International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 4, Issue 04, November 2015)

Truck Trailer Backer Upper Parking with Genetic Algorithm

Lalit Shrivastava
lalitshrivastava@gmail.com

Shubham Singhal
shubham22691@hotmail.com

Abstract — **This paper presents an approach to solve trucktrailer backer upper problem using Genetic algorithm. The results are compared with fuzzy-logic approach.**

Keywords — **Truck, Trailer, Genetic Algorithm, Backer Upper, Fuzzy-logic.**

I. Introduction

The Normal driving instincts can cheat us when attempting to back up a trailer truck to a loading dock. The task is so difficult that a lot of practice is needed to master the skill. And even then, when a truck driver backs up toward a loading dock, he or she will go forward and backward numerous times in order to position the truck at the dock successfully. If the driver is not allowed to make forward movements, successful backing becomes improbable. The problem has become an acknowledged benchmark in nonlinear control and as an example of a self-learning system in neural networks was proposed by Nguyen and Widrow in 1990 [1]. Careful experiments of their approach showed that the computational effort is very high [2]. Thousands (about 20000) of back-up cycles are needed before the network learns. Moreover the back propagation algorithm does not converge for some sets of training samples. Numerous other techniques have been used, including simplified neural network solution through problem decomposition [5]. Very interesting contribution is [6], where up to ten trailers can be controlled representing those as Takagi-Sugeno models and applying linear matrix inequalities method. A simplified version of the control problem (consisting of the cab part only) has been extensively investigated in the field of fuzzy control [2, 711]. In [12] we have shown that hierarchical control system significantly improves control performance and reduces the design load compared to all-in-one approaches investigated by other researcher. In present paper we use Genetic Algorithm to the full truck backer-upper problem.

II. Object Description

The control object consists of cab and trailer parts (Fig. 1). The trailer position is determined by three state variables *x* = [0,100], *y* = [0,100], and, Φ*t* = [-90, 270] the angle between trailer's onward direction and the *x* axis. Length and width of the trailer are 4 and 2 meters, respectively. The cab part is characterized by angle Φ*c* = [-90, 270] between its onward direction. The current implementation of truck backer-upper uses the set of equations from [4].

$$\begin{cases} x(t+1) = x(t) - B\cos(\Phi_t(t)) \\ y(t+1) = y(t) - B\sin(\Phi_t(t)) \\ \Phi_t(t+1) = \Phi_t(t) - \arcsin\left(\dfrac{A\sin(\Phi_c(t) - \Phi_t(t))}{l_t}\right), \\ \Phi_c(t+1) = \Phi_c(t) - \arcsin\left(\dfrac{r\sin(\theta)}{l_t + l_c}\right) \end{cases} \quad (1)$$

where

$$A = r\cos(\theta)$$
$$B = A\cos(\Phi_c(t) - \Phi_t(t)), \quad (2)$$

III. Fuzzy Logic Control

Fuzzy logic controllers (FLC) by using the fuzzy decision process that is based on fuzzy rules enable us to compose any complex translating function. In most cases the Mamdani type of rule is used:

if (X1 is A1) and (X2 is A2) . . . and (Xn is An) then (Y1 is B1) and (Y2 is B2) . . . and (Ym is Bm).

Where terms Xi (i = 1, . . . , n) represent the input variables, Yj (j = 1, . . . ,m) the output variables and the respective Ai , Bj the corresponding linguistic values (fuzzy sets). The numbers n and m consecutively represent the number of input and output variables.

The fuzzy controller in the control loop creates a mapping $\Phi^*_t - \Phi_t \rightarrow (\Phi_t - \Phi_c)^*$. Because the input of the controller the error of the trailer angle, it can be regarded as a proportional controller that determines the angle difference of cab and trailer parts that is necessary to obtain the expected angle of the trailer. It requires only very primitive understanding of the mechanics of the driving system to reach the conclusion that in order to rotate the trailer part to the left the angle of the cab must be negative and vice versa. Being a SISO system, this functional block can be easily tuned manually and is implemented using fuzzy logic in order to obtain a nonlinear mapping that is necessary to achieve high control performance (Fig. 1).

We see that problem decomposition enables us to design the control system because the sub-problems can be accessed individually and in greater detail at the same time. Hierarchical control system is very suitable for the implementation of the multi-level control principle and bringing it back together into one functional block.

IV. Genetic Algorithm

Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a

IJDACR
ISSN: 2319-4863

new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain, 2. A fitness function to evaluate the solution domain. A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming.

The fitness function is defined over the genetic representation and measures the *quality* of the represented solution. The fitness function is always problem dependent. Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions (usually randomly) and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators [3].

V. Experimental Setup

MATLAB 2008b is used to evaluate the results, Both Fuzzy Logic Controller and Genetic Algorithm are implemented and one by one applied on Objective function and Results are analysed.

VI. Results

Figure (1) shows the result for position (x=30, y=40, phi = 70) by Fuzzy Logic Controller:

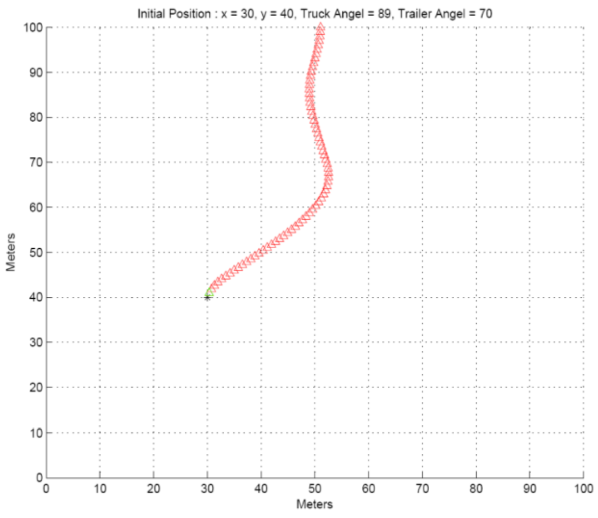

Fig. 1: Truck-Trailer Path For Fuzzy Logic Controller

Figure (2) shows the result for position (x=30, y=40, phi = 70) by Genetic Algorithm:

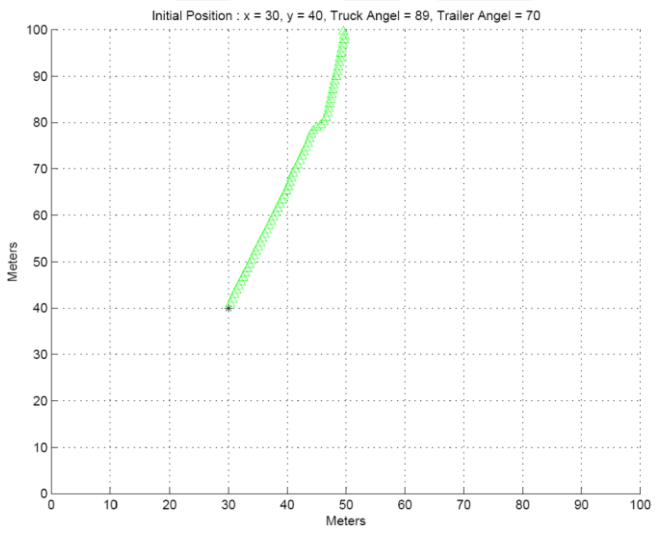

Fig. 4: Truck-Trailer Path Genetic Algorithm

Table below shows the comparison between both schemes:

Scheme	Length of Path	Error
FLC	72 meter	1.0568 meter
GA	67 meter	0.74327 meter

VII. Conclusion

In this paper, a study of backer upper truck-trailer parking is carried out and the results are compared and shown which concludes that Genetic Algorithm is a better choice in this problem.

References

[1]. D. Nguyen and B. Widrow, "The truck backer-upper: an example of self learning in neural networks". *Proc. IJCNN*, vol. 2, pp. 357-363, 1989.

[2]. S. Kong and B. Kosko, "Comparison of fuzzy and neural truck backer-upper control systems". *Proc. IJCNN*, vol. 3, pp. 349-358, 1990.

[3]. J.R. Koza, "A genetic approach to the truck backer upper problem and the inter-twined spirals problem". *Proc. Int. Joint Conf. Neural Networks*, Piscataway, NJ, vol. 4, pp. 310-318.

[4]. M. Schoenauer and E. Ronald. Neuro-genetic truck backer-upper controller. *Proc. First Int. Conf. Evolutionary Comp*., pages 720723. Orlando, FL, USA, 1994.

[5]. R.E. Jenkins and B.P. Yuhas, A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck BackerUpper, *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 718-720, 1993.

[6]. K. Tanaka, T. Kosaki and H.O. Wang, "Backing Control Problem of a Mobile Robot with Multiple Trailers: Fuzzy Modeling and LMIBased Design". *IEEE Trans. Syst., Man, Cybern. Part C*, vol. 28, no. 3, pp. 329-337, 1998.

[7]. P.A. Ramamoorthy and S. Huang, "Fuzzy expert systems vs. neural networks – truck backer-upper control revisited". *Proc. IEEE Int. Conf. Systems Engineering*, pp. 221-224, 1991.

[8]. L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Systems Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, 1992.

[9]. Ismail and E.A.G. Abu-Khousa, "A Comparative Study of Fuzzy Logic and Neural Network Control of the Truck Backer-Upper System". *Proc. IEEE Int. Symp. Intelligent Control*, pp. 520-523, 1996.