Online Neurocontrol Design Optimized by a Genetic Algorithm for a Multi-trailer System

Endusa MUH ANDO*, Hiroshi KINJO*, Eiho UEZATO*, Tomonobu SENJYU* and Tetsuhiko YAMAMOTO*

In this paper we present real-time neurocontrol system for a nonlinear dynamic plant. In order to improve the training and control performance we present a combined system with a neurocontroller (NC) and a linear quadratic regulator (LQR). We apply the control scheme to the backward control of a multi-trailer system. The function of the LQR is to cater for the linear part of the system thereby alleviating the load on the NC. An emulator of the plant is used to design the desired trajectory. The actual plant is subsequently run on this path. In the event that the plant fails to trace the desired trajectory, the control system is re-designed from this point and a new trajectory formulated. We utilize the GA to update the NC weights, while the evaluation function of the NC incorporates both the squared errors and the running steps errors; the latter having the function of realizing faster training of the NC. We have significantly reduced the computation time by utilizing one pattern training for the NCs in real time. Simulations show that the proposed online method has good control performance for the trailer truck system.

Key Words: adaptive neurocontrol, multi-trailer system, online design, back-up control, GA training

1. Introduction

Control of the trailer truck system is known to be one of the highly non-linear, multi-variable and unstable control problems. For this control problem we must avoid out of control states, characterized by the "jack-knife" phenomenon, and exceeding the physical limitation of the steering angle. Many control systems for the problem have been proposed. Astolfi et al. 1) have used Lyapunov techniques in the design of a path-tracking controller for the tractor-trailer vehicle. Sampei et al. 2) have presented a controller design for articulated vehicles to follow arbitrary paths by use of exact linearization. Bolzern et al. 3) have used an input-output linearization approach to the control of an n-body autonomous vehicle.

Concerning soft computing techniques, earlier works by Tanaka et al. 4) have shown that fuzzy methods exhibit good control performance. With regard to the neurocontrol system, Nguyen and Widrow 5) reported the pioneering work for the neurocontrol application on the trailer-truck system. They successfully designed a controller utilizing a back propagation (BP) algorithm. Jenkins and Yuhas 6) have presented a small-sized NC, also based on BP. In previous studies, Kinjo et al. proposed various control methods for a single trailer-truck combination, for a truck connected to two trailers and for a five trailer-truck combination, using NCs evolved by GA 7), 8).

In this study our main focus is on real-time training of the NCs with the aid of a real-coded GA and the LQR, and to confirm the performance of such a control system when applied to the back-up movement of a trailer-truck system. Previously, our research centered on a batch system, where the design of the control system is executed off-line. The off-line GA training is computationally expensive as it requires many training patterns prior to determining the best NC. Based on our previous work 7), 8) and having investigated backing up with up to 5 trailers, we established that it becomes very difficult to design the control system as the control variables increase proportionately with the number of trailers. Realizing that these offline techniques diminish in efficiency with increase in trailers, we have attempted to simulate on-line optimal controller design. This is achieved by using an emulator of the plant to determine the trajectory. A controller is then designed and utilized for control of the actual plant from only one initial configuration pattern. Our control system comprises the LQR and the NC. Essence of the LQR is to ensure faster training of the NC thereby reducing computational time.

We employ an interval-shemata crossover technique 9) on a real-coded GA to update the feed-forward neural
network weights. The evaluation function of the NC in the design has been modified from the conventional type to incorporate both the squared errors and the running steps errors. Significance of the running steps error is to enhance faster training of the NC. In this paper, we propose a monitoring system for the trajectory traced by the actual plant. If there exists a difference between the trajectories of the emulator and the actual plant, the scheme adaptively modifies the design and determines a new trajectory. The dynamic optimization algorithm utilizes real-time acquired data.

The remainder of this article is organized as follows. The model of the trailer-truck system is presented in Section 2. Section 3 explains the control scheme for the online training. Section 4 describes the monitoring system. Simulation results are presented in Section 5 and a discussion on performance of the scheme follows in Section 6. Finally, we draw our conclusions and perspectives in Section 7.

2. Modeling and Problem Formulation

2.1 The Trailer-truck Model

Fig. 1 details the geometry of our control object: the truck with seven trailers and its orientation in a coordinate base. Table 1 explains the trailer-truck system parameters. The steering angle $u(t)$ is the input to the plant and is determined by the state variables. Our model is considered as a discrete system such that samples of the trailer-truck movement are observed at constant sampling time $\Delta t$. The kinematics of both the emulator and the actual plant are described by the set of equations (1)~(17).

These equations have been extended on the basis of the set earlier presented by Tanaka et al.4) The equations describe a system that has nonlinear characteristics hence linear control theory can not be applied directly.

2.2 Problem Formulation

The multi-trailer system is to be backed-up at a constant velocity $v$ on a desired trajectory to a given destination. The steering angle $u(t)$ is to be controlled such that the system is asymptotically stabilized along the straight line $x_{15}(t)=0$. This requires that the relative angles, angle of last trailer and the vertical, position are kept to a

Table 1 Parameters of the trailer-truck system

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l$</td>
<td>Truck length</td>
</tr>
<tr>
<td>$L$</td>
<td>Trailer length</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Sampling time</td>
</tr>
<tr>
<td>$v$</td>
<td>Speed of system</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>Steering angle</td>
</tr>
<tr>
<td>$x_0, x_2, x_4, x_6, x_8$</td>
<td>Angles of truck, 1st, 2nd, 3rd, 4th, 5th, 6th and 7th trailers</td>
</tr>
<tr>
<td>$x_{10}, x_{12}$ and $x_{14}$</td>
<td>Angles of truck, 1st and 2nd trailer</td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>Vertical position of 7th trailer</td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>Horizontal position of 7th trailer</td>
</tr>
</tbody>
</table>
minimum, thus
\[ X(t) = [x_1(t), x_3(t), x_5(t), x_7(t), x_9(t), x_{11}(t), x_{13}(t), x_{14}(t), x_{15}(t)]^T \to 0. \] (18)

3. Control Scheme

3.1 Control System

Fig. 2 shows the control system. The plant to be controlled is the trailer-truck system; it receives the steering command \( u(t) \) and outputs the state variables \( X(t) \). The controller consists of a combination of the LQR for the linear part, and the NC for the nonlinear part of the system. Both LQR and NC receive the error of angles \( x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{14} \) and position \( x_{15} \) as inputs and output the steering angles \( u_L(t) \) and \( u_N(t) \), respectively.

The control input to the plant is the steering command \( u(t) \), comprising both outputs:
\[ u(t) = u_L(t) + u_N(t). \] (19)

Considering that this is a regulator problem, necessarily, the reference states \( X_{ref} = 0 \). GA denotes the genetic algorithm procedure used to train the NCs and determine the connecting weights.

3.2 LQR Controller

We used the LQR from control theory \(^{10}\) to solve the linear part of the design in which the state is accessible. The stochastic formulation of the LQR design problem for this system is linearized and described by
\[ X(t+1) = AX(t) + Bu_L(t) \] (20)
where \( A \) and \( B \) are linearized parameters of the nonlinear system described by Eqs. (1)-(17). \( A \) and \( B \) are obtained from the assumption that the control input, angular differences and angle of last trailer have magnitudes:
\[ |u_L, x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}, x_{14}| \ll 1. \] (21)

The obtained parameters are as follows \(^4\):

\[
A = \begin{bmatrix}
1 & \frac{-v \Delta t}{L} & 0 & 0 & 0 & \cdots & 0 \\
\frac{v \Delta t}{L} & 1 & \frac{-v \Delta t}{L} & 0 & 0 & \cdots & 0 \\
0 & \frac{v \Delta t}{L} & 1 & \frac{-v \Delta t}{L} & 0 & \cdots & 0 \\
0 & \cdots & 0 & \frac{v \Delta t}{L} & 1 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \frac{(v \Delta t)^2}{2L} & \frac{2L}{v \Delta t} & 1
\end{bmatrix}
\] (22)

\[ B = \begin{bmatrix}
v \Delta t/\ell \\
0, \cdots, 0
\end{bmatrix}^T. \] (23)

The cost function is the sum of both steady-state mean-square weighted state \( X(t) \) and actuator signal \( u_L(t) \)
\[ J = \sum_{t=0}^{\infty} \{X(t)^T \dot{W} X(t) + w u_L(t)^2 \} \] (24)
where \( W \) is a positive semi-definite weight matrix and \( w \) is a scalar quantity. One of the methods of LQR design is by use of the system gain to control a feedback system. The gain values of the linear combiner are calculated through some state-feedback technique, in this case, by use of the Riccati equation. Thus, discrete system control output is
\[ u_L(t) = -Gx(t). \] (25)

3.3 Construction of the NC

We used a 3-layer, 9-5-1 configuration neural network with nonlinear activity functions for the hidden and output layers. For the hidden layer we utilized the Sigmoid function:
\[ f(x) = \frac{1}{1 + e^{-x}} \] (26)
while the activity function of the output layer is a cubic,
\[ f(x) = ax^3. \] (27)

Generally when the magnitude of the state variables \( X(t) \) are small, the control object tends to be linear and this cubic function facilitates attainment of the desired output, \( u_N(t) = 0 \).

3.4 Training of the NCs

The methodology of training the NCs is as follows. The trailer-truck is set to an initial position and orientation. Then, initial connecting weights of the NCs are randomly set. By use of an emulator of the plant, the desired trajectory is designed by utilizing the GA. In the GA procedure, control simulations using the plant emulator are performed and NCs are evaluated. The fitness function
of the NCs is described in the next subsection. Based on the control performance, NCs are optimally evolved toward the best controller.

We apply a real-coded GA in the design of the NC. It is considered that a real-coded GA is faster than a bit-string GA because there is no need to convert and reconvert processes between real values and chromosomes in the GA procedure. Our GA utilizes an interval-schemata technique, the Blend crossover (BLX) scheme\(^9\), for adjustment of the NC connecting weights. The BLX scheme uniformly picks values that lie between two points that either contain the two parents, or extend equally on either side determined by a user specified GA parameter. Fig. 3 shows the mechanism of breeding the connecting weights, where \(w_{o1}\) and \(w_{o2}\) denote the produced connecting weights based on the parent weights \(w_{p1}\) and \(w_{p2}\).

Fig. 3 Production process for the connecting weights using BLX scheme

3.5 Fitness Function of the NC

We use the plant emulator to design the NC and determine the trajectory. Control simulation is performed by using the NC. The fitness function \(F\) is utilized for the Roulette wheel selection of parents for the evolution of the NCs and is computed as

\[
F = 1/E
\]  
(28)

where the evaluation function \(E\) relates to the cumulative error generated by the deviation in movement along the trajectory for each manoeuvre. The evaluation function is defined as

\[
E = \alpha E_S + \beta E_T
\]  
(29)

where \(E_S\) denotes the squared error, \(E_T\) the running steps error, and \(\alpha, \beta\) are associated weights.

The first term \(E_S\) is an evaluation of squared errors:

\[
E_S = q_3(x_{ref}^3 - x_{end}^3)^2 + q_4(x_{ref}^4 - x_{end}^4)^2
\]  
(30)

where \(x_{ref}\) is the reference variable and \(x_{end}\) is the final value of the state variable that starts from the initial configuration. The factor \(q\) adjusts the importance of the control variables.

The second term \(E_T\) refers to the running steps error on completion of the control simulation regardless of whether the plant covers all the steps or stops after attaining the out of control states. \(E_T\) is described by

\[
E_T = t_{max} - t
\]  
(31)

where \(t_{max}\) is the maximum allowable design steps and \(t\) refers to the running steps from the initial position along the desired trajectory. Ideally, \(E_T = 0\) when the emulator does not exhibit the out of control state.

On-line training and design ensure that the best NC for the desired trajectory is realized. Through GA training and subsequent evolution, eventually \(E_T = 0\) and \(E_S\) is minimized to small values. Generally, a NC is considered highly evolved when \(E_S\) is very small and a large quantity of running steps on the desired trajectory is attained.

4. Monitoring System

In order to reduce the computation time for real-time control, we trained the NC for only one pattern i.e. starting initial value sets to the current initial angles and position. However, the NC trained by one pattern has less generalization ability. So we applied a monitoring system for the control trajectory in the actual plant.

Fig. 4 shows the monitoring and re-design sequence. From the initial position, the NC is trained to control the plant using an emulator and designs a desired trajectory no. 1. After training and determining the trajectory, the NC controls the actual plant as the monitoring system.
checks the trajectory. If the plant tracks a different trajectory, the system is re-designed say, at position no. 2 so as to trace trajectory no. 2. In case trajectory no. 2 is not conformed to, the system is re-designed at position no. 3 and a new trajectory, no. 3, designed for. The learning continues as the emulator and controller improve as they track the physical process. All these training and re-design are done in real-time so that the plant is controlled by minimizing the associated errors. The flow chart in Fig. 5 depicts the process.

![Flow Chart](image)

**Fig. 5** Training and design algorithm

Process 1: Set the vehicle at the initial configuration.
Process 2: Design the controller and determine the trajectory.
Process 3: Run the actual plant and emulator simultaneously while observing the difference index $D$:

$$D = (X_M - X)^T (X_M - X)$$

where the state variables are, respectively, $X_M$ for the emulator and $X$ for the actual plant.
Process 4: If the plant does not trace the designed trajectory, that is, when $D > D_T$, goto Process 2. $D_T$ is the threshold value of the index $D$.
Process 5: If the plant does not reach the desired destination, goto Process 3.
Process 6: End the control.

5. Simulations and Results

5.1 Parameters

In order to confirm the effectiveness of our proposed scheme we utilized a seven trailer-truck configuration with parameters as shown in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck length, $l$</td>
<td>0.3m</td>
</tr>
<tr>
<td>Trailer length for design, $L_M$</td>
<td>1.0m</td>
</tr>
<tr>
<td>Trailer length for actual plant, $L$</td>
<td>0.7m &amp; 1.2m</td>
</tr>
<tr>
<td>Velocity of system, $v$</td>
<td>-0.2m/s</td>
</tr>
<tr>
<td>Sampling time, $\Delta t$</td>
<td>0.25a</td>
</tr>
</tbody>
</table>

In solving the Riccati equation, the diagonal of the matrix $W$ in Eq. (24) may be set variously to achieve the optimum gain. In our case $W$ is weighted such that $W=\text{diag}[1,1,1,1,1,1,1,1,100]$, and $w=100$. These yield the following values of the gain $G$, for use in determining $u_L(t)$ as given in Eq. (25):

$$G = [-4.58, 30.49, -115.07, 267.69, -387.73, 329.84, -133.47, 4.14, -0.68]$$

To ensure the physical limitations of the system are not exceeded, we set $|u(t)| < \pi/2 \text{ rad}$. However, some of the $G$ term values may be large thus by Eq. (25) the LQR output $u_L(t)$ may exceed the limit imposed in Eq. (21).

In constructing of the NCs, the output neuron function with good performance was a cubic, $f(x) = ax^3$. We determined and set $a=0.1$ in Eq. (27) by trial and error.

In the implementation of the real-coded GA, we utilized a BLX with a range factor of 0.8. Table 3 shows the other GA properties.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>50</td>
</tr>
<tr>
<td>No. of offspring</td>
<td>30</td>
</tr>
<tr>
<td>Selection scheme</td>
<td>Roulette wheel</td>
</tr>
</tbody>
</table>

For the evaluation function as given in Eq. (29), we set $\alpha = 1.0$ and $\beta = 1.0$. In Eq. (30) we set $q_1 = q_3 = q_5 = q_7 = q_9 = q_{11} = q_{13} = q_{14} = 1.0$ and $q_{15} = 0.1$. We set the design running steps limit in Eq. (31), $t_{\text{max}} = 600$ steps. We set the threshold value, $D_T = 0.05$ for the difference index $D$.

For the evolutionary training in designing the NCs the stop condition is executed either when $E$ is less than $0.001$ or training generation times are over $500$.

5.2 Guideline for Selection of Parameters

Though determination is by trial and error, choice of parameters is crucial in designing a high performance controller. Firstly we design the LQR: weights $W$ and $w$ in (24) are established by tuning the gain $G$ to get good control results. After that, we train the NC and systematically determine $a$, $\alpha$, $\beta$, $q_1 \sim q_{15}$, $t_{\text{max}}$ and $D_T$. The coefficient $a$ in (27) is chosen based on the expected magnitude of the output neuron function. In (29), typically
\( \alpha = 1.0; \) if \( \beta = 0.0 \) then the evaluation function is purely the squared error. We set \( \beta = 1.0 \) to effect faster NC training. Realizing that angular deviations are critical to the jack-knife aspect, we assign \( q_1 - q_{14} = 1.0. \) We set \( q_{15} = 0.1 \) as a discounting factor to realize uniform summation of distance with angular deviations. \( t_{max} \) is based on the target destination distance. \( D_T \) is selected on the basis of a compromise between training frequency and smooth trajectory execution.

5.3 Control Results I

Fig. 6 Design performance for the emulator, trailer length \( L=1.0 \) m. Display interval: 150 time steps

Fig. 6 shows movement along the desired trajectory when the length of the actual plant \( L = 1.0 \) m. The initial configuration for the design is set at vertical position \( x_{15} = 8.0 \) m and orientation of \( \pi/2 \) rad.

Fig. 7 is an example of the simulation results, for the actual plant when \( L = 1.2 \) m, where the initial orientation and starting vertical position are set to \( \pi/2 \) rad and 8.0 m respectively. It is observed that the actual plant successfully backs up along \( x_{15} = 0 \) m.

Fig. 8 shows that the proposed adaptive controller optimally controls the state variables. Fig. 8(a) shows variation of the relative angles \( (x_1, x_3, x_5, x_7, x_9, x_{11}, x_{13}) \) and angle of last trailer \( x_{14} \) with running steps. We see that the angles are controlled successfully by the adaptive controller to within the set range \( (-\pi/2, \pi/2) \) rad. Fig. 8(b) gives the vertical position \( x_{15} \) of the last trailer. The trailer smoothly reaches the desired path with no zig zag.

Fig. 9 shows training results for the design frequency. Fig. 9(a) shows variation of the difference index \( D \). When \( D > D_T \) there is need to retrain the NC and determine the new trajectory in real time. From Fig. 9(b) we note that on-line re-design takes very few generations.

In Fig. 10 we see the variation of \( u_L(t), u_N(t) \) and \( u(t) \), the LQR, NC and combined controller outputs respectively, with running steps. We observe that the LQR output is initially high but stabilizes after a while. The output \( u(t) \) does not exceed the set range \( (-\pi/2, \pi/2) \) rad.
since the high value of $u_L(t)$ is suppressed by the NC. At about 220 running steps $u_N(t)$ is practically zero and only the LQR is effective.

5.4 Control Results II

We simulated for an actual plant, trailer length $L = 0.7$ m. Starting configuration is at vertical position 5.0 m and orientation 0 rad. As seen from Fig. 11, the plant is successfully stabilized along the line $x_{15} = 0$ m. Fig. 12 confirms that the state variables are controlled to within required limits. Fig. 13 shows the difficulty in designing for this pattern; it requires several generations to evolve the controller initially. However, after several training and re-designs, $D$ falls below the threshold and the system is eventually fully controlled along the trajectory in 270 steps. Fig. 14 confirms that the steering command $u(t)$ is controlled successfully to within the limit $(-\pi/2, \pi/2)$ rad.
5.5 Control Performance

Table 4 shows the control performance of the three controllers in 500 trials for several trailer truck systems. The initial conditions are randomly set in the ranges [0, 6] m for the starting vertical position and [0, π/2] rad for the initial orientation. The criterion for success in control is limiting the squared errors to less than 0.001. It can be seen that the combination of NC with LQR has better control performance than that of either the NC or the LQR. We can observe that the parameter β = 1.0 is essential in the NC training.

Table 5 shows control performance when there exists parameter discrepancy between actual plant and plant emulator. We can observe that the proposed monitoring system and re-design method is effective for the parameter discrepancy.

5.6 Merits of Proposed Control Method

Our proposed control scheme is able to realize online control of seven trailers, despite complexity in control rendered by the nine control variables. This is due to the combined controller and also the evaluation function, a characteristic of our system that ensures faster training of the NCs. Further, the adaptive nature of the controller design ensures the computation time is significantly reduced.

The NC design utilizes only forward calculation, hence it is applicable in cases where derivative functions of the plant are hard to obtain. Previous approaches employed BP methods that require derivatives of the plant.

Our control scheme is applicable to both continuous and discrete systems.

6. Discussion

In this paper we examine the problem of backing the truck and trailer to a target location by on-line design of the controller for the plant. Previous research has shown that controller design gets difficult with increase in connected trailers. Further, the batch system of control design is not effective in evolving a high performance controller as trailers increase. Our proposed adaptive scheme yields a suitable controller for a seven trailer vehicle and reduces computation time significantly.

Complexity in control arises from the tendency for the plant to jack-knife when angular differences are over π/2 rad. Further, when the control input is over π/2 rad then the steering command exceeds the physical limitation of the system.

We employed three methods in order to shorten the NC training times. Firstly, only one training pattern is utilized; due to the neural network generalization ability, the online monitoring and re-design of the system is easily achieved. Secondly, we distribute the control problem

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Success rate of control (α = 1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of trailers</td>
<td>NC with LQR</td>
</tr>
<tr>
<td>β = 1.0</td>
<td>β = 0.0</td>
</tr>
<tr>
<td>4</td>
<td>99.2</td>
</tr>
<tr>
<td>5</td>
<td>93.6</td>
</tr>
<tr>
<td>6</td>
<td>78.6</td>
</tr>
<tr>
<td>7</td>
<td>54.8</td>
</tr>
</tbody>
</table>
such that the LQR handles the linear part while the NC is purely for the nonlinear part. Lastly, the evaluation function is modified to include running steps errors.

Having successfully designed the controller based on the emulator, we run the actual plant with similar parameters. Fig. 6 shows that, in the absence of parameter discrepancy, the trailer-truck system successfully tracks the trajectory. However, when the trailer length $L$ changes or even a small variation in $D$ exists, significant offsets from the designed path may result. The proposed scheme makes use of real-time acquired data to successively retrain the NCs and generate suitable trajectories for the actual plant.

In running the actual plant, the $D$-factor that has been introduced determines whether the controller is able to guide the plant along the desired trajectory. On exceeding the threshold value $D_T$, there is need to instantaneously retrain the NCs and re-design for another trajectory. This on-line process is seen to be very effective. The instantaneous input command, $u(t)$, controls the trajectory of the truck and trailers.

Fig. 7 and 11 show the effectiveness of adaptive design in steering actual plants, trailer lengths $L=1.2$ m and $L=0.7$ m respectively, along the desired path while backing up. Fig. 8 and 12 show that the state variables are successfully controlled in minimal steps to conform to the reference variables. Fig. 9 and 13 show that to successfully evolve the best controller, repeated on-line training is necessary, else the system gets out of control. From Fig. 9a and Fig. 13a $D_T$ is a compromise of antagonistic demands. If $D_T$ is large then there is low training frequency hence difficult to effect redesign in ample time, while if $D_T$ is small then there is high frequency of re-design, which interferes with the smooth running of the system.

We note from Fig. 10 and 14 that $u_L$ is suppressed by $u_N$. Generally, the NC output falls to zero more rapidly than the LQR output; from this point onward only the LQR is active, thus $u(t) = u_L(t)$. The problems of out of control steering and jack-knife locks are eliminated since all the relative angles are kept to within $(-\pi/2, \pi/2)$ rad.

The effectiveness of the real-time design scheme is demonstrated by the fact that though the complexity of the control problem increases with the number of connected trailers, it takes very few generations to evolve the controller, hence very few calculations. The scheme yields a very powerful and robust controller that operates optimally to enable the actual plant execute the trajectory.

### Table 5

<table>
<thead>
<tr>
<th>No. of trailers</th>
<th>Success rate [%]</th>
<th>Re-design</th>
<th>Without re-design</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$L = 0.7m$</td>
<td>91.4</td>
<td>92.8</td>
</tr>
<tr>
<td>5</td>
<td>$L = 1.2m$</td>
<td>70.6</td>
<td>71.6</td>
</tr>
<tr>
<td>6</td>
<td>$L = 0.7m$</td>
<td>53.8</td>
<td>55.2</td>
</tr>
<tr>
<td>7</td>
<td>$L = 1.2m$</td>
<td>37.4</td>
<td>37.8</td>
</tr>
</tbody>
</table>

### 7. Conclusion

In this research we have attempted to verify how neurocontrol can be improved by, first, utilizing a real-coded GA and a modification of the evaluation function to include running steps for effective NC training, and secondly, employing a LQR together with the NC to boost performance. We have applied this control scheme in the control of the backward movement of a multi-trailer system on-line, and have presented simulation results to support the successfulness of our proposed approach. We have shown that our system is able to adaptively learn very quickly, enable the design of the system based on on-line acquired data, and effectively control the real plant along a predetermined trajectory without any annoying zig zag or jack-knife locks. The paradigm is rich with possibilities for further study.

### References

8. A. Kiyuna, H. Kinjo, K. Kurata and T. Yamamoto: Con-


Endusa Muhando

He received the M. Eng. degree from the University of the Ryukyus in 2005. He is currently enrolled for a doctorate course at the University of the Ryukyus. His research interests are power systems engineering, optimization, and intelligent control systems.

Hiroshi Kinjo (Member)

He received the M. Eng. degree and Doctor Eng. degree from Tokushima University in 1984 and 1994, respectively. He is a professor at the University of the Ryukyus. His research interests are intelligent control systems and signal processing.

Eiho Uezato (Member)

He received the M. Eng. degree from the University of the Ryukyus in 1993. He received the Doctor Eng. degree from Kobe University in 1996. He is an associate professor at the University of the Ryukyus. His research interests are descriptor systems.

Tomonobu Senjyu (Member)

He received the M. Eng. degree from the University of the Ryukyus in 1988. He received the Doctor Eng. degree from Nagoya University in 1994. He is a professor at the University of the Ryukyus. His research interests are power systems engineering, optimization, and intelligent control systems.

Tetsuhiko Yamamoto (Member)

He received the Doctor Eng. degree from Doshisha University in 1994. He is a professor at the University of the Ryukyus. His research interests are mechanical control systems.