Fitting Membership Function with PSO Inertia Weight for Truck Backer-Upper Problem

Siti Zaiton Mohd Hashim
Faculty of Computer Science and Information System
University Teknologi Malaysia
Johor Bahru, Malaysia
sitizaiton@utm.my

Kurniawan Eka Permana [1,2]
[1] Faculty of Computer Science and Information System
University Teknologi Malaysia
[2] Faculty of Engineering
Trunojoyo University
Kamal, Indonesia
kurniawan@inf.trunojoyo.ac.id

Abstract—**In this paper, optimization is utilized for fuzzy control system that automatically backs up a truck to a specified point on a loading dock. Particle Swarm Optimization with Inertia Weight (FPSOw) is used to fit the membership function of fuzzy system in searching for better decision. From the simulation conducted, it shows that the FPSOw is able to generate an optimal set of parameter for the fuzzy membership functions automatic adjustment and then the adjusted fuzzy system achieve better position for Truck Backer-Upper result.**

Keywords- Particle Swarm Optimization; Fuzzy Membership Function; Fuzzy control; Truck Backer-Upper

I. INTRODUCTION

The task of optimization is that of determining the values of set parameters so that some measure of optimality is satisfied. Many scientific, social, economic and engineering problems have parameters that can be adjusted to achieve more optimize outcome, including solving control problem. The truck backer-upper problem has become a standard control problem to be investigated

The objective of this problem is to back up the truck so that it arrive perpendicular to the dock at specified position from any start position. These systems can be considered as knowledge-based systems, incorporating human knowledge into their knowledge base through fuzzy inference system and fuzzy membership functions.

This paper discourses an optimization method for fitting membership functions. The method adjusts membership function automatically based on Particle Swarm Optimization with Inertia Weight.

The PSO optimization technique was introduced by Kennedy and Eberhart in [2] as a stochastic search through an *n*-dimensional problem space aiming the minimization (or maximization) of the objective function of the problem. The PSO with Inertia Weight is a scaling factor associated with the velocity during the previous time step. Its aiming at further improving the rate convergence of the algorithm.

This paper is organized as follows: Section II presents an overview of the Particle Swarm Optimization (PSO) with Inertia Weight. Section III presents the problem to be solved. Section IV shows the PSO training mode. Section V shows the tests performed with a fuzzy control, which membership functions were adjusted using PSO algorithms. Finally, Section VI presents the conclusion.

II. PSO WITH INERTIA WEIGHT

The Particle Swarm Optimization Algorithm (PSO) is a population-based optimization method finds the optimal solution using a population of particles [1-4]. Every swarm of PSO is a solution in the solution space. PSO is basically developed through simulation of bird flocking in two-dimension space. The PSO definition is presented as follows:

Each individual particle *i* has the following properties: A current position in search space, *xi*, a current velocity, *pid*, and a personal best position in search space, *pid*.

• The personal best position, *pid*, corresponds to the position in search space where particle *i* presents the smallest error as determined by the objective function *f*, assuming a minimization task.

• The global best position denoted by represents the position yielding the lowest error amongst all the p_{gd}.

During the iteration every particle in the swarm is updated using equations (1) and (2).

$$v_{id}= v_{id}+c1*rand()*(p_{id}-x_{id})+c2*rand()*(p_{gd}-x_{id}) \quad (1)$$

The current position of the particle is updated to obtain its next position:

$$x_{(t+1)} = x_{(t)} + v_{(t+1)} \quad (2)$$

where *c1* and *c2* are two positive constants random numbers within the range [0,l].

The inertia weight is a scaling factor associated with the velocity during the previous time step, resulting in a new velocity update equation, so that;

$$v_{id}=w*v_{id}+c1*rand()*(p_{id}-x_{id})+c2*rand()*(p_{gd}-x_{id}) \quad (3)$$

where *w* is the inertia weight.

978-0-7695-3648-4/09 $25.00 © 2009 IEEE
DOI 10.1109/AMS.2009.131

IEEE computer society

The original PSO velocity update equation can be obtained by setting *w*.=1. Shi and Eberhart investigated the effect of *w* values. Their result indicate that choosing *w* between 0.8 to 1.2 result in faster convergence.[3]

III. TRUCK BACKER-UPPER PROBLEM

The objective of the truck backer-upper problem is to back up the truck so that it arrive perpendicular to the dock at specified position from any start position[6].

First is develop a set of fuzzy control rules and membership functions, which will define the truck path. The variable fuzzing and defuzzing process is running by the program.

Figure 1 shows a simple model. The objective of the control system is to back up the truck so that it arrive perpendicular to the dock at position (x_f, y_f). The coordinate is (50,100). The point (*x, y*) is at the center of the rear of the truck, *f* is the angle of the truck axis to the horizontal, and *q* is the steering angle measured from the truck axis. The controller takes as input the position of the truck, specified by the pair (*x, f*), and outputs the steering angle *q*.

The coordinate *x* ranges from 0 to 100, *f* ranges from -100 to 280 and *q* ranges from -30 to 30 as shown in Table I, II, and III.

Next is defined the truck angle, *f*, and the truck *x*-position coordinate, *x*. At every stage the fuzzy should produce the steering angle, *q*, which backs up the truck to the dock.

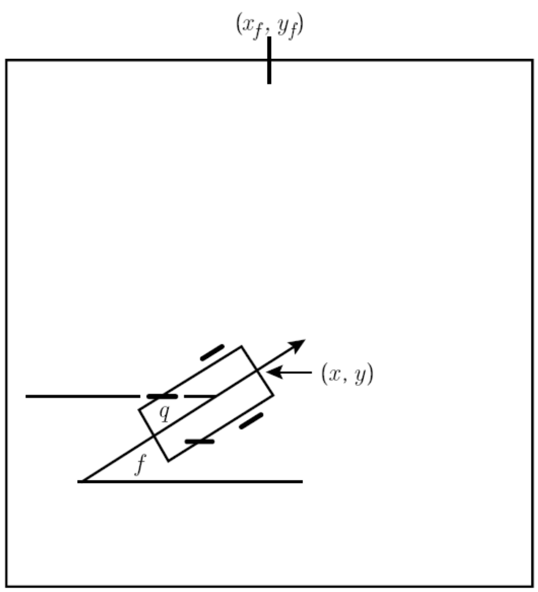

Figure 1. The model for the simple truck backer upper. The point (*x,y*) is the center of the rear of the truck, *f* is the angle of the truck axis to the horizontal and *q* is the steering angle measured from the truck axis

The linguistic variables associated with the fuzzy sets for the *x* position are: LE (left), LC (left center), CE (center), RC (right center), and RI (right). The following membership functions define those sets:

TABLE I. FUZZY SETS FOR X POSITION

MF	Position		
	left	***center***	***right***
LE	0	10	35
LC	30	40	50
CE	45	50	55
RC	50	60	70
RI	65	90	100

Fuzzy sets for the angle *f* are: RB (right below), RU (right upper), RV (right vertical), VE (vertical), LV (left vertical), LU (left upper), and LB (left below).

TABLE II. FUZZY SETS FOR ANGLE

MF	Position		
	left	***center***	***right***
RB	-100	-45	10
RU	-10	35	60
RV	45	67,5	90
VE	80	90	100
LV	90	112,5	135
LU	120	155	190
LB	170	225	280

And then for the output variable, the sets are: NB (negative big), NM (negative medium), NS (negative small),ZE (zero), PS (positive small), PM (positive medium), and PB (positive big).

TABLE III. FUZZY SETS FOR STEERING ANGLE

MF	Position		
	left	***center***	***right***
NB	-30	-15	-15
NM	-25	-15	-5
NS	-12	-6	0
ZE	-5	0	5
PS	0	6	12
PM	5	15	25
PB	18	30	30

Finally, the fuzzy rules for this problem appear in Table IV. It is used to determine the output given by the input conditions with min-max fuzzy combinational method.

The fuzzy rules are composed of 35 rules determined by the input *x* position and the angle *f*.

TABLE IV. FUZZY RULES

	LE	**LC**	**CE**	**RC**	**RI**
RB	PS	PM	PM	PB	PB

	LE	LC	CE	RC	RI
RU	NS	PS	PM	PB	PB
RV	NM	NS	PS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	PS	PM
LU	NB	NB	NM	NS	PS
LB	NB	NB	NM	NM	NS

IV. PSO TRAINING MODE

The integration between PSO with inertia weight algorithms with fuzzy control is as follow:

1. The parameters are the lefts and rights of each foot fuzzy membership function.
2. These parameters act as particles and looking for the global best fitness.
3. It starts with initial set parameters
4. After the parameters had been adjusted using PSO with inertia weight, it will be used to check the performance of the fuzzy system.
5. The process is repeated until goal is achieved or PSO with inertia weight reached the global best.

PSO with inertia weight starts with the initial set parameter and gets the fitness function to define new values, representing the membership function parameter set of values. This new values will be used by the problem to search the path.

A. Calculation of fitness function

For each fuzzy controller membership function, three parameters are defined. They are: *C* (Center), *L*(left) and *R*(right).

For the adjustment of membership functions the following equations were defined:

$$C=(C+k_i)-w_i \tag{4}$$

$$L=(L+k_i)-w_i \tag{5}$$

$$R=(R+k_i)-w_i \tag{6}$$

Being k_i and w_i adjustment coefficient, k_i makes each membership function move to the right or left with no change in the form. The membership function shrinks or expands through the factor w_i

PSO with inertia weight will be used to find the optimum values according to the strategy and the initial points used, as well as for k_i and w_i for the membership function.

B. Algorithm Presentation

The fuzzy system process through PSO with inertia weight is summarized into the following procedure:

1. Generate initial swarm parameter
2. If maximum iteration achieved, go to step 7
3. Build fuzzy membership function for each particle
4. Evaluate fitness for each particle
5. Update the particle position
6. Go to step 2.
7. End.

According this procedure, each particle will build fuzzy model by changing the coefficient k_i and w_i . With regard to particle size, like how many elements each particle will have, this will depend on the number of membership functions.

V. SIMULATIONS AND RESULTS

This section presents simulations and results for Truck Backer-Upper Problem. The problem used with original fuzzy membership function define by expert. Then compare it with fuzzy membership function that adjust with original PSO(FPSO) and with PSO with inertia weight (FPSOw).

The FPSOw parameters are shown in Table V. For initial k_i and w_i are use -1 and 5, respectively. Iteration is 500.

TABLE V. PSO PARAMETER

Parameter	Value
c1	2
c2	2
dt	0.1
Inertia weight	0.8
Number of particles	5

The result is shown in Figure 2, the simulation succeeds at smoothly backing up the truck. Fuzzy control needs more time and iteration to converge, FPSO reach faster in term of iteration to converge and the other hand, FPSOw achieve the precision position than the others method.

Two different position and angle has been tested and showed. First case, the truck center is in the position *x*=50, *y*=20 and the angle *f*=30. The movement goes up and then reaches *x*-position 50 as a goal. Secondly, initial position is set to *x*=90, *y*=10 and the angle *f*=110. The movement also reaches *x*-position 50 as a goal.

For the case 1, FPSO takes converge on iteration 31. Fuzzy takes 56 to achieve convergence and FPSOw need 42 iteration to converge. In term of precision, FPSOw achieve more precisely at position 50.30118. Second case achieve the same, FPSOw achieve better convergence and closer to destination than fuzzy and get more precisely than FPSO.

Table VI shows the result obtained from simulations made with total 97 iteration.

TABLE VI. DATA RESULT FOR *X*-POSITION

no	Case I			Case II		
	Fuzzy	*FPSO*	*FSPSOw*	*Fuzzy*	*FPSO*	*FPSOw*
1	50	50	50	50	50	50
2	53.69	53.6770	53.56516	48.52442	48.55972	48.54036
3	56.68	56.6285	56.40586	47.29058	47.42479	47.33889
4	59.07	58.9311	58.63896	46.30097	46.61642	46.41669
12	63.92	62.4161	62.4299	45.32062	47.57747	46.66806
13	63.55	61.8315	61.97475	45.64086	48.04615	47.08541
14	63.10	61.1557	61.44166	45.98381	48.49954	47.49331
15	62.58	60.4153	60.85498	46.33912	48.92231	47.87614
16	62.00	59.6328	60.23465	46.69853	49.30337	48.22261
91	52.22	51.5699	50.30193	52.21545	49.98625	50.32081
92	52.22	51.5695	50.30177	52.21621	49.98629	50.32079
93	52.22	51.5692	50.30162	52.2169	49.98632	50.32077
94	52.22	51.5688	50.30149	52.21754	49.98634	50.32075
95	52.22	51.5684	50.30138	52.21812	49.98636	50.32073
96	52.22	51.5681	50.30127	52.21866	49.98637	50.32072
97	52.22	51.5678	50.30118	52.21915	49.98637	50.32071

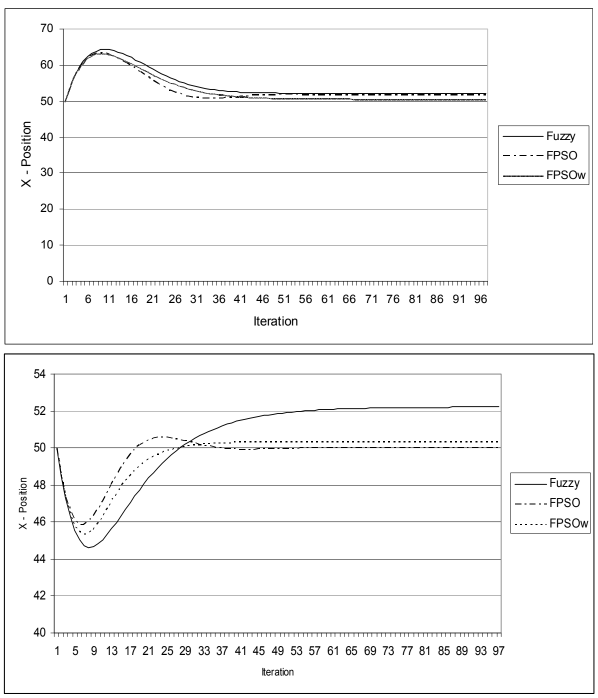

Figure 2. Simulation Result. Compare between the result from Case I and Case II

VI. CONCLUSIONS

The use of Fuzzy Logic to solve control problem have been increasing considerably. The truck backer-upper problem has become a standard control problem. The object of the control system is to back up the truck so that it arrive perpendicular to the dock at specified position from any start position.

The successfulness of fuzzy application depends on a number of parameters, such as fuzzy membership function, that are usually decided by the expert. One way to improve the performance of the fuzzy system is to hybrid with optimization algorithm.

This paper presents an automatic strategy to adjust membership function by using Particle Swarm Optimization with inertia weight.

Compared with fuzzy alone, this FPSO providing better result and faster to converged. But the best result to reach the destination is FPSOw for some positions. Almost all position gave the result that FPSOw is outstanding to achieve the destination.

The optimal result is presented as a promising result for optimization process. The authors intend to further investigate this problem in quest for development of an improved PSO, which can train the problem in faster time to converge.

ACKNOWLEDGMENT

The authors thank the Malaysian Government, Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia for the Fundamental Research Grant Scheme (FRGS), Vot. No. 78257.

REFERENCES

[1] Eberhart, R. and Kennedy, I. “A new optimizer using particle swarm theory”.[J] Proc. 6 Int. Symposium on Micro Machine and Human Science, 1995.39-43.

[2] J. Kennedy and R.C. Eberhart, "Particle swarm optimization" , Proceeding of the 1995 IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Centre, Piscataway, NI, 1995, Iv: 1942-1948

[3] Y. Shi and R.C. Eberhart, "A modified Particle Swarm Optimiser", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[4] Esmin, A. A. A, “Generating Fuzzy Rules from Examples Using the Particle Swarm Optimization Algorithm”, Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on 17-19 Sept. 2007 Page(s):340 – 343.

[5] S Michael Negnevitsky, Artificial Intelligence, Addison Wesley, 2004

[6] S-G Kong and B. Koskos, “Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer, IEEE Trans on Neural Networks, vol. 3, no.2, March, 1992