Fuzzy Control of A quadruped Robot Foot Trajectory

S. Mohammad Jabbaryfar
Alumnus
Electrical Engineering School
Sharif University of Technology
Tehran, Iran, 11365-11155
Email: M.Jabbarifar@yahoo.com

Saeed Bagheri Shouraki
Professor
Electrical Engineering School
University
Tehran, Iran, 11365-11155
Email: bagheri-s@sharif.edu

Ali Meghdari
Professor
Mechanical Engineering School
Sharif university of Technology
Tehran, Iran, 11365-11155
Email: meghdari@sharif.edu

Abstract— **The progressive deployment of robotic industry and the capabilities of quadruped robots led us to introduce an approach to use fuzzy finite state machine for determining gait parameters of a quadruped robot. In this method, each step of robot- in a trot gait- is represented by two parameters: the maximum length (L) and height (H) of the foot tip in a step. Generated fuzzy rules were used for calculating those parameters by which the robot could pass unknown routes. To apply this algorithm, a simple robot is simulated in WEBOTS and finite surfaces with different slopes and friction coefficients were designed. After obtaining suitable parameters by which the robot could pass the surfaces, the rule base was generated. By implementing the resulted fuzzy if-then rules, the robot could pass three distinct unknown routes. The results convinced us that fuzzy approach can make the control of quadruped robot easier and further experiments can be performed to give the ability of passing rougher terrains to a quadruped robot which may decrease the complexity and cost of the robot.**

Keywords-component; Fuzzy Finite State Machine; quadruped Robot; gait.

I. INTRODUCTION (HEADING 1)

Nowadays, by increasing the applications of robots in military, industry and also home usages, researchers are looking for the most efficient algorithms to control the robots. As Prof. Zadeh presented the fuzzy systems [1], fuzzy control has been developed and now is one of the most applicable procedures in new aspects of controlling because of its less required computational capacity and more proper stability. Although the human like and quadruped robots are more applicable because of their coincidence with more difficult conditions compared to the wheeled robots, more complicated methods should be implemented to control them. Some previous efforts have used artificial neural networks for generating leg movement [2] and static gait [3]. Raibert used table methods to choose the foot angels with ground in a monoped. In his method, the desired touch angels which could minimize the error function based on the real and suitable jump were obtained by searching in a table of touch angels [4]. Recently fuzzy clustering methods have been used to control a monoped robot jumping by means of some sample movements [5]. The primary idea of both methods [4], [5] was learning from samples and it seems there was a lack of human perception. Passino and Yurkovich used fuzzy algorithms to move a robot moving with gallop gait [6]. In this method the capability of fuzzy methods was used in modeling nonlinear systems between leg movement and running speed. It is not required to know the mechanical model to control the robot but the system's physics should be known. In contrast with learning from samples [4], [5] this fuzzy controller, has capability of online learning of rules. Marhefka et al. controlled the gallop gaits of a simulated quadruped robot. In this method, they used fuzzy algorithms to model the complicated nonlinear relations between inputs and outputs, because in this way it was not required to calculate exact dynamic model of system [7]. Herr and MacMahon used finding suitable angle of front and rear leg of robot, touching the ground to control a quadruped robot [8-9]. Orin and Krasny utilized genetic algorithm in order to search and find suitable gain and angels parameters in a search space [10]. Palmer et al. developed fuzzy algorithm on a real monoped robot and used intelligent methods in real world [11]. Hornby et al. implemented walking gait on a Sony quadruped robot. This gait was applied by embedded electrical system but it required manually gain adjustment [12]. Tsujita et al. used nonlinear oscillator to control a quadruped robot turning motion, which had three-DOF legs and a joint in its body, when it was moving slow and fast [13]. Fukuoka implemented a neural oscillator in order to control the dynamic walking of a quadruped robot named Tekken based on biological concepts. This robot could control the direction of its movement by changing the angel of its yaw joint [14]. Fuzzy Finite State Machine (FFSM) or Fuzzy Automata was introduced by wee [15]. Finite States Machine (FSM) is a computational model of the machine with an initial internal memory which defines some finite states for the machine. Firstly, the initial state would be defined for the system and by changing inputs the state of the system will be changed based on the model. In Fuzzy Finite States Machine, this procedure is more complicated. In fuzzy concepts, the system does not stand thoroughly in a defined state rather has the membership levels versus different states and based on these membership levels, the next state of the system should be defined.

As mentioned above, researchers used diverse algorithms of control such as classic methods, Artificial Neural Networks, Revolutionary and fuzzy algorithms to make the legged robot more efficient and intelligent. But mechanical and electrical complexity of such robots has made the control procedure complicated and expensive. In this article, it was tried to introduce a fuzzy method to make the required hardware and software simpler and cheaper by dividing the conditions into several states and control the robot in each state to make the algorithm simpler. Thus a simple simulated quadruped robot was manually controlled to pass several different surfaces by changing its steps parameters, length (L) and height (H). Based on the previous stage data, membership functions and fuzzy rules were extracted. Then the robot could cross three unknown surfaces.

II. METHOD

First, a simple quadruped robot was designed in SOLIDWORKS. The robot has four two-degree of freedom leg (as shown in Fig. 1). Hip and knee were simulated by revolute joints with parallel axis. Then it was exported in Webots and 8 servomotors were placed in joints. A GPS sensor was also embedded to measure traversed distance. All four legs are similar. Fig. 1 illustrates this robot at the slopes.

A. *Input, Output and Fuzzy Rules Extraction*

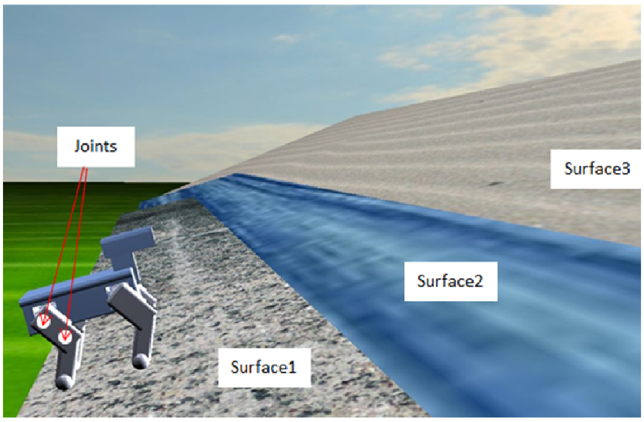


Figure 1. Simulated robot in Webots and 3 different slopes

Two different methods were considered, in first way, all joint angles are inputs and the changes of angles are outputs of fuzzy system. Another input was required that shows whether the last movement of robot has been forward or not. Totally, there were 9 inputs and 8 outputs. In each state a human controller used keyboard and tried to move forward the robot by changing the angles and finally he extracted the fuzzy rules from his experiences [16]. Such method is extremely time consuming and complex. He should have done the same steps for all states.

Eventually, each state corresponds to a subsystem and in operating phase, new inputs trigger related states on the basis of the degree of membership of the inputs in the rule-antecedents. Then the outputs are calculated.

In second method, more suitable outputs were used to make the procedure of extracting control rules simpler. The length and height of each step were considered as outputs instead of angles. Using length and height helps us to use inverse kinematic and in this way there is no need the angles to be changed directly and everyone has a clear and visual perception of length and height of the step compared with joint angles to move the robot forward. Another change that simplifies our method and has no effect on algorithm is using similar length and height parameters for all legs. This simplification provides us with more practical and simpler fuzzy rules extracting. Fig. 2 shows this system. The slope and friction coefficient are inputs and the length and height of steps are outputs.

B. *Inverse Kinematic of Two-joint Leg*

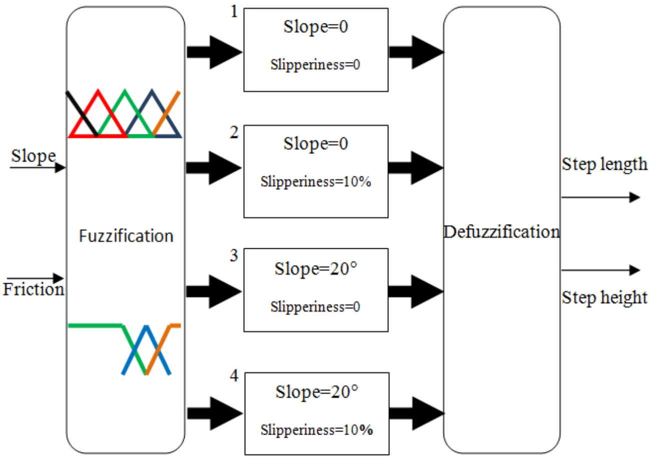


Figure 2. The control system with two inputs and two outputs

The Denavit-Hartenberg (D-H) notation was used in kinematic modeling (Fig. 3) and the inverse kinematic of one leg was calculated. After calculating homogeneous transformation matrix and solving inverse kinematic equations, joint angles Determined and they are given bellow:

$\theta_1 = \mathrm{Arctan2}(y, x) - \mathrm{Arctan2}\,(k_2, k_1)$

$\theta_2 = \mathrm{Arctan2}\,(\mathrm{Sin}\,(\theta_2), \mathrm{Cos}\,(\theta_2))$

Where:

$\mathrm{Cos}\,(\theta_2) = (x^2+y^2 - l_1^2 - l_2^2) / (2\times l_1\times l_2)$

$k_1 = l_1 + l_2\times\mathrm{Cos}\,(\theta_2)$ and $k_2 = l_2\times\mathrm{Sin}\,(\theta_2)$

x and y are the desired coordination of leg tip. l_1 and l_2 are respectively the length of the hip and leg of the robot as shown in Fig 3. k_1 and k_2 are auxiliary variables.

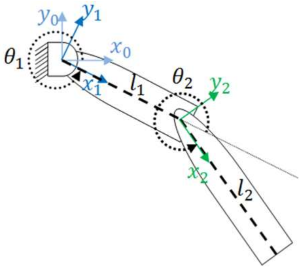


Figure 3. Denavit-Hartenberg notation on a schematic leg

C. Design of the Gait

Diverse gates could be used; nonetheless, selecting all of them has no significant effect on prime procedure of our algorithm, so trot was chosen because it is neither too complex like gallop nor too simple as walking.

Next, elliptical trajectory was selected. This trajectory provides a smooth movement and its major and minor radius respectively used as length and height of step. In swing phase the leg follows the upper half of ellipse (as shown in Fig. 4) even though in supporting phase the tip of the foot is immobile, the lower half of ellipse was considered as its trajectory in order to help the body to move forward. The major and minor semi axes were determined as the robot can pass the surfaces without slippage.

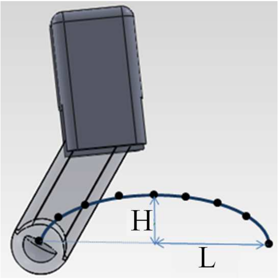


Figure 4. Ellipse trajectory of foot

D. Determination algorithm of trajectory parameters

In short, slope and friction coefficient are inputs and the length and height of foot tip are outputs. Each surface with its slope and friction is considered as a state.

At first state, the step parameters were changed, length and height, till the robot could pass maximum distance in 10 paces without slippage. Then the state was altered by changing the slope or friction or both and did the same procedure and found the best parameters. All the best parameters were noted in a table and finally the trapezoidal membership functions and extracted fuzzy rules were determined.

Afterward, input and output membership functions and fuzzy rules were applied in fuzzy toolbox of MATLAB. Mamdani method and max-min method for fuzzification and method of centroid for defuzzification were used to construct a fuzzy interface system (FIS).

If the simulations were not sufficient and proper, the robot cannot pass the new unknown surfaces successfully. So we must conduct new experiments and extract new rules till the robot can cross the terrain with new slop and friction. The slope (S) of surfaces was selected from 0 to 20 degree. Webots uses coulomb friction (CF) approximation to represent friction and CF was used ranging from 1 to 25. The bigger amount of this parameter is attributed to lower friction and strengthens the probability of slippage.

As first stage, slopes 0, 5, 10, 15, 20 and CF from 1, 5, 10, 15 and 20 were chosen for surfaces. For finding new rules, in second stage, surfaces which their slopes were similar to first stage were selected, but the CF amounts were changed to 1, 4, 7, 10, 13, 16, 19, 22 and 25. Therefore 5×5=25 and 5×9=45 states were respectively experimented in stage 1 and 2. Furthermore, three different new surfaces, with new S and CF, were designed to test the appropriateness of rules (as shown in Fig. 1) and they are given as follows.

Surface1: S=13°, CF=18.

Surface2: S=17°, CF=17.

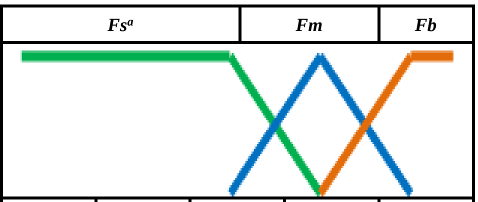
Surface3: S=22°, CF=13.

III. RESULTS

In order to obtain the best trotting parameters in each state we used expert knowledge for extracting fuzzy rules. First, we select a horizontal surface with CF=1. The robot could pass the surface by the maximum possible length, 50mm and 8mm step height. The robot paced 1560 in the direction of X in 10 complete steps. Next, we changed The CF to 5, 10, 15 and 20 and repeated the experiments. Increase in the amount of CF caused the robot to slip, so we had to decrease the length and height of the steps to prevent slippage and result in the decrease in passing distance. For instance, the robot completely slipped on a horizontal surface with CF=20 and with L=50mm. So we reduced L to 30 to avoid slippage and the robot paced 1150mm.

In table I, all suitable L and H parameters for each state are shown. Next we used this table and extracted fuzzy rules. On the basis of values, membership functions were determined which were shown in table I.

TABLE I. INPUTS AND OUTPUTS MEMBERSHIP FUNCTIONS AND SIMULATIONS RESULTS AT STAGE 1

Coulomb Friction		***Mem. Func.***	***Fs***[a]						***Fm***		***Fb***	
												
		Value	1		5		10		15		20	
Slope ***Mem. Func.***		***Value***	L	H	L	H	L	H	L	H	L	H
Svs[a]		0	50	8	50	8	50	8	40	5	30	8
Ss		5	50	8	50	8	50	8	30	5	20	5
Sm		10	50	8	50	8	50	8	30	5	20	6
Sb		15	50	8	50	8	50	8	50	3	50	2
Svb		20	50	5	50	5	50	5	50	4	40	3

a. F: Friction, S: Slope, s: small, m: medium, b: big, v: very

Fuzzy rules and membership functions were applied in fuzzy toolbox of MATLAB as it is mentioned in Method section. Because of 3 membership functions for friction and 5 membership functions for slope we had 15 fuzzy rules. Three sample rules are given bellow.

1. If Friction is small and Slop is very small then Length is very big and Height is very big.

2. If Friction is small and Slop is very big then Length is very big and Height is very big.

3. If Friction is big and Slop is medium then Length is small and Height is small.

Afterwards, test surfaces parameters were applied as inputs to validate fuzzy rules and membership functions. The results indicated that fuzzy rules were not suitable and the robot slipped on the first surface and could not walk and just passed 300mm in 25 steps. Thus we did more experiments. In other words, we needed more data to know the conditions to edit fuzzy rules and membership functions. The result of the second stage of the experiments and membership functions are shown in table II.

Then, 5 friction and 5 slope membership functions and 25 fuzzy rules were entered in MATLAB. To confirm the capability of new rules and membership functions we used surfaces like previous stage. In second stage the robot successfully crossed three surfaces. The robot passed 1790mm through 25 steps in direction of x and 790mm in direction of y. Fig. 5 shows the procedure and results in this stage. First, the inputs and degrees of membership determined. 9 states were triggered by 3 different inputs parameters. Next, suitable length and height of steps related to each surfaces calculated by defuzzification.

IV. DISCUSSION AND CONCLUSION

In this paper, a trot gait was selected for a simple simulated quadruped robot to pass different surfaces. Each leg uses an ellipsoid trajectory to move and the axis of the ellipse represents the length and height of steps. An expert human tried different length and height on diverse surfaces in order that the robot could pass the surfaces successfully. Results of this experiment were used to extract the fuzzy rules and membership functions were extracted. At first stage the robot was not able to pass the test surfaces because the simulated surfaces were not sufficient and suitable, so new fuzzy rules and membership functions were determined on the basis of new length and height parameters on more surfaces. Eventually the robot passed three test surfaces which had new slope and friction coefficient values. The properties of the last test surface were chosen beyond tried surfaces to know the operation of fuzzy rules out of knowledge space and the robot had superficial slippage in last steps which could be vanished by simulating some new surfaces.

We avoid complexity in all simulations because the primary goal was to design a practical fuzzy algorithm for finding suitable gait parameters in a quadruped robot. For instance, simulated robot had three degrees of freedom in each leg but it was only needed to move in one direction so utilizing two joints in each leg was enough and had no effect on the results and it just simplified the kinematic equations. The mechanical details of robot were also neglected. The selected gait and surfaces were tried to be neither complex nor simple.

One of the main advantages of fuzzy algorithms is that they have no need to know the exact dynamic model of robot, so in this study, classic control and stability were ignored. In other words, fuzzy methods decrease the complexity of the control of the robot. FFSM was used in order to make the method simpler. Selecting length and height of the step as gait parameters made the determining procedure of fuzzy rules, membership functions and control algorithm more simple and practical.

It is clear that passing more complicated routes requires extending knowledge about the conditions and extracting more fuzzy rules. The results indicate that passing more complicated surfaces is possible by using more simulations. The concept of Information granularity which was presented by Zadeh can be used to find the least required rules in order to discard redundant rules and make the procedure simpler.

In future studies, simulating more detailed and intelligent robots, by using more complex surfaces and different gaits can improve the capabilities of robot.

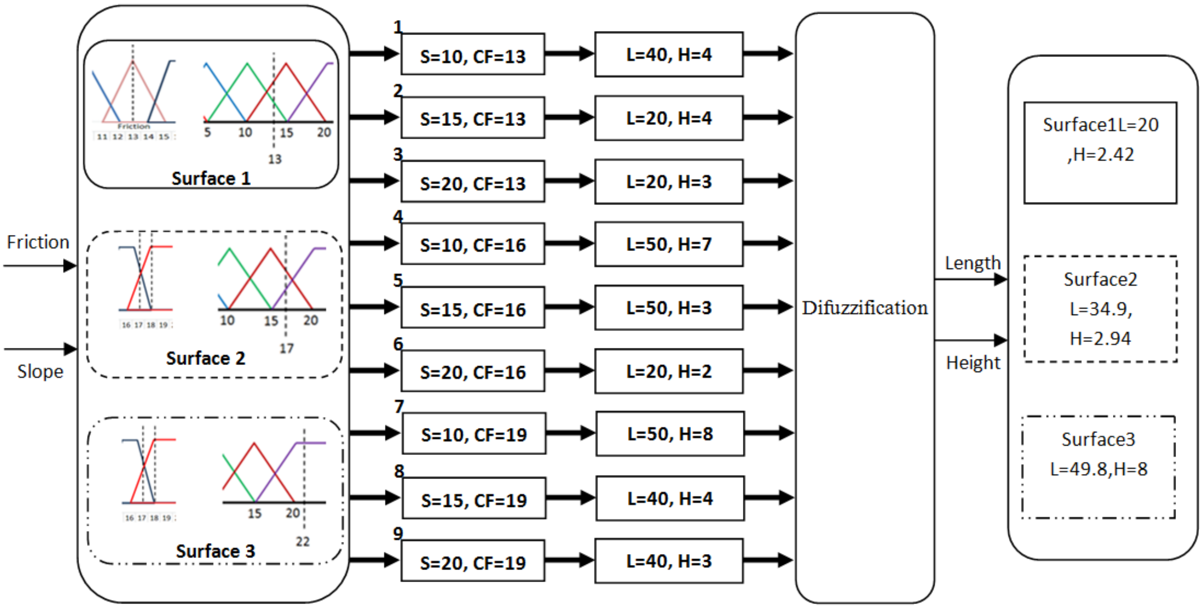


Figure 5. The procedure of calculating desired outputs and the results

TABLE II. INPUTS AND OUTPUTS MEMBERSHIP FUNCTIONS AND SIMULATIONS RESULTS AT STAGE 2

<table>
<tr><td colspan="2" rowspan="3">Coulomb Friction</td><td rowspan="2"><i>Mem. Func.</i></td><td colspan="9"><i>Fvs</i>^{a}</td><td><i>Fs</i></td><td colspan="3"><i>Fm</i></td><td colspan="3"><i>Fb</i></td><td colspan="2"><i>Fvb</i></td></tr>
<tr><td colspan="18"></td></tr>
<tr><td><i>Value</i></td><td colspan="2">1</td><td colspan="2">4</td><td colspan="2">7</td><td colspan="2">10</td><td colspan="2">13</td><td colspan="2">16</td><td colspan="2">19</td><td colspan="2">22</td><td colspan="2">25</td></tr>
<tr><td colspan="3">Slope</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td><td rowspan="2">L</td><td rowspan="2">H</td></tr>
<tr><td colspan="2"><i>Mem. Func.</i></td><td><i>Value</i></td></tr>
<tr><td><i>Svs</i>^{a}</td><td rowspan="5"></td><td>0</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>40</td><td>5</td><td>30</td><td>8</td><td>30</td><td>8</td><td>30</td><td>8</td><td>30</td><td>8</td></tr>
<tr><td><i>Ss</i></td><td>5</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>40</td><td>4</td><td>30</td><td>3</td><td>20</td><td>3</td><td>20</td><td>3</td><td>20</td><td>3</td></tr>
<tr><td><i>Sm</i></td><td>10</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>40</td><td>4</td><td>20</td><td>4</td><td>20</td><td>3</td><td>20</td><td>3</td><td>20</td><td>1</td></tr>
<tr><td><i>Sb</i></td><td>15</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>7</td><td>50</td><td>3</td><td>20</td><td>2</td><td>20</td><td>2</td><td>20</td><td>1</td></tr>
<tr><td><i>Svb</i></td><td>20</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>50</td><td>8</td><td>40</td><td>4</td><td>40</td><td>3</td><td>40</td><td>3</td><td>40</td><td>3</td></tr>
</table>

REFERENCES

[1] L.A. Zadeh, “Fuzzy sets*,”. Information and Control, 1965. 8(3): p. 338-353.

[2] W. Ilg, T. Muhlfriedel, and K. Berns. “A hybrid learning architecture based on neural networks for adaptive control of a walking machine,” in Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on. 1997.

[3] M.A. Lewis, A.H. Fagg, and A. Solidum. “Genetic programming approach to the construction of a neural network for control of a walking robot,” in Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on. 1992.

[4] M.H. Raibert, H.B. Brown, and M. Chepponis, “Experiments in balance with a 3D one-legged hopping machine,” The International Journal of Robotics Research, 1984. 3(2): p. 75.

[5] P.I. Doerschuk, W.E. Simon, V. Nguyen, and A. Li, “A modular approach to intelligent control of a simulated jointed leg,” Robotics & Automation Magazine, IEEE, 1998. 5(2): p. 12-21.

[6] K.M. Passino, and S. Yurkovich, “Fuzzy control,” 1998: Citeseer.

[7] D.W. Marhefka, D.E. Orin, J.P. Schmiedeler, and K.J. Waldron, “Intelligent control of quadruped gallops,” Mechatronics, IEEE/ASME Transactions on, 2003. 8(4): p. 446-456.

[8] H.M. Herr, and T.A. McMahon, “A trotting horse model,” The International Journal of Robotics Research, 2000. 19(6): p. 566.

[9] H.M. Herr, and T.A. McMahon, “A galloping horse model,” The International Journal of Robotics Research, 2001. 20(1): p. 26.

[10] D.P. Krasny, and D.E. Orin, “A 3D galloping quadruped robot,” Climbing and Walking Robots, 2006: p. 467-474.

[11] L.R. Palmer, D.E. Orin, D.W. Marhefka, J.P. Schmiedeler, K.J. Waldron, “Intelligent control of an experimental articulated leg for a galloping machine,” 2003: IEEE.

[12] G.S. Hornby, et al. “Evolving robust gaits with AIBO,” in Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. 2000.

[13] K. Tsujita, H. Toui, and K. Tsuchiya, “Dynamic turning control of a quadruped locomotion robot using oscillators,” Advanced Robotics, 2005. 19(10): p. 1115-1133.

[14] Y. Fukuoka, H. Kimura, and A.H. Cohen, “Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts,” The International Journal of Robotics Research, 2003. 22(3-4): p. 187.

[15] W. Wee, and K. Fu, “An adaptive procedure for multiclass pattern classification,” IEEE Transactions on Computers, 1968: p. 178-182.

[16] R.M. Zand, and S.B. Shouraki. “Designing a fuzzy logic controller for a quadruped robot using human expertise extraction,” in Electrical Engineering (ICEE), 2013 21st Iranian Conference on. 2013.