Design of a Genetic-Fuzzy System for Planning Optimal Path and Gait Simultaneously of a Six-legged Robot

Dilip Kumar Pratihar, Kalyanmoy Deb, and Amitabha Ghosh
Kanpur Genetic Algorithms Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur
Kanpur, Pin 208 016, India
E-mail:{dkpra,deb,amitabha}@iitk.ac.in

Abstract

This paper describes a *genetic-fuzzy* system used for generating optimal path and gait simultaneously of a six-legged robot. No single traditional approach is found to be successful in handling this complicated task. Moreover, the conventional methods are computationally expensive and the generated path and gaits may not be optimal in any sense. Thus, there is still a need for the development of an efficient and computationally faster algorithm for solving this problem. In the proposed algorithm, optimal path and gaits are generated by fuzzy logic controllers (FLCs) and optimized FLCs are found by genetic algorithms (GAs). Design of an optimized FLC (only rule base optimization) involves the problem of dealing with discrete variables and GA is an efficient tool for this purpose. The actual optimization is done off-line and the hexapod can use these GA-tuned FLCs to navigate in real-world scenarios, in an optimal sense.

1 INTRODUCTION

Genetic algorithms (GAs) are population based search and optimization techniques which work based on the mechanics of natural genetics (Goldberg, 1989), whereas fuzzy logic controller (FLC) is a potential tool for handling imprecision and uncertainty (Kosko, 1994). The FLC is found to be very attractive for solving this type of problems because in actual navigation, the data regarding the presence of obstacles, ditch, and others are collected using the sensors and the sensor readings are not always precise. To get the advantages of both the techniques - in one approach, an FLC is used to improve the performance of a GA (Herrera et al., 1994), whereas in other implementation, a GA is used to design an optimized FLC (Karr, 1991). Our present work is based on the second approach. Three steps are to be followed for legged-robot locomotion: (i) determination of vehicle's trajectory, (ii) foothold selection, and (iii) design of a sequence of leg movements. In practice, path and gait generations of a legged vehicle are to be done simultaneously. An attempt was made by Cho et al. (1995) to solve the problem of path and gait generations simultaneously but the trajectory planning scheme is neglected there. Moreover, a hybrid technique based on GA-Fuzzy combinations has been developed by Magdalena and Velasco (1996) for solving bipedal locomotion problem but in their work, only the foothold selection part has been studied in details. Our aim is to design a *genetic-fuzzy* system which can tackle both the path generation and gait generation problems simultaneously.

2 A FEW DEFINITIONS

1. *Gait*: It is defined as a sequence of leg movements for the purpose of transporting the body of the legged robot from one place to another. There are two types of gaits - *periodic gait* and *non-periodic gait*.

2. *Stroke*: It is defined as the distance through which the foot is translated relative to the body during the support phase.

3. *Stability margin*: It is the distance of the vertical projection of center of gravity (CG) to the boundaries of the support pattern in the horizontal plane.

4. *Kinematic margin*: It is defined as the distance from the current foothold of leg i to the boundary of the reachable area of leg i, measured in the opposite direction of body motion.

5. *Duty factor*: It is defined as the time fraction of support phase in one cycle time.

6. *Crab angle*: It is defined as the angle from the longitudinal axis to the direction of motion.

3 PROBLEM FORMULATION

A six-legged robot (Fig. 1) will have to find a collision-free, time-minimal path and to plan its gait in optimal sense (with minimum number of ground-legs having the maximum average kinematic margin) while moving on flat terrain. Moreover, its stability margin should always be positive to ensure static stability. As the number of ground-leg increases, the probability of occurring a deadlock situation increases and as the average kinematic margin of the ground-legs is more, the potential progress of the vehicle towards the goal is also more. Each obstacle is represented by its bounding circle. It is assumed that the mass of all legs is lumped into the body and the center of gravity is located at the centroid of the body. The vehicle's complete path is a collection of a number of small straight-line segments and circular arcs each traveled for a constant time-step, ΔT. The hexapod starts from zero velocity and accelerates for some time so that its velocity reaches the maximum value. The magnitude of acceleration is kept same as that of deceleration and it is a. Moreover, when the robot comes closer to the destination and there is no critical obstacle in its way, the robot starts decelerating so that it can reach its destination with a zero velocity. The total travel time, T is then calculated as follows:

$$T = \frac{D}{v} + \frac{v}{a}, \qquad (1)$$

where D is the total distance traveled, v is the maximum velocity of the vehicle along the trajectory. Besides time-optimal path planning, the hexapod will have to determine its optimal gait. While navigating, the hexapod will have to move along a straight path (periodic gait), to take a circular turn (non-periodic gait), to cross a ditch (non-periodic gait) as the situation demands. For each mode of gait generation, the distance to be traveled by the vehicle during time-step, ΔT is divided into Q equal parts usually known as *motion segments*. The decisions regarding lifting and placing of legs are taken at the end of each motion segment. A leg with negative kinematic margin means the leg is lifted in the air and a leg with the highest positive kinematic margin is placed on the ground for maintaining the stability of the robot.

Two reference frames, namely world coordinate frame $\{W\}$ and body coordinate frame $\{B\}$ have been considered for the purpose of gait analysis (Fig. 2). Here, ${}^W_B E$ represents the transformation vector from $\{W\}$ to $\{B\}$ and ${}^B l_i$ indicates the position of i-th leg with respect to the body coordinate frame $\{B\}$. The position of i-th leg with respect to the world coordinate frame $\{W\}$ is represented by ${}^W l_i$. The position of a foot in the body coordinate frame is related to the position in the world coordinate frame as given by the expression:

$${}^B l_i = {}^W l_i - {}^W_B E. \qquad (2)$$

The problem can be stated mathematically as follows:

$$\text{Maximize} \quad z = \sum_{i=1}^{NPG} (w_1 \times (6 \times Q_i - C_i) + w_2 \times K_i) \qquad (3)$$

subject to the condition that the stability margin is positive, where NPG is the total number of non-periodic gait generation mode, Q_i is the number of motion segments in the i-th mode, C_i is the total number of ground-legs in the i-th mode, K_i indicates the average kinematic margin of the ground-legs in the i-th mode, w_1 and w_2 are the weighting factors.

4 PROPOSED ALGORITHM

In our proposed genetic-fuzzy system, a GA-based tuning is used to improve the performance of an FLC. There are seven FLCs (one for path generation and six for gait generation), running in parallel, to solve the combined problem of path and gait generations. At the beginning of each time-step, the vehicle first searches whether there is any ditch ahead. If it finds a ditch, the ditch crossing module is activated, otherwise, it moves along a straight path following the periodic gait pattern. After moving along a straight path or crossing the ditch, if the vehicle is forced to change its direction of movement it takes a circular turn in the next time-step. The performance of an FLC depends on its rule base and membership function distributions. It has been observed that optimizing rule base of an FLC is a rough tuning process, whereas optimizing the scaling factors of membership function distribution is a fine tuning process. Thus, we optimize the rule base only of an FLC. Fig. 3 shows a genetic-fuzzy system. The combined problem includes many modules, as discussed below.

4.1 PATH GENERATION MODULE

Two inputs, namely distance and angle are fed to the fuzzy logic controller and there is one output - deviation(Fig. 4). Fig. 5 shows the membership function distributions for input and output variables. As there are four and five different values for distance and angle, respectively, we have considered 4×5, that is, 20 rules in the manually constructed rule base of the FLC (Table 1).

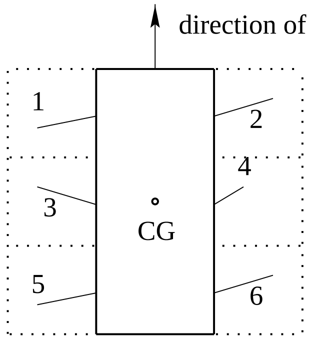

Figure 1: A schematic diagram of a six-legged robot

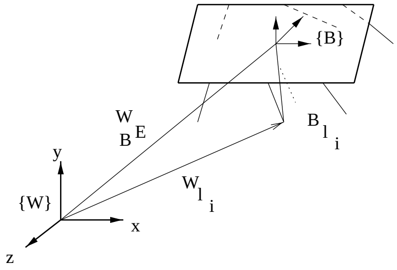

Figure 2: A schematic showing world frame and body frame

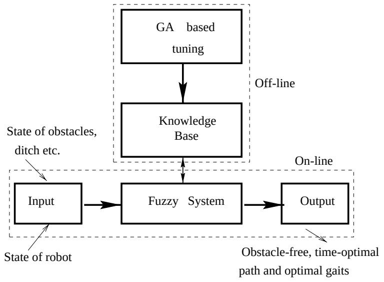

Figure 3: A genetic-fuzzy system

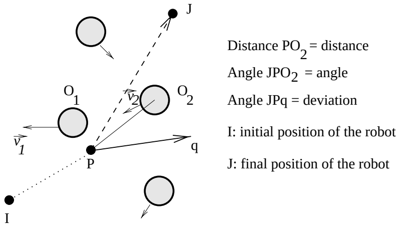

Figure 4: A schematic showing inputs and output of an FLC - path generation module

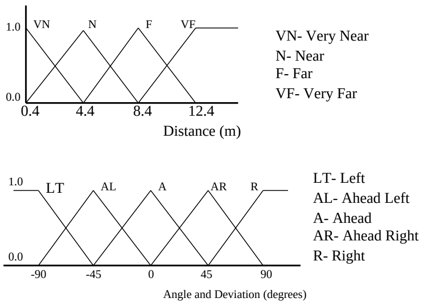

Figure 5: Author-defined membership function distributions for input and output of an FLC - path generation module

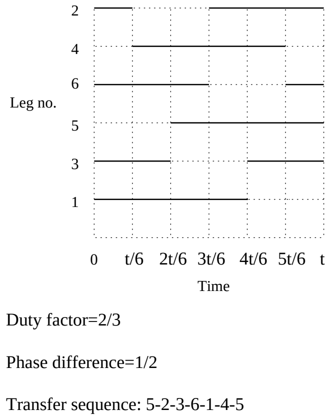

Figure 6: A schematic showing gait-diagram (wave gait)

Thus, a typical rule looks as follows:

If distance is VN and angle is AL, then deviation is AR.

4.2 PERIODIC GAIT GENERATION MODULE

Periodic gaits are generated by a legged robot only when it moves on a smooth terrain, along a straight-line path (Tomovic and Karplus, 1961). For a periodic gait (wave gait) with a duty factor β, the phases of legs 1,3,5,2,4,6 with respect to leg 1 are 0, β, $2\beta - 1$, $1/2$, $\beta - 1/2$, $2\beta - 1/2$, respectively. For fixed values of duty factor, $\beta(= 2/3)$ and phase difference, $\gamma(= 1/2)$, we have determined the sequence of leg transfer (Fig. 6). The wave gait is optimal in terms of ground-legs. All the six FLCs used for controlling the legs have the same and fixed output, that is, stroke.

4.3 DITCH CROSSING MODULE

There are two inputs - distance and relative angle(Fig. 7) and one output (leg stroke) of the fuzzy logic controller. The proposed algorithm is based on the *stroke control strategy*. The membership function distributions for input and output are shown in Fig. 8. For each FLC, 20 rules are considered (Table 1). Thus, we consider 120 rules for all the six FLCs.

4.4 TURNING GAIT GENERATION MODULE

Two inputs (distance and crab angle) are given to the FLC and it produces one output (stroke). Fig. 9 shows the condition variables and the author-defined membership function distributions for the variables are shown in Fig. 10. Table 1 shows the author-defined rule base containing 20 rules for an FLC and it is the same for all the six FLCs. A binary coded GA with 260-bit string is used to represent a solution. The GA string looks as follows:

$$\underbrace{10111\ldots1}_{\text{path}}\underbrace{01100\ldots01100111}_{\text{ditch crossing gait}}\underbrace{11001\ldots10001101}_{\text{turning gait}}$$

Here, 1 and 0 represent the presence and absence of the corresponding fuzzy rule, respectively. The first

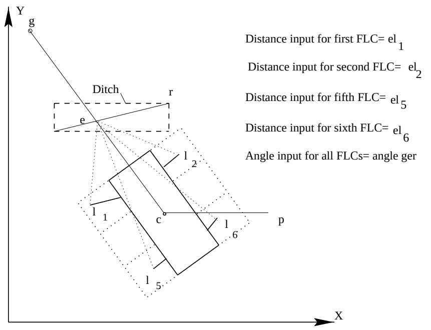
Y
g
Ditch
r
e
c
p
X
l_1
l_2
l_5
l_6
Distance input for first FLC= el_1
Distance input for second FLC= el_2
Distance input for fifth FLC= el_5
Distance input for sixth FLC= el_6
Angle input for all FLCs= angle ger

Figure 7: A schematic showing inputs of an FLC - ditch crossing module

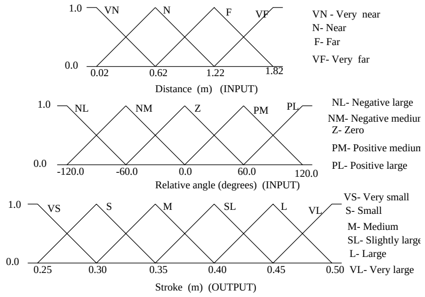

Figure 8: Author-defined membership function distributions for input and output of the FLCs - ditch crossing module

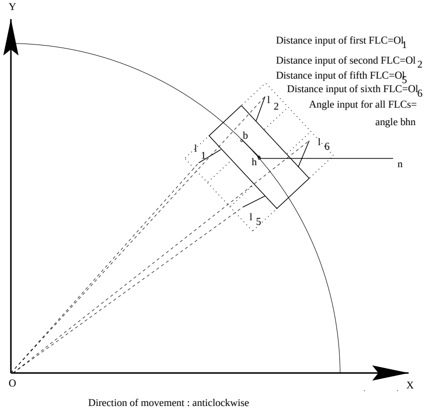
Y
O
X
b
h
n
l_1
l_2
l_5
l_6
Distance input of first FLC=Ol_1
Distance input of second FLC=Ol_2
Distance input of fifth FLC=Ol_5
Distance input of sixth FLC=Ol_6
Angle input for all FLCs= angle bhn
Direction of movement : anticlockwise

Figure 9: A schematic showing inputs of an FLC - turning gait generation module

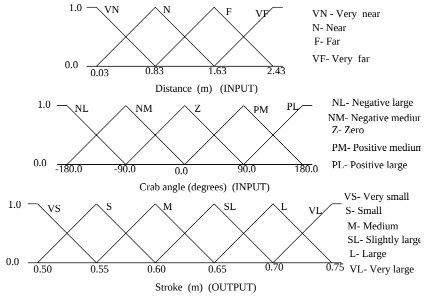

Figure 10: Author-defined membership function distributions for input and output of the FLCs - turning gait generation module

20-bits in this string represent the information regarding path generation, the next 120-bits give gait generation information in the ditch crossing mode and the remaining 120-bits carry information regarding turning gait generation.

We combine both objectives (minimizing travel time T and maximizing z) as follows:

$$\text{Maximize} \qquad f = \alpha_1 \frac{1}{T} + \alpha_2 z, \qquad (4)$$

where α_1 and α_2 are two weighting factors. We use $\alpha_1 = \alpha_2 = 1$ here, and adjust w_1 and w_2 (equation 3) to give two objectives equal importance. This problem may be better posed as a multi-objective optimization problem and multi-objective GA implementations can be used to find multiple Pareto-optimal solutions (Srinivas and Deb, 1995). But, here, we do not make the an already complex problem more complicated, instead use the above weighting scheme to solve the resulting single-objective optimization problem.

Each solution is evaluated to calculate a function value f_j for H different scenarios ($j = 1, 2 \ldots, H$). The fitness of the GA-string is assigned as $FS = \left(\sum_{j=1}^{H} f_j\right)/H$. After each solution in the population is evaluated and fitness is assigned, the population is modified by using three operators, namely reproduction, crossover and mutation (Goldberg, 1989).

5 RESULTS AND DISCUSSIONS

In all GA runs, the population size, crossover probability, mutation probability, maximum number of generations are set to 100, 0.95, 0.02, 50, respectively. The effectiveness of the proposed algorithm is tested through simulations. Moreover, Q and a are assumed to be equal to 15 and 0.333 ms^{-2}, respectively. It is also assumed that the vehicle accelerates and decelerates during the first 3 seconds and the last 3 seconds of its travel, respectively. Thus, the maximum velocity of the robot comes out to be 1 ms^{-1}. After a careful study, we select w_1 and w_2 (in equation 3) to be 1.0 and 6.0, respectively. We consider $H = 10$ different scenarios (in which the size and location of the obstacles and the ditch are varied) during the tuning phase. We have studied two different approaches, as discussed below:

Table 1: Author-defined rule base for an FLC

(Path generation module)

angle

distance	LT	AL	A	AR	R
VN	A	AR	AL	AL	A
N	A	A	AL	A	A
F	A	A	AR	A	A
VF	A	A	A	A	A

(Ditch crossing module)

relative angle

distance	NL	NM	Z	PM	PL
VN	M	SL	L	SL	M
N	S	SL	L	SL	S
F	S	M	L	M	S
VF	S	S	SL	S	S

(Turning gait module)

crab angle

distance	NL	NM	Z	PM	PL
VN	VS	S	M	S	VS
N	S	M	SL	M	S
F	M	SL	L	SL	M
VF	SL	L	VL	L	SL

Approach 1: Author-defined FLC: In this approach, a fixed set of 260 rules (refer to Table 1) and author-defined membership functions (Figs. 5, 8, and 10) are used.

Approach 2: Tuning rule base alone of FLCs: In this study, we optimize the rule base of the FLCs keeping the membership function distributions same as shown in Figs. 5, 8 and 10, using a GA. The maximum number of possible rules is 260 and a GA finds through search for which (and how many) rules from these 260 rules will result in a situation in which the hexapod will plan its path and gaits simultaneously, in an optimal sense after satisfying the constraint of stability margin.

The combined problems of path and gait generations are solved using both the approaches mentioned above and the results are presented in Table 3. In this table, three scenarios (out of 10) used during the optimization process are shown in the first three rows. The subsequent three rows show three different and new scenarios, which are not used during the optimization process. In scenario 6 (Table 3), it is seen that the author-defined FLCs have failed to generate stable gaits for the hexapod, whereas the GA-tuned FLCs have successfully done it. In all cases, the GA-tuned FLCs are found to perform better than the author-defined FLCs. It happens because the author-defined rule base of the FLCs may be far from being optimal. Table 2 shows the GA-tuned rule base of an FLC used for path generation. The obstacles 1 and 2 are approaching the robot from its left side, whereas obstacle 3 is approaching from the right side. This fact is reflected on the optimized rule base (Table 2). There are three rules when the input angle is LT and three other rules when the input angle is AR and R. The optimized rule bases for first through sixth FLCs used in the ditch crossing module are shown in Table 4. It is also intuitive to note that there are still some redundant rules in the optimized rule base and a second stage GA-based tuning will further reduce the number of rules. Similarly, Table 5 shows the optimized rule bases obtained for first through sixth FLCs, respectively, in the turning gait generation module. Thus, a GA has selected only 115 rules (6 for path generation and 109 for gait generations) out of 260 author-defined rules. The generated path and gaits obtained using Approach 2 for the test scenario 5 (Table 3) at the 25-th, 40-th, and 79-th motion segment are shown in Fig. 11. It is interesting to note that in Approach 2, the vehicle reaches its destination in 79-th motion segment, whereas in Approach 1, it is on its mid-way in the same 79-th motion segment (Figure 12). Fig. 13 shows the comparison of both approaches when started from the same initial configuration. The figure shows the superiority of the proposed approach (Approach 2).

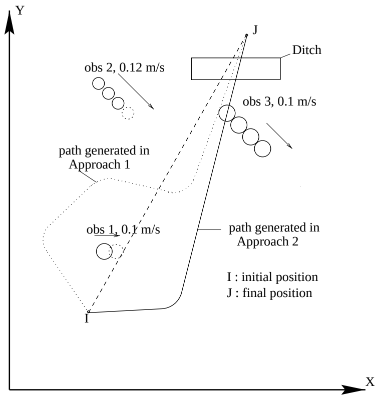

Figure 13: Comparison of overall trajectory generated by Approaches 1 and 2.

6 Conclusions

The proposed algorithm is able to solve the combined problem of path and gait generations simultaneously of a hexapod effectively. Simulation results show that a GA-tuned FLC has performed better than an author-defined FLC. As optimization is done off-line, the proposed algorithm is suitable for on-line implementation. As a fuzzy logic controller is less expensive computationally, the proposed algorithm will be computationally quicker compared to the traditional methods of gait generation. Moreover, rule-base optimization involves the problem of dealing with discrete variables

Table 2: Optimized rule base (having six rules) for an FLC - path generation module

angle

distance	LT	AL	A	AR	R
VN				AL	A
N	A			A	
F	A				
VF	A				

Table 3: Number of ground-legs, C, average kinematic margin of ground-legs, K, traveling distance, D (m) and time, T (sec) obtained by two approaches

Scenario	Approach 1				Approach 2			
	C	K	D	T	C	K	D	T
1	336	1.317	37.280	40.280	333	1.343	37.168	40.168
2	548	1.379	42.529	45.529	329	1.316	37.068	40.068
3	340	1.284	37.398	40.398	333	1.337	37.156	40.156
4	551	1.408	42.900	45.900	332	1.338	37.153	40.153
5	550	1.389	42.572	45.572	333	1.352	37.198	40.198
6	-	-	-	-	331	1.320	37.117	40.117

Table 4: Ditch crossing module: Optimized rule base for first to sixth FLC obtained using Approach 2.

(First leg)

relative angle

distance	NL	NM	Z	PM	PL
VN		SL	L		
N	S	SL			S
F	S	M			S
VF	S		SL	S	

(Second leg)

relative angle

distance	NL	NM	Z	PM	PL
VN		SL	L		M
N		SL	L		S
F	S	M			
VF	S		SL		S

(Third leg)

relative angle

distance	NL	NM	Z	PM	PL
VN	M	SL			M
N		SL	L		
F		M			S
VF		S		S	S

(Fourth leg)

relative angle

distance	NL	NM	Z	PM	PL
VN	M			SL	M
N		SL			
F	S	M	L		
VF				S	

(Fifth leg)

relative angle

distance	NL	NM	Z	PM	PL
VN			L		
N		SL			
F	S	M			
VF			SL	S	

(Sixth leg)

relative angle

distance	NL	NM	Z	PM	PL
VN	M				M
N	S	SL			S
F					S
VF		S		S	

Table 5: Turning gait generation module: Optimized rule base for first to sixth FLCs obtained using Approach 2.

(First leg)

crab angle

distance	NL	NM	Z	PM	PL
VN	VS		M		VS
N	S		SL		S
F	M	SL	L		
VF			VL		SL

(Second leg)

crab angle

distance	NL	NM	Z	PM	PL
VN	VS	S	M	S	
N		M	SL	M	
F			L		M
VF	SL			L	

(Third leg)

crab angle

distance	NL	NM	Z	PM	PL
VN	VS	S	M	S	
N		M	SL		S
F			L	SL	
VF				L	

(Fourth leg)

crab angle

distance	NL	NM	Z	PM	PL
VN				S	VS
N	S				
F					
VF	SL				

(Fifth leg)

crab angle

distance	NL	NM	Z	PM	PL
VN		S	M		VS
N					
F		SL	L		
VF		L		L	

(Sixth leg)

crab angle

distance	NL	NM	Z	PM	PL
VN		S	M	S	
N		M	SL	M	
F		SL	L	SL	
VF	SL	L	VL		

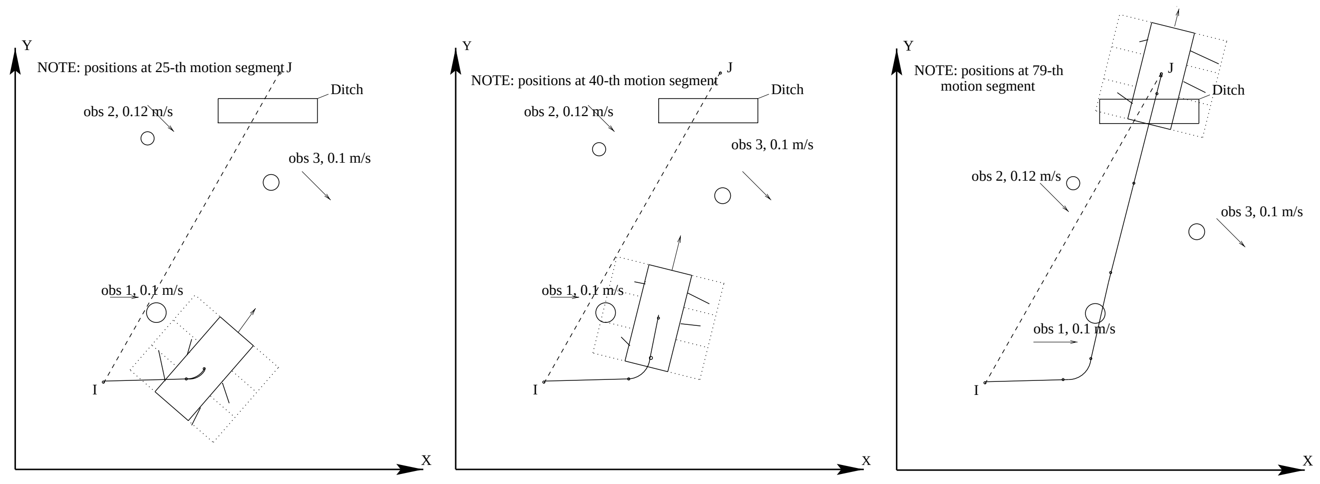

Figure 11: Generated path and gaits obtained using Approach 2 for test scenario 5 (Table 3)

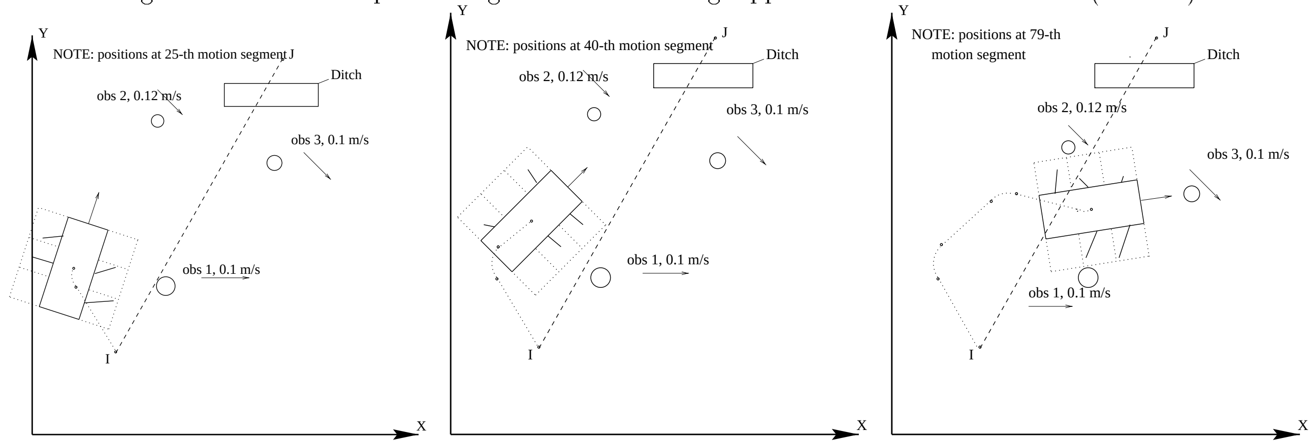

Figure 12: Generated path and gaits obtained using Approach 1 for test scenario 5 (Table 3)

and other complex tasks. This paper has shown that a combination of GAs and fuzzy logic technique can be efficiently used to make the problem modular and tractable to solve. In this study, a GA tries to find a set of good rules through search from a manually constructed large rule base. The approach can be made more flexible by allowing GAs to discover rules by optimally choosing an output linguistic for input linguistic combinations, which we are pursuing currently.

References

1. D. E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA, Addison-Wesley Publishing.

2. B. Kosko (1994). *Neural networks and fuzzy systems.* New Delhi, Prentice-Hall.

3. F. Herrera, E. Herrera-Viedma, M. Lozano, J. L. Verdegay (1994). Fuzzy tools to improve genetic algorithms. *Proc. of the Second European Congress on Intelligent Techniques and Soft Computing*, 1532-1539.

4. C. Karr (1991). Design of an adaptive fuzzy logic controller using a genetic algorithm. *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, San Mateo CA, Morgan Kaufmann Publishers, 450-457.

5. D. J. Cho, J. H. Kim, D. G. Gweon (1995). Optimal turning gait of a quadruped walking robot. *Robotica*, 13, 559-564.

6. L. Magdalena, J. R. Velasco (1996). Fuzzy rule-based controllers that learn by evolving their knowledge base. In F. Herrera and J. L. Verdegay (eds.) *Genetic Algorithms and Soft Computing*, 172-201. Physica-Verlag.

7. R. Tomovic, W. J. Karplus (1961). Land Locomotion Simulation and Control. *Proc. Third International Analogue Computation Meeting*, Opatija, Yugoslavia, 385-390.

8. N. Srinivas and K. Deb (1995). Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation* 2(3): 221-248.