training and nearest neighbour concept is used instead of topological neighbour centre. Three networks are trained and tested in parallel for the use in position control loop of a 3-dof robot by using a time-optimal trajectory and its position feedback as the training and testing samples. Note that this time-optimal trajectory is obtained for a straight line path in Cartesian space with torque limits on joint 1, 2 and 3 of ±15, ±25 and ±5 Nm, respectively. The parameter settings for training neural network are summarised in Table 1.

Table 1. Parameter settings for training neural network.

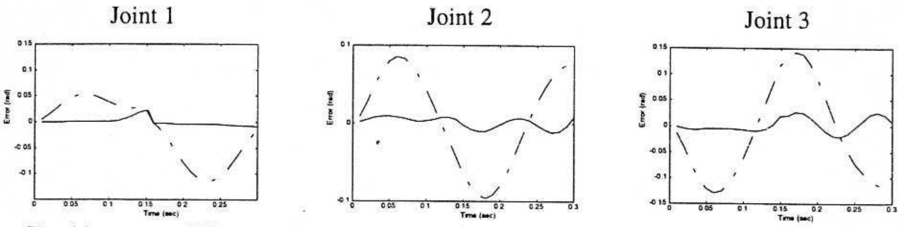
Parameter	Value
Number of Gaussian functions in each network	30
Number of connection weights in each network	30
Number of input nodes in each network	2
Number of output nodes in each network	1
Learning rate parameter (weights training)	0.001
Number of training samples	30
Number of training epochs	200

The simulation results for the case of PID controller with no torque limits and the case of PID and neural network controllers with torque limits are shown in Figs. 2, 3 and 4. In Figs. 1 and 2, the simulation results indicate that with the use of neural network controllers as assistants to PID controllers, significant improvements can be observed in tracking errors. In Fig. 3, with the use of neural network controllers, the characteristics of closed-loop torque profiles are similar to those of open-loop control. This indicates that time-optimality has been achieved within the torque constraints. Note that these trained radial-basis function networks are subsequently used in the following multi-objective optimisation problem without any further training.



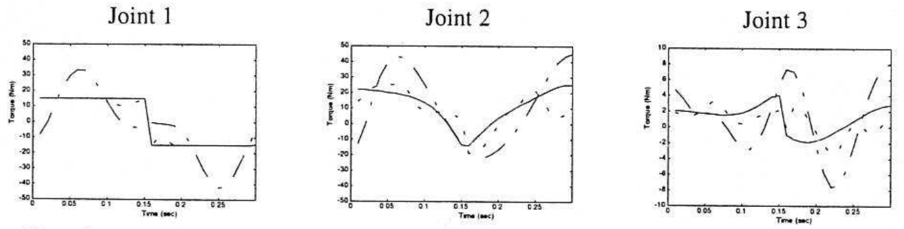
— Reference angle
-.-. Angle output - PID controller
... Angle output - PID + NN controller

Fig. 2. Angular positions from each joint.



-.-. Tracking error - PID controller
— Tracking error - PID + NN controller

Fig. 3. Tracking errors from each joint.



— Open loop torque
-.-. Closed loop torque - PID controller
... Closed loop torque - PID + NN controller

Fig. 4. Open-loop torque and closed-loop torque on each joint.

4. Genetic Algorithms and Multi-Objective Optimisation Problem

In practice, the maximum torque limits, which are used in time-optimal trajectory calculation process for a closed-loop control, are usually less than the actual torque limits on the actuator. This safety precaution is done in order to allow some margins of error for possible discrepancies introduced to the system by modelling errors and controller dynamics (Shiller et al, 1996). This implies that for a given set of actual torque limits of the actuators, there is a set of admissible torque limit combinations that can lead to a certain level of time-optimality within an acceptable range of tracking error. This leads to a design problem in robotics application in which the objective is to find a combination of torque limits from a set of admissible torque ranges which will lead to a trajectory which meets time-optimality and tracking error constraints. This is a multi-objective optimisation problem since it would be highly unlikely to obtain a single trajectory that can minimise both trajectory time and tracking error simultaneously. Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993) and genetic algorithm with weighted-sum approach has been used to solve the problem associated with torque limits selection in this study. The problem formation and genetic operator used are discussed as follows.

4.1. Decision Variables

In this paper, a 3-dof robot model with a task of tracking a straight line path in Cartesian space is used to study this multi-objective optimisation problem. Assuming that the moduli of maximum and minimum torque limits are the same for each actuator, the decision variables of a possible solution would be the magnitude of torque limits of each joint. In this study, the range of magnitudes of torque limit on joint 1, 2, and 3 are set to 15-30, 25-40 and 5-20 Nm, respectively.

4.2. Objective Variables

There are two optimisation objective variables in this problem: tracking error and trajectory time. Tracking error is expressed in terms of sum of mean absolute value over three joints, calculated over the whole trajectory. Trajectory time is the optimal time obtained from the motion control algorithm described in section 2. Note that the sampling period used in the simulation of this 3-dof robotics system closed-loop control is 0.01 sec. The trajectory time will always be in the form of $0.01m$, where m is a positive integer.

4.3. Chromosome Coding

Three decision variables, magnitudes of torque limit of each joint, are concatenated together and coded using Gray code to form a chromosome. The torque ranges of all three joints are discretised using a search step of 0.5 Nm. This leaves 31 search points for magnitude of torque limit of each joint which can be coded with a Gray code of length 5. The total length of chromosome in this case is 15. Note that there are certain search points obtained after decoding the chromosome which lie outside the required search space. These points are mapped back into the feasible region by changing the most significant bit of the Gray code section representing particular decision variable that violates feasible constraint into zero.

4.4. Selection Method

Stochastic universal sampling is used in fitness selection. The elitist strategy used is to select two individuals with the highest fitness and pass on to the next generation without crossover or mutation.

4.5. Fitness Sharing, Crossover and Mutation Methods

Fitness sharing with the use of triangular sharing function is done in the implementation of MOGA. Standard one-point crossover is used in recombination. Two individuals are allowed to perform crossover if they are within mating restriction distance from each other. In this study, mating restriction radius is set as equal to the sharing radius. Standard bit-flipped mutation is in use.

As mentioned earlier, two approaches to multi-objective optimisation has been investigated. In the MOGA, fitness is assigned to each individual based on its rank and a linear function is used for fitness interpolation. Fitness sharing and mating restriction is done in normalised objective variable space. The second approach is a weighted-sum, where objective functions are weighted and added together to form a single objective. This will force the algorithm to concentrate the search on only one area of the Pareto front. For this reason, fitness sharing is not implemented in the weighted-sum approach. However, mating restriction is still in use to discourage the production of lethal individuals after crossover. The parameter settings for both MOGA and weighted-sum approaches are summarised in Table 2.

Table 2. Parameter settings for genetic algorithms.

Parameter	Value	Parameter	Value
Chromosome length	15	Mating restriction radius	0.003
Crossover probability	0.8	Population size	30
Mutation probability	0.07	Number of elitist individuals	2
Sharing radius (MOGA only)	0.003	Number of generations	30

5. Simulation Results

Two case studies are investigated in this paper. The aim of the first case study is to find a set of torque limits which leads to trajectories with mean absolute tracking error $\leq$ 0.10472 radians (2 degrees) and trajectory time $\leq$ 0.27 sec. The aim of the second case study is to find a set of torque limits which leads to trajectories with mean absolute tracking error $\leq$ 0.05236 radian (1 degree) and trajectory time $\leq$ 0.3 sec. The purpose of the first case study is to find solutions that concentrate more on optimising the trajectory time while the second case study emphasises on tracking error optimisation. The simulation results for these two cases are summarised in Table 3 - 6. Note that non-dominated individual results from genetic algorithm with weighted-sum approach are obtained by applying the ranking method described by Fonseca and Fleming (1993) to the last generation of the population.

Table 3. Results from MOGA (case 1).

Decision Variable			Objective Val.	
T_1	T_2	T_3	Error	t
28.5	39.5	14.0	0.08230	0.22
26.0	38.5	20.0	0.06267	0.23
24.0	38.5	19.5	0.05264	0.24
24.0	39.5	18.0	0.05264	0.24
21.5	39.5	18.5	0.04925	0.25
20.0	37.0	7.5	0.03465	0.26
18.5	33.0	5.0	0.03085	0.27

Table 4. Results from WSGA (case 1).

Decision Variable			Objective Val.	
T_1	T_2	T_3	Error	t
27.5	40.0	18.5	0.08595	0.22
26.0	40.0	18.5	0.06129	0.23
23.5	37.5	20.0	0.05302	0.24
22.0	36.5	11.0	0.04251	0.25
20.0	31.5	11.0	0.03674	0.26
18.5	33.0	16.0	0.03044	0.27

weight on mean absolute tracking error objective = 1.0
weight on trajectory time objective = 1.0

Table 5. Results from MOGA (case 2).

Decision Variable			Objective Val.	
T_1	T_2	T_3	Error	t
22.0	38.0	18.0	0.04248	0.25
22.0	38.5	17.5	0.04248	0.25
20.0	36.5	16.0	0.03465	0.26
20.0	38.0	17.0	0.03465	0.26
20.0	38.0	17.5	0.03465	0.26
18.5	32.5	9.0	0.03056	0.27
17.5	27.5	5.0	0.02814	0.28
16.0	35.0	16.0	0.02246	0.29
16.0	39.0	7.0	0.02246	0.29
15.0	28.5	18.5	0.01971	0.30
15.0	28.5	19.0	0.01971	0.30

Table 6. Results from WSGA (case 2).

Decision Variable			Objective Val.	
T_1	T_2	T_3	Error	t
22.0	38.5	5.5	0.04465	0.25
18.5	36.0	12.5	0.03038	0.27
16.0	35.5	7.0	0.02246	0.29
16.0	39.0	11.5	0.02246	0.29
15.0	27.0	19.5	0.01970	0.30

weight on mean absolute tracking error objective = 10.0
weight of trajectory time objective = 0.1

In the above tables, T_1, T_2 and T_3 are magnitudes of torque limits on joint 1, 2 and 3, respectively. The heading "Error" represents sum of mean absolute tracking error over three joints, calculated over the whole trajectory and t denotes trajectory time. In all cases, an algorithm must produce a set of six unique solutions in order to cover the Pareto front of each case. However, more than one combination of torque limits can result in the same time-optimal trajectory as indicated in Table 3, 5 and 6.

6. Discussions

In the first case study, both MOGA and genetic algorithm with weighted-sum approach can produce results that cover the Pareto front and meet all the constraints.

This means that the performances from both approaches are similar for the first case. However, in the second case study, genetic algorithm with weighted-sum approach fails to locate two possible solutions - the ones with trajectory times of 0.26 and 0.28 sec. Even with this, the weighted-sum approach is still capable of locating solutions at both ends of the Pareto front (the ones with trajectory times of 0.25 and 0.30 sec). The explanation of these results is given as follows.

In the first case study, the values of weighted-sum objective of the non-dominated solutions are within the same range. Weighted-sum approach does not have any difficulties in finding the solutions that can cover the whole Pareto front. However, in the second case study, the weighted-sum objective values of non-dominated solutions are more diverse. With the current weight setting, weighted sum approach will concentrate the search on the area of solutions with trajectory times of 0.29 and 0.30 sec since these will give the smallest weighted-sum objectives. On a close inspection, it can be seen that the solutions with trajectory times of 0.25 and 0.27 sec from weighted-sum approach have higher weighted-sum objective values than the rest of the solutions. These solutions would be the results of less fit individuals from the last generation which were picked out by the ranking mechanism. Overall, it can be said that for this particular application, MOGA can produce better results than genetic algorithm with weighted-sum approach.

7. Conclusions and Further Works

In this paper, genetic algorithms have been used to solve a multi-objective optimisation involving the selection of torque limits subject to time-optimality and tracking error constraints for one predefined path. These torque limits are employed in the operation of described model-based neural controller, thus achieving closed-loop control of the robotics system. The combination of the model-based neural controller and genetic algorithm optimisation has allowed an implementation of closed-loop time-optimal control without the use of trajectory pre-shaping which has not been possible before in earlier literature.

This work can be extended to cover the application of finding a collision-free path in workspace environment as described in Shiller and Dubowsky (1991) and Fiorini and Shiller (1996) where neural networks can be used to assist any standard controllers in a closed-loop robotics system and genetic algorithms are used to perform an optimisation task as described in this paper.

References

1. Bobrow, J. E., Dubowsky, S., Gibson, J. S.: *Time-Optimal Control of Robotic Manipulators Along Specified Path.* International Journal of Robotics Research (Fall 1985), 4 (3) (1985) 3-17
2. Fiorini, P., Shiller, Z.: *Time optimal trajectory planning in dynamic environments.* In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN (April 22-28, 1996). IEEE, New York, NY, 2 (1996) 1553-8

3. Fonseca, C. M., Fleming, P. J.: *Genetic algorithms for multiobjective optimization: formulation, discussion and generalization*. In: Genetic Algorithms: Proceedings of the Fifth International Conference, Urbana-Campaign, IL (July 17-22, 1993). Morgan Kaufmann, San Mateo, CA (1993) 416-23
4. Freund, E.: *Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators*. International Journal of Robotics Research (Spring 1982), 1 (1) (1982) 65-78
5. Fritzke, B.: *Growing Cell Structures - A Self-Organizing Network for Unsupervised and Supervised Learning*. Neural Networks, 7 (9) (1994) 1441-60
6. Shiller, Z.: *Time-Energy Optimal Control of Articulated Systems with Geometric Path Constraints*. Transactions of ASME. Journal of Dynamic Systems, Measurement and Control (March 1996), 118 (1) (1996) 139-43
7. Shiller, Z., Chang, H., Wong, V.: *The Practical Implementation of Time-Optimal Control for Robotic Manipulators*. Robotics and Computer-Integrated Manufacturing (March 1996), 12 (1) (1996) 29-39
8. Shiller, Z., Dubowsky, S.: *On Computing the Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles*. IEEE Transactions on Robotics and Automation (December 1991), 7 (6) (1991) 785-797
9. Shiller, Z., Lu, H.-H.: *Computation of Path Constrained Time Optimal Motions with Dynamic Singularities*. Transactions of ASME. Journal of Dynamic Systems, Measurement and Control (March 1992), 114 (1) (1992) 34-40
10. Ziauddin, S. M., Zalzala, A. M. S.: *Model-Based Compensation and Comparison of Neural Network Controllers for Uncertainties of Robotic Arms*. IEE Proceedings. Control Theory and Applications (September 1995), 142 (5) (1995) 501-507.