The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Neurocontroller with Genetic Algorithm for Nonholonomic Systems: Flying Robot and Four-wheel Vehicle Examples

Hiroshi Kinjo, Duong Chau Sam, Eiho Uezato
University of the Ryukyus
Senbaru 1, Nishihara, Okinawa 903–0213, Japan

Tetsuhiko Yamamoto
Tokushima College of Technology
Itano–gun, Tokushima 779–0108, Japan

Abstract

This paper presents a control method using neurocontroller (NC) and genetic algorithm (GA) for a class of nonholonomic systems, known as difficult control problems. First, we introduce the design of NC with GA and then we apply the NC to control two typical examples of nonholonomic systems: flying robot and four-wheel vehicle. Simulations show that the NC controls the systems effectively without the need of the chained form conversion, which is used in time-state control method, and the use of NN and GA provides a straightforward solution for the problem.

Keywords: neurocontroller, genetic algorithm, nonholonomic systems, four-wheel vehicle, flying robot

1 Introduction

Nonholonomic systems have been the subject of an increasing number of researches, especially in control engineering. Among several control methods available for nonholonomic systems, time-state control method using chained form conversion is well known [1]. However, it may have limitations in the controllable ranges [1]. Also, symmetric affine nonholonomic system is uncontrollable with continuous differentiable state feedback control [2]. This paper focuses on control of such systems. Since the use of neurocontroller (NC) with genetic algorithms (GAs) has been applied effectively to the systems that are difficult to be controlled by conventional means, and particularly to nonholonomic systems [3] – [4]. In this study we apply the NC to control two typical nonholonomic systems: a four-wheel vehicle and a hopping robot in flight phase. Using the method, a straightforward solution is provided without using chained form conversion.

This paper is structured as follows: In Section 2 we introduce the design of NC with GA. In Section 3 and 4 the NC is applied to control a flying robot and a four-wheel vehicle, respectively. Finally we conclude this study in Section 5.

2 Neurocontroller with Genetic Algorithm

2.1 Control System

Let $X = [x_1, x_2, \cdots]^T$ be the state of the system, the task of the NC is to control the system from a certain configuration that has initial state variable X^{init} to the desired configuration that has state variable X^{ref}.

Fig. 1 shows the proposed control system, a state feedback controller. From the input u, the state X of the system is determined, this state will be feedback and the deviation $\left(X^{ref} - X\right)$ will be the input of the NC for producing output u. The error between the desired and actual responses is used to update the connecting weights of NC by GA.

Fig. 2 illustrates the structure of NC which uses a three-layer I–J–K architecture NN consisting of input layer, hidden layer, and output layer.

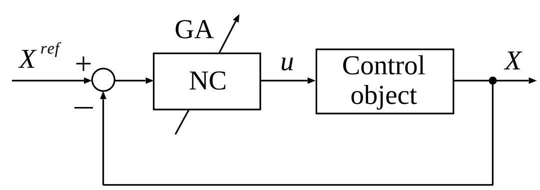


Figure 1: Control system

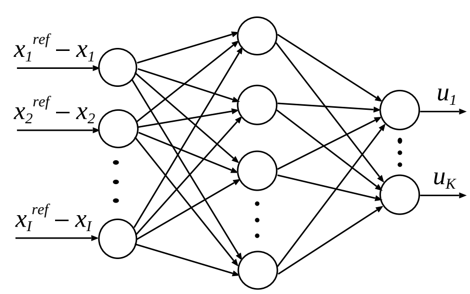


Figure 2: Neurocontroller

2.2 Genetic Algorithm in Controller Design

This paper relates to the combination of GAs and NNs in design of a controller, where GA is employed to train NNs. In the training process, GA searches for the optimal sets of connecting weights of NNs, which are transformed into the genetic codes encoded by 16−bit binary codes.

The algorithm flow of GA is as follows:

Step 1: Initializing a population of NCs with sets of connecting weights drawn randomly from a uniform distribution from the range of $[-0.3, 0.3]$.

Step 2: Control simulations are performed using the NCs. The control performance of each NC is then evaluated.

Step 3: Offspring NCs are produced by the parent NCs which are selected based on the evaluated performances.

Step 4: Control simulations are implemented for the offspring NCs, and their performances are then evaluated.

Step 5: Ranking the NC individuals in the pool of both parent NCs and offspring NCs. The poor-fitness individuals are eliminated from the population.

Step 6: Stop if the termination condition is satisfied. Otherwise go back to Step 3.

In NC training, GA uses of Roulette wheel technique in selection of parents for evolution based on the fitness of NC as:

$$F^{(i)} = \frac{1}{E^{(i)}}, \qquad i = 1, 2, \cdots N \tag{1}$$

where E is error value of i^{th} NC individual, and N is population size. The error function is defined as:

$$E = \sum_{i=1}^{I} Q_i \left(x_i^{ref} - x_i \right)^2 \tag{2}$$

where Q_i is weight coefficient, x_i is coordinate of the state variable X, and x_i^{ref} is coordinate of the desired state X^{ref}.

3 Flying Robot Control by NC

In this section, we consider a hopping robot in flight phase. The mechanism, which consists of a body and an actuated leg which can both extend and rotate, and its mechanical parameters are illustrated in Fig. 3. The configuration of the robot is given by the length (extension) l and the angle ψ of the leg and the angle of the body θ, thus we define the state variable of the system as $X = [x_1, x_2, x_3]^T = [l, \psi, \theta]^T$. Since we control the leg extension and angle directly, we choose their velocities as inputs of the system [1], i.e. $u = [u_1, u_2]^T = \left[\dot{l}, \dot{\psi}\right]^T$. The dynamics of the system is as following:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{b}{a} & -\frac{c}{a} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3}$$

where

$$\begin{cases} a = J + \frac{c_1 + 2c_2 \cos x_2 + c_3}{m_0} \\ b = \frac{m_1 M r \sin x_2}{m_0} \\ c = \frac{c_1 + c_2 \cos x_2}{m_0} \\ c_1 = m_1 (m_2 + M) x_1^2 + 2m_1 (m_2 + 2M) x_1 d \\ \qquad + (m_1 m_2 + 4m_1 M + m_2 M) d^2 \\ c_2 = rM \{ m_1 (x_1 + 2d) + m_2 d \} \\ c_3 = M (m_1 + m_2) r^2 \\ m_0 = m_1 + m_2 + M \end{cases} \tag{4}$$

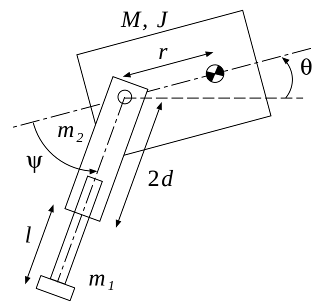


Figure 3: Flying robot

The control problem of this robot we are considering in this study is that from an initial configuration X^{init} in flight phase, the NC is desired to control the robot to the configuration that has state variable $X^{ref} = [0, 0, 0]^T$. In this study, parameters of the robot are as:

$M = 2,\ J = \frac{2}{3},\ m_1 = 0.5,\ m_2 = 0.1,\ d = 1,\ r = 1$

The system is tested with two distinct initial configurations as $X^{init.1} = [2.0, \pi/4, \pi/4]^T$, and $X^{init.2} = [1.0, \pi/6, \pi/3]^T$ (see Ref. [2]).

In this paper, fourth-order Runge-Kutta method is utilized with step size of 0.005. While linear function $f(x) = x$ is used for input and output layers, the tangent hyperbolic activation function $f(x) = \tanh(x)$ is introduced into hidden layer, as it is known as an effective activation function for NN. The initial connecting weights of NN is drawn randomly from a uniform distribution from the range of $[-0.3, 0.3]$. The parameters of GA are selected to be small values in respect to computation cost as depicted in Table 1.

Table 1: Parameters of genetic algorithm

Parameter	Value/scheme
Population N	50
No. of Offspring	$0.6 \times N$
No. of generation	100
Bit number	16
Solution range	$[-20, 20]$
Mutation rate	0.2
Selection scheme	*Roulette wheel*
Q_i	1.0

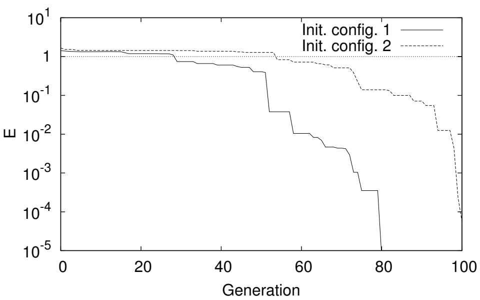


Figure 4: Learning curves

Using the defined parameters and a 3–5–2 structured NN, simulations are implemented. The simulation results are shown in Figs. 4, 5, and 6. We can see that the NC evolves over generations and the trained NC can control the robot effectively.

4 Four-wheel Vehicle Control by NC

Figure 7 shows a four-wheel vehicle system where $L = 2.5$m is distance between two axles of the vehicle.

Let $X = [x_1, x_2, x_3, x_4]^T = [x, y, \theta, \phi]^T$ be state variable of the system. The input of this system is $u = [u_1, u_2] = \left[v, \dot{\phi}\right]$, where v and $\dot{\phi}$ is forward velocity of the rear wheels and velocity of steering wheel of the vehicle, respectively. The dynamics of the system is as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \cos x_3 & 0 \\ \sin x_3 & 0 \\ \frac{1}{L}\tan x_4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5}$$

The task of the control system is to regulate the vehicle from a certain configuration X^{init} to the desired state at origin that has state $X^{ref} = [0, 0, 0, 0]^T$. We implement the tests with four different initial configurations X^{init} as described in Table 2. In this case, a 4–5–2 structured NN is utilized.

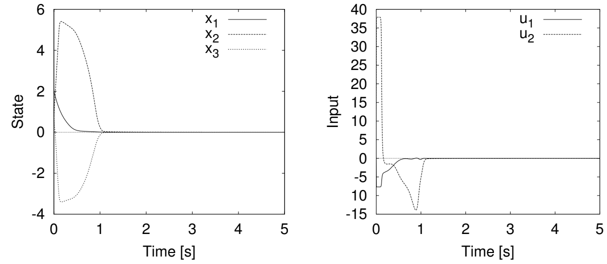


Figure 5: Control result ($X^{init.\,1} = [2,\ \pi/4,\ \pi/4]^T$)

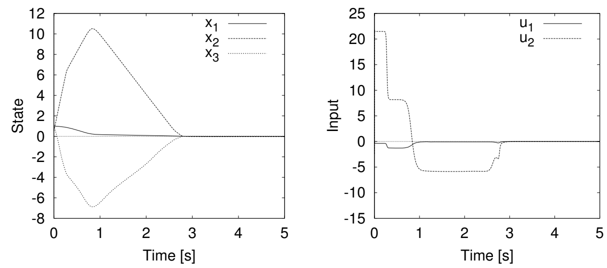


Figure 6: Control result ($X^{init.\,2} = [1,\ \pi/6,\ \pi/3]^T$)

Simulations are performed with the defined parameters and similar ones as in the last section. The simulation results are shown in Figs. 8, 9, and 10. It appears that the NC could control the vehicle effectively, and due to the difficulties arising from the initial configurations, it is harder for training the NC when using the two later sets (see Fig. 8).

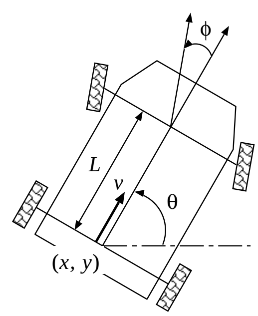


Figure 7: Four-wheel vehicle

5 Summary

This paper has focused on the nonholonomic systems control using NC and GA. Two typical examples of such systems are investigated. By simulations we have shown that the NC could control the systems effectively without the need of chained form conversion.

Table 2: Initial configurations for testing

Set No.	x [m]	y [m]	θ [rad]	ϕ [rad]
Set 1	8.0	3.0	0	0
Set 2	-7.5	3.5	$-\pi/2$	0
Set 3	-7.0	-4.0	$-5\pi/6$	$\pi/12$
Set 4	8.5	-3.5	$3\pi/4$	$\pi/6$

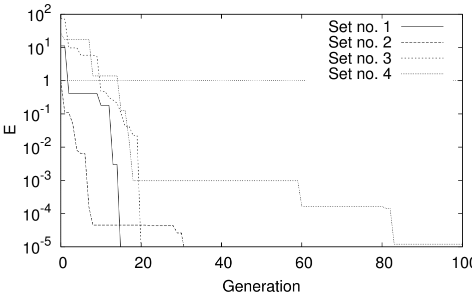


Figure 8: Learning curves

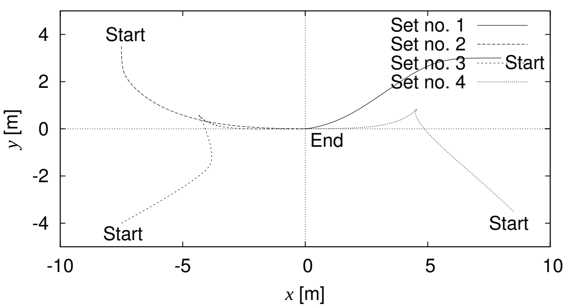
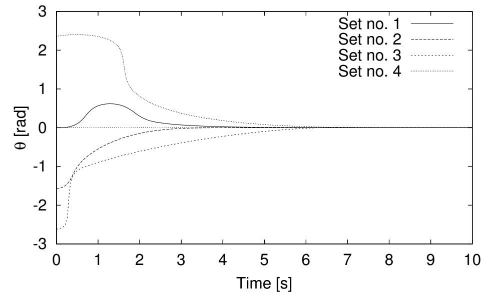
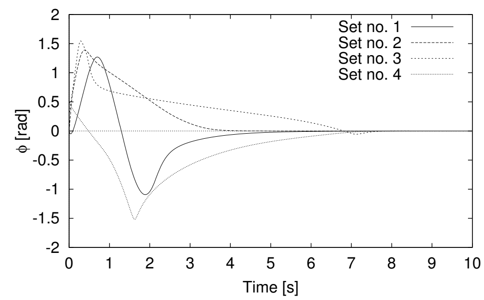


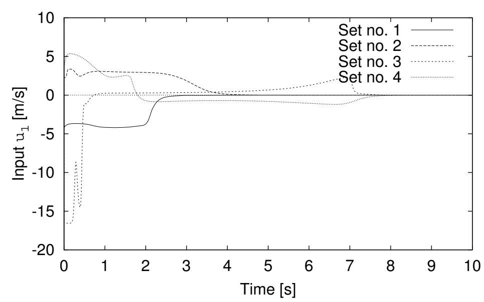
Figure 9: Trajectories of the vehicle



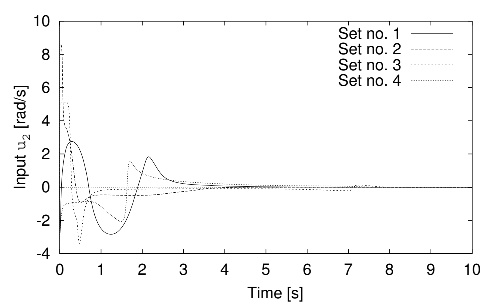
(a) Angles of vehicle body



(b) Steering angles



(c) Control inputs u_1



(d) Control inputs u_2

Figure 10: Control results

References

[1] Richard M. Murray and S. Shankar Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids", *IEEE Trans. on Automatic Control*, Vol. 38, No. 5, pp. 700-716, May 1993.

[2] K. Shimizu and K. Tamura , "Control of Symmetric Affine Systems via Direct Gradient Descent Control", *Trans. of the Society of Instrument and Control Engineers, SICE*, Vol. 43, No. 4, pp. 285-292, 2007. (*in Japanese*)

[3] H. Kinjo, E. Uezato, K. Nakazono, and T. Yamamoto, "Regulator Design of Two-wheel Vehicle Using Neurocontroller Optimized by Genetic Algorithm", *Trans. of the Society of Instrument and Control Engineers, SICE*, Vol. 42, No. 9, pp. 1051-1057, 2006. (*in Japanese*)

[4] Duong C. Sam, H. Kinjo, E. Uezato, and T. Yamamoto, "Applications of Sinusoidal Neural Network and Momentum Genetic Algorithm to Two-wheel Vehicle Regulating Problem", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol. 3, 2008 (*in press*)