Application of Differential Evolution in Truck Trailer Backer Upper Control problem

Lalit shrivastava[1] Prof. Rakesh Sharma[2]

[1] Research Scholar, Department of EI&CE of *Institute of engineering and technology* Alwar, Rajasthan (India)

[2] Department of ECE of *Institute of engineering and technology* Alwar, Rajasthan (India)

***Abstract*— The differential evolution algorithm is used to optimize the control problem of truck trailer backer upper parking. The results are simulated with different parameters, like initial and final position of the TTBU, angles of cabin and trailer. This paper presents an approach to solve truck-trailer backer upper problem which is a bench mark nonlinear problem of the control system. The results are compared with fuzzy-logic approach.**

Keywords— control ,truck Backer Upper, Fuzzy-logic Differential Evolution Algorithm,

1 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 non-linear control and as an example of a self-learning system in neural networks was proposed by Nguyen and Widrow in 1990 [1]. However the approach was to use thousands of backups to train the controller that was not feasible at all. 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 genetic programming [3] neuro-genetic controller [4] and simplified neural network solution through problem decomposition [5]. Very interesting contribution was given by [6], where up to ten trailers can be controlled representing those as Takagi Sugeno models and applying linear matrix inequalities method.

The problem was cited by the different researcher, we have proposed and alternate possible solution for backing of the truck trailer problem, this paper is suggesting and comparing the results of two method fuzzy logic and differential evolution to show the novelty of the research.

2 OBJECT DESCRIPTIONS

2.1 TRUCK TRAILER BAKING UP SYSTEM

The parking of truck and trailer is difficult as it is a benchmark nonlinear problem of the control systems stated by Nguyen and Widrow[1] 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, Φt = [-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 Φc = [-90, 270] between its onward direction. The current implementation of truck backer-upper uses the set of equations from [4].

2.2 EQUATION OF THE TRUCK MOTION

$$\begin{cases} x(t+1) = x(t) - B\cos(\phi_t(t)) \\ y(t+1) = y(t) - B\sin(\phi_t(t)) \\ \phi_t(t+1) = \phi_t(t) - \arcsin\left(\dfrac{A\sin(\phi_C(t) - \phi_t(t)}{l_t}\right) \rightarrow (1) \\ \phi_C(t+1) = \phi_C(t+1) - \arcsin\left(\dfrac{r\sin\theta}{l_t + l_C}\right) \end{cases}$$

where

$A = r\cos\theta$

$B = A\cos(\phi_C(t) - \phi_t(t)) \rightarrow (2)$

2.3 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 (X_1 is A_1) and (X_2 is A_2) . . . and (X_n is A_n) then (Y_1 is B_1) and (Y_2 is B_2) . . . and (Y_m is B_m).

where terms X_i (i = 1, . . . , n) represent the input variables, Y_j (j = 1, . . . ,m) the output variables and the respective A_i , B_j 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 $\Phi^*_c - \Phi_t \rightarrow (\Phi_c - \Phi_t)^*$. 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. 2).

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.

2.4 DIFFERENTIAL EVOLUTION ALGORITHM

The differential evolution (DE) is a method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Such methods are commonly known as metaheuristics as they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as DE do not guarantee an optimal solution is ever found.

Algorithm Differential evolution

1 generate an initial population P = (x1, x2, . . . , xN), xi ∈ D

2 repeat

3 for i := 1 to N do

4 generate a new trial vector y

5 if f(y) < f(xi) then insert y into new generation Q

6 else insert xi into new generation Q

7 endif

8 endfor

9 P := Q

10 until stopping condition

DE has been used in several science and engineering applications to discover effective solutions to nearly intractable problems without appealing to expert knowledge or complex design algorithms.

DE is used for multidimensional real-valued functions but does not use the gradient of the problem being optimized, which means DE does not require for the optimization problem to be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. DE can therefore also be used on optimization problems that are not even continuous, are noisy, change over time, etc.

DE optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. In this way the optimization problem is treated as a black box that merely provides a measure of quality given a candidate solution and the gradient is therefore not needed.

The major difference between Genetic Algorithms and Differential Evolution is that Genetic Algorithms rely on crossover, a mechanism of probabilistic and useful exchange of information among solutions to locate better solutions, while evolutionary strategies use mutation as the primary search mechanism.

DE is a population based search technique which utilizes NP variables as population of D dimensional parameter vectors for each generation. The initial population is chose randomly if no information is available about the problem. In the case of the available preliminary solution, the initial population is often generated by adding normally distributed random deviations to the preliminary solution. The basic idea behind DE is a new scheme for generating trial parameter vectors. DE generates new parameter vectors by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. In addition, the best parameter vector is evaluated for every generation in order to keep track of the progress that is made during the optimization process. Extracting the distance and the direction information from the population to generate random deviations result in an adaptive scheme with excellent convergence properties (Price et al., 2005).

A basic variant of the DE algorithm works by having a population of agent's solutions (candidate). These candidates are moved around in the search-space by using simple mathematical formulae to combine the positions of existing agents from the population. If the new position of an agent is an improvement it is accepted and forms part of the population, otherwise the new position is simply discarded. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Different strategies can be adopted in the DE algorithm depending upon the type of problem to which DE is applied. The strategies can vary based on the vector to be perturbed, number of difference vectors considered for perturbation, and finally the type of crossover used. The following are out of the ten different working strategies:

1. DE/best/1/exp
2. DE/rand/1/exp
3. DE/rand-to-best/1/exp
4. DE/best/2/exp
5. DE/rand/2/exp

The general convention used above is DE/x/y/z. DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used(exp: exponential; bin: binomial). Hence the perturbation can be either in the best vector of the previous generation or in any randomly chosen vector.

Formally, let $f : R^n \rightarrow R$ be the cost function which must be minimized or fitness function which must be maximized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the fitness of the given candidate solution. The gradient of f is not known. The goal is to find a solution m for which $f(m) \leq f(p)$ for all m in the search-space, which would mean m is the global minimum. Maximization can be performed by considering the function $h := -f$ instead.

Let $X \in R^n$ designate a candidate solution (agent) in the population. The basic DE algorithm can then be described as follows:

- Initialize all agents X with random positions in the search-space. Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat the following:
- For each agent X in the population do:
- Pick three agents **a, b** and **c** from the population at random, they must be distinct from each other as well as from agent X.
- Pick random index $R \in \{1,2,.....n\}$, n being the dimensionality of the problem to be optimized).
- Compute the agent's potentially new position=[y_1,y_2,……y_n]as follows:
- For each i, pick a uniformly distributed number $r_i \equiv U(0,1)$
- If $r_i < CR$ or i=R then set $y_i = a_i + F(b_i - c_i)$ otherwise set $y_i = x_i$.
- If $f(y) < f(x)$ then replace the agent in the population with the improved candidate solution, that is, replace X with Y in the population.
- Pick the agent from the population that has the highest fitness or lowest cost and return it as the best found candidate solution.

$f \in [0,2]$ is called the *differential weight* and $CR \in [0,1]$ is called the *crossover probability*, both these parameters are selectable by the practitioner along with the population size $NP \geq 4$.

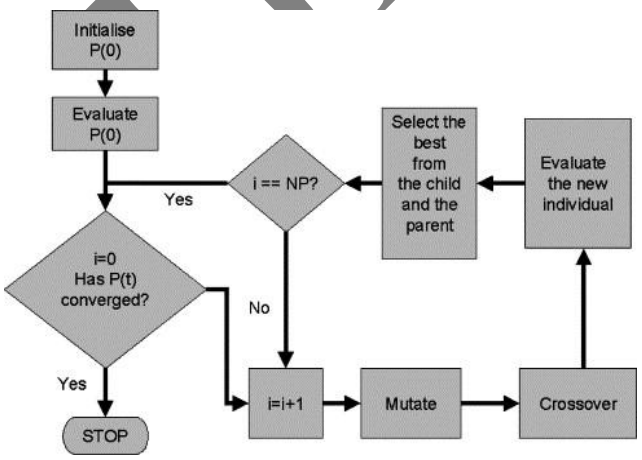

Figure – 1 Differential Evolution Algorithm

3 EXPERIMENTAL SETUP

To validate over project, MATLAB 2008b is used to evaluate the results. We also added different sets of parameter for each we have calculated the path length travelled by the truck and trailer and the error occurred during the each approach. Both Fuzzy Logic Controller and Differential Evolution Algorithm are implemented on a set of parameters to analyse and compare the path and error to reach dock. We have applied different initial and final condition for the truck and trailer for different angles one by one on Objective function and Results are compared and analysed. We have used following indicator for the experiment

PARKING POSITION [X = 50 | Y = 100 | PHI = 90]

Length of the trailer =14 m

Length of cab =6 m

Speed of the truck= constant

The steering angle = constant

Angle between trailer's onward direction and the x axis Φt = [-90,270]

Angle between The cab & its onward direction. Φc = [-90, 270] .

3.1 RESULTS

Figure 3 shows the result for position (x=45, y=45, phi = 70) by Fuzzy Logic Controller:

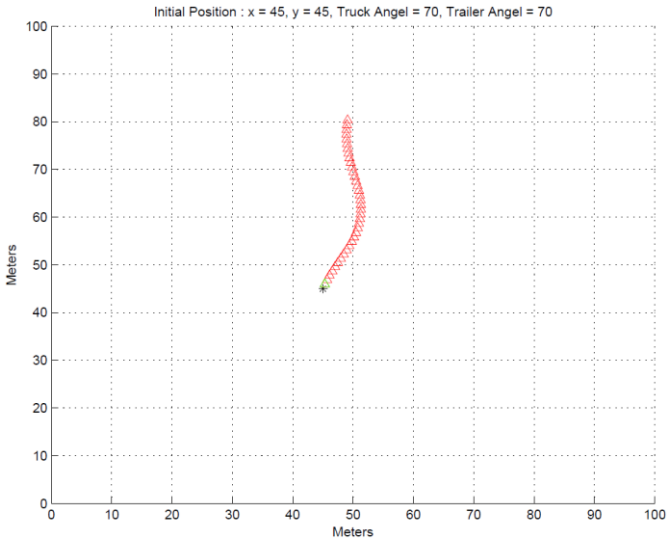

Figure- 2: Truck-Trailer Path For Fuzzy Logic Controller

Figure 1 shows the result for position (x=45, y=45, phi = 70) by Differential Evolution Algorithm MaxGen=1000:

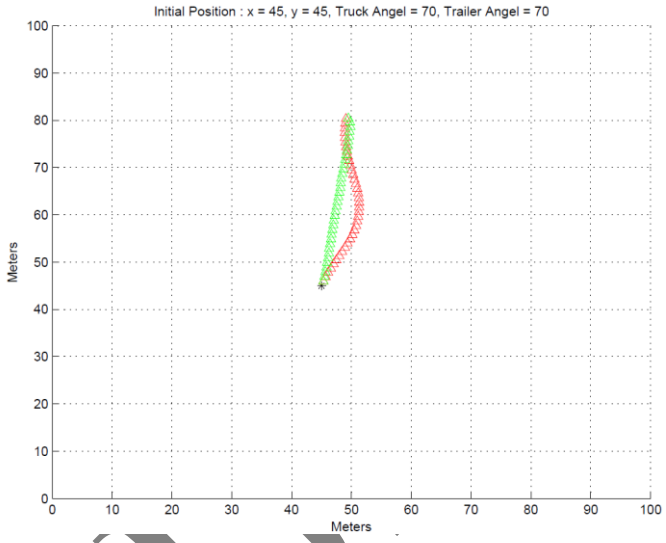

Figure 3: comparison of FLC and DE

Table2: below shows the comparison between both scheme :

Scheme	Length of Path	Error
FLC	38	0.9065
DE	37	0.5032

4. Conclusion

The overall object of this paper is to develop an algorithm which can be used to optimize the fuzzy controller for backing up of the truck\ with along trailer. A study of backer upper truck-trailer parking is carried out and the results are compared. Different parameter sets are applied to the objective function to show that Differential evolution algorithm will provide the better time and the less dock error in comparison to the fuzzy logic controller. Finally some simulations are presented to show a desired smooth movement of the truck. It may be suggested then that if the system inputs are measured with small errors and applied to the fuzzy controller, the proposed method can be applied in practice.

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 720-723. 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 Backer-Upper, *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 LMI-Based 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] A. 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.

[10] D. Kim, "Improving the fuzzy system performance by fuzzy system ensemble". *Fuzzy Sets and Systems*, vol. 98, pp. 43-56, 1998.

[11] I. Dumitrache and C. Buiu, "Genetic learning of fuzzy controllers", *Mathematics and Computers in Simulation*, vol. 49, pp. 13-26, 1999.

[12] A. Riid and E, Rüstern. "Fuzzy logic in control: Truck backer-upper problem revisited". *Proc. IEEE 10th Int. Conf. Fuzzy Systems*, 1. Melbourne, Australia, 2001.