Reinforcement Learning of Parameters for Humanoid Rhythmic Walking based on Visual Information

Masaki Ogino[1], Yutaka Katoh[1], Minoru Asada[1,2] and Koh Hosoda[1,2]
[1]Dept. of Adaptive Machine Systems, [2]HANDAI Frontier Research Center,
Graduate School of Engineering, Osaka University
{ogino, yutaka}@er.ams.eng.osaka-u.ac.jp, {asada, hosoda}@ams.eng.osaka-u.ac.jp

Abstract

This paper presents a method for learning the parameters of rhythmic walking to generate a purposive motion. The controller consists of the two layers. Rhythmic walking is realized by the lower layer controller which adjusts the speed of the phase on the desired trajectory depending on the sensor information. The upper layer controller learns (1) the feasible parameter sets that enable a stable walking for a robot, (2) the causal relationship between the walking parameters to be given to the lower layer controller and the change of the sensor information, and (3) the feasible rhythmic walking parameters by reinforcement learning so that a robot can reach to the goal based on the visual information. The method was examined in the real robot, and it learns to reach the ball and to shoot it into the goal in the context of RoboCupSoccer competition.

1 Introduction

Recently, a number of humanoid projects have started and various kinds of humanoid platforms have been developed. The typical method for real robot adopted in those platforms is planning the desired trajectory of each joint based on Zero Moment Point [3, 14]. In that method, the ZMP trajectory for not falling down is planned and the trajectory of each joint is calculated based on the ZMP trajectory. This method needs very precise dynamics parameters for the robot and much calculation time for planning.

The other method to realize bipedal walking is rhythmic-walking-based approach. This method doesn not use the precise structural parameters of a robot. Instead, the controller adjusts its inherent frequency depending on the sensor information so that the entrainment between dynamics of the controller and those of environment takes place. Taga et al. proposed the model of CPG (Central Pattern Generator) system [2] for human walking based on the nonlinear dynamics equations [11]. The network system changes its frequency depending on the sensor information. In the simulation experiment, this model realizes the stable walking under the various kinds of disturbances [6]. In the original Taga's CPG model, the output value of each neuron is used as a reference of torque applied to a corresponding joint. While almost all of the currently existing humanoid robots are driven by high gain PD controllers, instead of torque control. Therefore, it is difficult to apply Taga's CPG model to real robots directly. However, even such a robot with high gain PD controllers can realize the stable walking with a controller which utilizes sensor information properly. Pratt [9] realizes the energy efficient walking in real robot with a controller which consists of state machines. The state transition of the controller occurs when the swing leg touches the ground. Tsuchiya et al. [13] realized stable walking based on a method in which a trajectory controller determines the shape of the trajectory, and a phase controller changes the speed of the desired angle on the trajectory. In this controller the phase speed is adjusted by the sensor information.

In rhythmic walking, the control parameters are found heuristically, not by planning as ZMP approach. This makes it difficult to construct the upper layer controller to control the movement of a robot because the walking parameters such as walking step are not found until the robot interacts with the real world. Taga [12] and Fukuoka et al. [4] constructs the upper layer controller which gives the control parameters to the lower CPG controller depending the visual information so that the robot can avoid obstacles or climb over a step. In these methods, the adjusting parameters were given by the designer in advance. However, for making a more adaptive robot to the dynamic situations, it is necessary that the relationship between the parameters of the lower rhythmic walking controller and the resultant change of the environment should be learned.

In this paper, the layered controller is introduced, in which the lower controller realizes rhythmic walking based on the controller proposed by Tsuchiya et al. [13] and the upper controller learns the parameters of the controller of the lower layer based on the visual information. There are three points in learning of the upper layer controller. (1) In the first stage, it learns the feasible parameters of the lower layer controller which enables a robot to walk. (2) To accelerate a learning process, the upper layer controller learns the model of the world : the relationship between the control parameters given to the lower rhythmic walking controller and the change of the visual sensor information. (3) The upper layer controller learns what parameters should be given to reach a goal by the reinforcement learning.

The rest of this paper is organized as follows. First, the lower controller which enables a rhythmic walk is introduced. Next, we describe the upper layer controller in which the parameters of the lower controller is learned by reinforcement learning. Then, the suggested controller is applied to the RoboCupSoccer task [8], "approaching to a ball", and experimental results are shown. Finally, conclusions are given.

2 Rhythmic walking controller

2.1 Biped robot model

Fig. 1 shows a biped robot model used in the experiment which has one-link torso, two four-link arms, and two six-link legs. All joints are single DOF rotation ones. Each foot has four FSRs to detect reaction force from the floor and a CCD camera with a fish-eye lens is attached at the top of the torso.

2.2 Rhythmic walking controller based on CPG principle

Here, we build a lower-layer controller based on the controller proposed by Tuchiya et al. [13]. The proposed controller consists of two sub-controllers: *a trajectory controller* and *a phase controller* (Fig. 2). The trajectory controller outputs the desired trajectory of each limb depending on the phase which is given by the phase controller. The phase controller consists of four oscillators, each of which is responsible for movement of each limb (Fig. 4). Each oscillator changes its speed depending on the touch sensor signal, and the effects reflected on the oscillator in each limb. As a result, the desired trajectory of each joint is adjusted so that global entrainment between dynamics of the robot and those of the environment is realized. In the following, the details of each controller are explained.

2.2.1 Trajectory controller

The trajectory controller calculates the desired trajectory of each joint depending on the phase given by the corresponding oscillator in the phase controller.

Here, the trajectory of each joint is characterized by four parameters as shown in Fig. 3. For joints 3, 4 and 5, of which axes coincide with pitch axis, the desired trajectory is determined so that in the swing phase the foot trajectory draws a ellipse that has the radiuses, h in vertical direction and β in horizontal direction, respectively. For joints 2 and 4, of which axes coincide with roll axis, the desired trajectory is determined so that the leg tilts from $-W$ to W relative to the vertical axis. The desired trajectory of joint 1 is determined by the amplitude of the oscillation, α. The desired trajectories are summarized as following functions,

$$\begin{aligned}
\theta_1 &= \alpha \sin(\phi) && (1)\\
\theta_2 &= W \sin(\phi) && (2)\\
\theta_i &= f_i(\phi, h, \beta) \qquad (i = 3, 4, 5) && (3)\\
\theta_6 &= -W \sin(\phi). && (4)
\end{aligned}$$

The detail of f_i is explained in Appendix. Among four parameters described above, α, which determines the walking step length, and β, which determines the walking direction are selected as rhythmic parameters of walking. Although these parameters characterize approximate direction and step length, resultant walking is not as precisely determined by those parameters because of the slips between the support leg and the ground. These parameters are learned in the upper layer learning module, explained in **3**.

2.2.2 Phase controller

The phase which determines the desired value of each joint is given by the phase controller. The phase controller consists of two oscillators, ϕ_R for right leg and ϕ_L for left leg. The dynamics of each oscillator is determined by basic frequency, ω, the interaction term between two oscillators, and the feedback signal from sensor information,

$$\begin{aligned}
\dot{\phi}_L &= \omega - K(\phi_L - \phi_R - \pi) + g_L && (5)\\
\dot{\phi}_R &= \omega - K(\phi_R - \phi_L - \pi) + g_R. && (6)
\end{aligned}$$

The second term of RHS in above equations keeps the phases of two oscillators in opposite. The third term, feedback signal from sensor information, is given as follows:

$$g_i = \begin{cases} K' Feed_i & (0 < \phi < \phi_C) \\ -\omega(1 - Feed_i) & (\phi_C \leq \phi < 2\pi) \end{cases} \qquad (7)$$
$$i = \{R, L\},$$

where K', ϕ_C and $Feed_i$ denote feedback gain, the phase when the swing leg contacts with the ground, and the feedback sensor signal, respectively. $Feed_i$ returns 1 if the FSR sensor value of the corresponding leg exceeds the certain threshold value, otherwise 0. The third term enables that the mode switching between the free leg phase and the support one happens appropriately according to the ground contact information from the FSR sensors. In this paper, the value of each parameter is set as follows; $\phi_C = \pi$, $\omega = 5.23$[rad/sec], $K = 15.7$, $K' = 1$.

3 Reinforcement learning with rhythmic walking parameters

3.1 Principle of reinforcement learning

Reinforcement learning has recently been receiving increased attention as a method for robot learning with little or no *a priori* knowledge and higher capability of

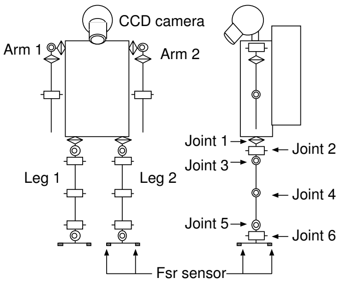


Figure 1: Model of biped locomotion robot

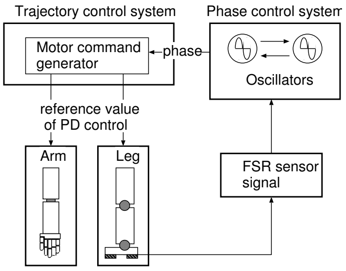
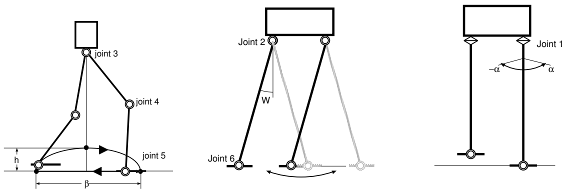


Figure 2: Walking control system



(a) Joint angles around pitch axis

(b) Joint angles around roll axis

(c) Joint angle around yaw axis

Figure 3: Joint angles

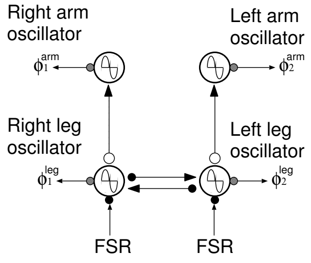


Figure 4: Phase control system

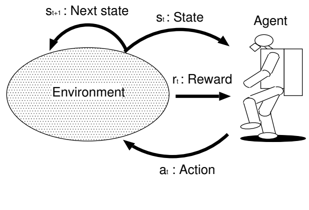


Figure 5: Basic model of agent-environment interaction

reactive and adaptive behaviors. Fig. 5 shows the basic model of robot-environment interaction [10], where a robot and environment are modelled by two synchronized finite state automatons interacting in a discrete time cyclical processes. The robot senses the current state $s_t \in \boldsymbol{S}$ of the environment and selects an action $a_t \in A$. Based on the state and action, the environment makes a transition to a new state $s_{t+1} \in \boldsymbol{S}$ and generates a reward r_t that is passed back to the robot. Through these interactions, the robot learns a purposive behavior to achieve a given goal. In order for the learning to converge correctly, the environment should satisfy the Markovian assumption that the state transition depends on only the current state and the taken action. The state transition is modelled by a stochastic function $\boldsymbol{T}$ which maps a pair of the current state and the action to take to the next state ($\boldsymbol{T} : \boldsymbol{S} \times \boldsymbol{A} \rightarrow \boldsymbol{S}$). Using $\boldsymbol{T}$, the state transition probability $P_{s_t,s_{t+1}}(a_t)$ is given by

$$P_{s_t,s_{t+1}}(a_t) = Prob(\boldsymbol{T}(s_t, a_t) = s_{t+1}). \tag{8}$$

The immediate reward r_t is given by the reward function in terms of the current state by $R(s_t)$, that is $r_t = R(s_t)$. Generally, $P_{s_t,s_{t+1}}(a_t)$ (hereafter $\mathcal{P}^a_{ss'}$) and $R(s_t)$ (hereafter $\mathcal{R}^a_{ss'}$) are unknown.

The aim of the reinforcement learner is to maximize the accumulated summation of the given rewards (called *return*) given by

$$return(t) = \sum_{n=0}^{\infty} \gamma^n r_{t+n}, \tag{9}$$

where $\gamma\,(0 \leq \gamma \leq 1)$ denotes a discounting factor to give the temporal weight to the reward.

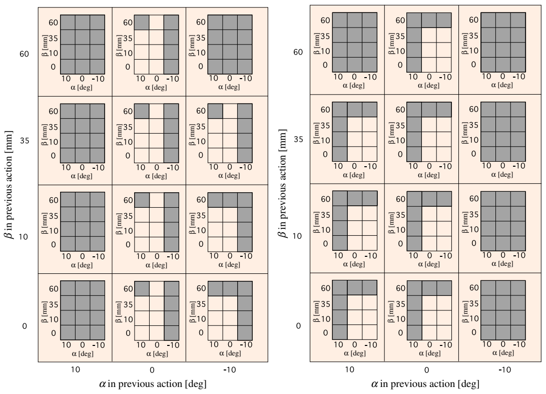
If the state transition probability is known, the optimal policy which maximize the expected *return* is given by finding the optimal value function $V^*(s)$ or the optimal action value function $Q^*(s, a)$ as follows. The derivation of them can be found elsewhere [10].

$$\begin{aligned} V^*(s) &= \max_a E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} \hat{\mathcal{P}}^a_{ss'} \left[\hat{\mathcal{R}}^a_{ss'} + \gamma V^*(s') \right] \qquad (10) \end{aligned}$$

$$\begin{aligned} Q^*(s, a) &= E\{r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a\} \\ &= \sum_{s'} \hat{\mathcal{P}}^a_{ss'} \left[\hat{\mathcal{R}}^a_{ss'} + \gamma \max_{a'} Q^*(s', a') \right] \qquad (11) \end{aligned}$$

3.2 Construction of action space based on rhythmic parameters

The learning process has two stages. The first one is to construct the action space consisting of feasible combinations of two rhythmic walking parameters (α, β). To do that, we prepared the three-dimensional posture space s_p in terms of the forward length β (quantized into four lengths: 0, 10, 35 60 [mm]), the turning angle α (quantized into three angles: -10, 0, 10 [deg]) both of which mean the previous action command, and the leg side (left or right). Therefore, we have 24 kinds of postures. Firstly, we have constructed the action space of the feasible combinations of (α, β) excluding the infeasible ones which cause collisions with its own body. Then, various combinations of actions are examined for stable walking in the real robot. Fig. 6 shows the feasible actions (empty boxes) for each leg corresponding to the previous actions. Due to the differences in physical properties between two legs, the constructed action space was not symmetric although it should be theoretically.



(a) left leg

(b) right leg

Figure 6: Experimental result of action rule

3.3 Reinforcement learning with visual information

Fig. 7 shows an overview of the whole system which consists of two layers: adjusting walking based on the visual information and generating walking based on neural oscillators. The state space consists of the visual information s_v and the robot posture s_p, and adjusted action a is learned by dynamic programming method based on the rhythmic walking parameters (α, β). In a case of ball shooting task, s_v consists of ball substates and goal substates both of which are quantized as shown in Fig. 8. In addition to these substates, we add two more substates, that is, "the ball is missing" and "the goal is missing" because they are necessary to recover from loosing their sight.

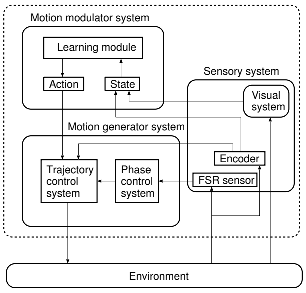
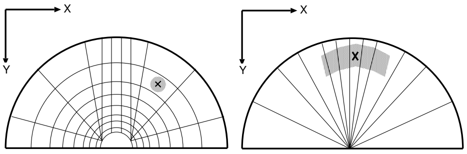


Figure 7: Biped walking system with visual perception



(a) State space of ball

(b) State space of goal

Figure 8: State space of ball and goal

Learning module consists of a planner which determines an action a based on the current state s, a state transition model which estimates the state transition probability $\mathcal{P}^a_{ss'}$ through the interactions, and a reward model (see Fig. 9). Based on DP, the action value function $Q(s,a)$ is updated and the learning stops when no more changes in the summation of action values.

$$Q(s,a) = \sum_{s'} \mathcal{P}^a_{ss'}[\mathcal{R}_s + \gamma \max_{a'} Q(s',a')], \tag{12}$$

where $\mathcal{R}_s$ denote the expected reward at the state s.

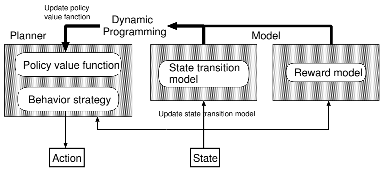


Figure 9: Learning module

4 Experiments

4.1 Robot platform and environment set-up

Here, we use a humanoid platform HOAP-1 by Fujitsu Automation LTD. [7] attaching a CCD camera with a fish-eye lens at the head. Figs. 10 and 11 show a picture and a system configuration, respectively. The height and the weight are about 480[mm] and 6[kg], and each leg (arm) has six (four) DOFs. Joint encoders have high resolution of 0.001[deg/pulse] and reaction force sensors (FSRs) are attached at soles. The colour image processing to detect an orange ball and a blue goal is performed on the CPU (Pentium3 800MHz) under RT-Linux. Fig. 12 shows an on-board image.

The experimental set-up is shown in Fig. 13 where the initial robot position is inside the circle whose center and radius are the ball position and 1000 [mm], respectively, and the initial ball position is located less than 1500 [mm] from the goal of which width and height are 1800 [mm] and 900 [mm], respectively. The task is to take a position just before the ball so that the robot can shoot a ball into the goal. Each episode ends when the robot succeeds in getting such positions or fails (touches the ball or the pre-specified time period expires).

4.2 Experimental results

One of the most serious issues in applying the reinforcement learning method to real robot tasks is how

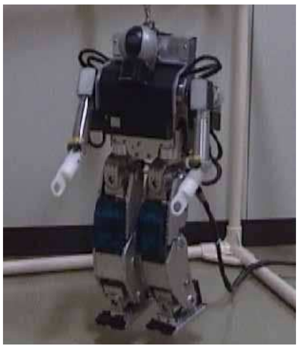


Figure 10: HOAP-1

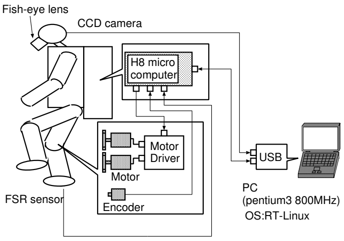


Figure 11: Overview of robot system

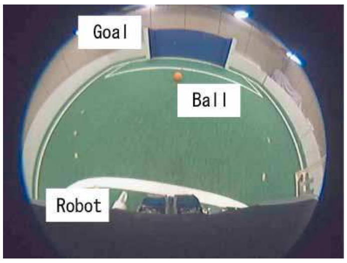


Figure 12: Robot's view (CCD camera image through fish-lens)

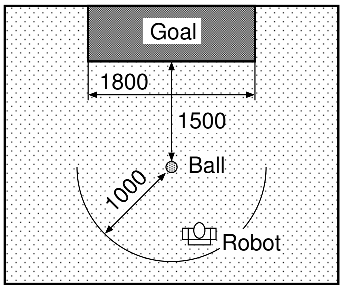
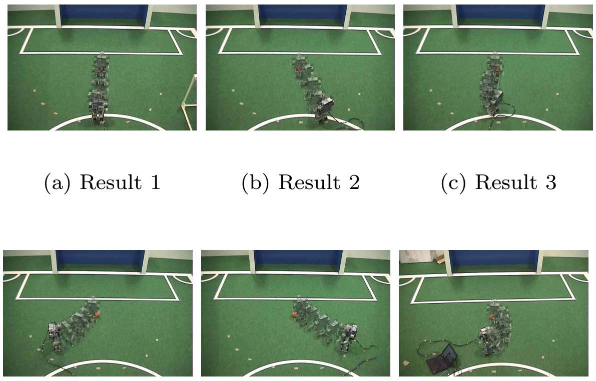


Figure 13: Experimental environment

to accelerate the learning process. Instead of using Q-learning that is most typically used in many applications, we use a DP approach based on the state transition model $\mathcal{P}^{a}_{ss'}$ that is obtained separately from the behavior learning itself. Further, we give the instructions to start up the learning, more correctly, during the first 50 episodes (about a half hour), the human instructor avoids the useless exploration by directly specifying the action command to the learner about 10 times per one episode. After that, the learner experienced about 1500 episodes. Owing to the state transition model and initial instructions, the learning converged in 15 hours, and the robot learned to get the right position from any initial positions inside the half field.

Fig. 14 shows the learned behaviors from various initial positions. In Fig. 14, the robot can capture the image including both the ball and the goal from the initial position while in Fig. 14 (f) the robot cannot see the ball or the goal from the initial position.



(a) Result 1 (b) Result 2 (c) Result 3

(d) Result 4 (e) Result 5 (f) Result 6

Figure 14: Experimental results

5 Concluding remarks

A vision-based behavior of humanoid was generated by reinforcement learning with rhythmic walking parameters. Since the humanoid generally has many DOFs, it is very hard to control all of them. Instead of using these DOFs as action space, we adopted rhythmic walking parameters, which drastically reduces the search space and therefore the real robot learning was enabled in reasonable time. In this study, the designer specified the state space consisting of visual features and robot postures. State space construction by learning is one of the future issues.

Acknowledgments

This study was performed through the Advanced and Innovational Research program in Life Sciences from the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.

References

[1] A. Fujii, A. Ishiguro. Evolving a cpg controller for a biped robot with neuromodulation. In *Climbing and Walking Robots* (2002), pp. 17–24.

[2] S. Grillner. Neurobiological Bases of Rhythmic Motor Acts in Vertebrates. *Science*, Vol. 228, (1985), pp. 143-149.

[3] S. Kajita and K. Tani. Adaptive Gait Control of a Biped Robot based on Realtime Sensing of the Ground Profile. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems* (1996), pp. 570-577.

[4] H. Kimura, Y. Fukuoka, and H. Nakamura. Biologically inspired adaptive dynamic walking of the quadruped on irregular terrain. In *Proc. of 9th International Symposium of Robotics Research* (1999), pp. 271-278.

[5] M. Laurent and J. A. Thomson. The role of visual information in control of a constrained locomotor task. *J. Mot. Behav*, Vol. 20, (1988), pp. 17–37.

[6] S. Miyakoshi, G. Taga, Y. Kuniyoshi, and A. Nagakubo. Three Dimensional Bipedal Stepping Motion using Neural Oscillators –Towards Humanoid Motion in the Real World–. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems* (1998), pp. 84-89.

[7] Y. Murase, Y. Yasukawa, K. Sakai, etc. Design of a Compact Humanoid Robot as a Platform. In *19th conf. of Robotics Society of Japan*, (2001), pp. 789-790.

[8] H. Kitano, M. Asada The RoboCup humanoid challenge as the millennium challenge for advanced robotics. In *Advanced Robotics*, vol. 13, (2000), no. 8, pp. 723-736.

[9] J. Pratt Exploiting Inherent Robustness and Natural Dynamics in the Control of bipedal Walking Robots Doctor thesis, MIT, June. (2000).

[10] Richard S.Sutton and Andrew G.Barto. "Reinforcement learning:An Introduction", MIT Press/Bradford Books, March, (1998).

[11] G. Taga, Y. Yamaguchi, H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, Vol. 65, (1991), pp. 147–159.

[12] G. Taga. A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, Vol. 78, (1998), pp. 9–17.

[13] K. Tsuchiya, K. Tsujita, K. Manabu, S. Aoi. An emergent control of gait patterns of legged locomotion robots. *IAV2001*, pp. 271-276, (2001), pp. 271-276.

[14] J. Yamaguchi, N. Kinoshita, A. Takanishi, and I. Kato. Development of a Dynamic Biped Walking System for Humanoid –Development of a Biped Walking Robot Adapting to the Human's Living Floor–. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems* (1996), pp. 232-239.

Appendix: planning the reference trajectory around the pitch axis

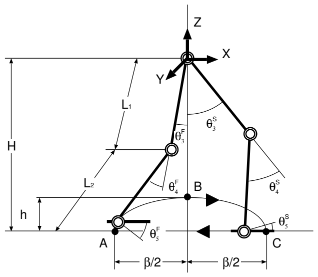


Figure 15: Joint angles and the reference trajectory of the foot

The reference trajectories of joints 3, 4 and 5 are determined by the position of the foot. Let x and z be the position of the foot in the plane XZ which is perpendicular to the pitch axis, the reference trajectory of the foot is given by,

$$\begin{aligned} x_F &= \frac{\beta}{2}\cos(\phi^F), \\ z_F &= -H + h\sin(\phi^F), \\ x_S &= -\frac{\beta}{2}\cos(\phi^S), \\ z_S &= -H, \end{aligned}$$

where (x_F, z_F) and (x_S, z_S) are the positions of the foot in the free and support phase, respectively, H is the length from the ground to the joint 3, β is the step length, and h is the maximum height of the foot from the ground (Fig. 15). When the position of the foot is determined, the angle of each joint to be realized is calculated by the inverse kinematics as follows,

$$\begin{aligned} \theta_3 &= \frac{\pi}{2} + atan2(z, x) - atan2(k, x^2 + z^2 + L_1^2 - L_2^2) \\ \theta_4 &= atan2(k, x^2 + z^2 - L_1^2 - L_2^2) \\ \theta_5 &= -(\theta_3 + \theta_4), \end{aligned}$$

where k is given by the following equation,

$$k = \sqrt{(x^2 + z^2 + L_1^2 + L_2^2)^2 - 2\{(x^2 + z^2)^2 + L_1^4 + L_2^4\}}.$$

In this research, the value of each parameter is set as follows; $H = 185$[mm], $h = 8$[mm], $W = 13$[deg], $L_1 = 100$[mm], $L_2 = 100$[mm].