Vision-Based Reinforcement Learning for Humanoid Behavior Generation with Rhythmic Walking Parameters

Masaki Ogino[1], Yutaka Katoh[1], Masahiro Aono[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, aono}@er.ams.eng.osaka-u.ac.jp, {asada, hosoda}@ams.eng.osaka-u.ac.jp

***Abstract*—This paper presents a method for generating vision-based humanoid behaviors by reinforcement learning with rhythmic walking parameters. The walking is stabilized by a rhythmic motion controller such as CPG or neural oscillator. The learning process consists of two stages: the first one is building an action space with two parameters (a forward step length and a turning angle) that inhibits combinations that are not feasible. The second is reinforcement learning with the constructed action space and the state space consisting of visual features and posture parameters to find feasible actions. The method is applied to a situation of the RoboCupSoccer Humanoid league [6], that is, to approach the ball and to shoot it into the goal. Instructions by human are given to start up the learning process and the rest is completely self-learning in real situations.**

I. Introduction

Since the debut of the Honda humanoid [3], the research community for biped walking has been growing and various approaches have been introduced. Among them, there are two major trends in biped walking. One is a model based approach with ZMP (zero moment point) principle [4] or the inverted pendulum model [15] both of which plan the desired trajectories and control their bipeds to follow them. To stabilize the walking, these methods need very precise dynamics parameters for both the robot and its environment.

The other one is inspired by the findings [2] in neurophysiology that most animals generate their walking motions based on the central pattern generator (hereafter, CPG) or neural oscillator. CPG is a cluster of neural structures that oscillate each other under the constraint of the relationships in their phase spaces and generate rhythmic motions that interact with the external environment. The observed motion can be regarded as a result of the entrainment between robot motion and the environment. This sort of approach does not need model parameters that are as precicise as ZMP or the inverted pendulum.

Taga et al. [12] gave the mathematical formulation for the neural oscillator, constructed a dynamic controller for biped walking on the sagittal plane, and showed the simulation results which indicated that his method could generate stable biped motions similar to human walking. Others extended his method to three dimensions and adaptive motion on the slope by adjusting the neural oscillator [1].

The second approach seems promising for adaptation against changes in the environment. To handle more complicated situations, visual information has been used. Taga [13] studied how the robot can avoid an obstacle by adjusting the walking pattern assuming that the object height and the distance to it can be measured by the visual information. Fukuoka et al. [5] also adjusted CPG input so that a quadruped can climb over a step through the visual information. In these methods, however, the adjusting parameters were given by the designer in advance. Therefore, it seems difficult to apply them to more dynamic situations, and learning method seems necessary.

This paper presents a method for generating vision-based humanoid behaviors by reinforcement learning with rhythmic walking parameters. A rhythmic motion controller such as CPG or neural oscillator stabilizes the walking [14]. The learning process consists of building an action space with two parameters (a forward step width and a turning angle) so that infeasible combinations are inhibited, and reinforcement learning with the constructed action space and the state space consisting of visual features and posture parameters to find a feasible action. The method is applied to a situation from the Humanoid RoboCupSoccer league [6], that is, to approach the ball and to shoot it into the goal. Instructions by human are given to start up the learning process, and the rest is solely self-learning in real situations.

II. Rhythmic walking controller

A. Biped robot model

Fig. 1 shows a biped robot model used in the experiment which has a one-link torso, two four-link arms, and two six-link legs. All joints rotate with a single DoF. 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.

B. Rhythmic walking controller based on CPG principle

Here, we build a lower-layer controller based on the controller proposed by Tuchiya et al. [14]. The proposed controller consists of two sub-controllers: *a trajectory controller* and *a phase controller* (Fig. 2). The trajectory

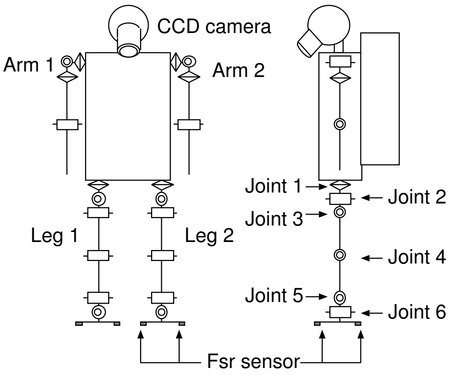


Fig. 1. Model of biped locomotion robot

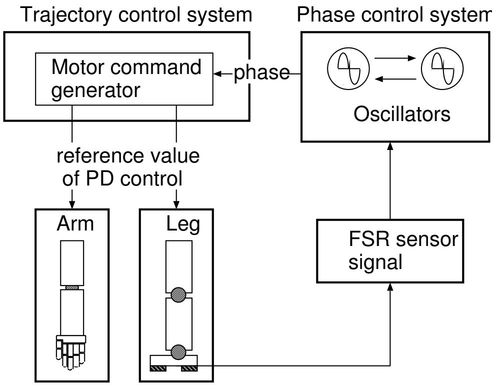
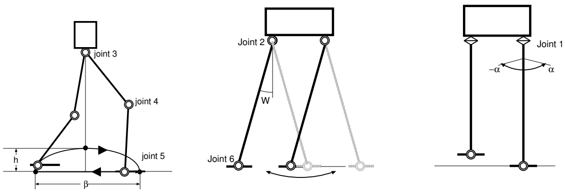


Fig. 2. Walking control system

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.

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, which coincide with pitch axis, the desired trajectory is determined so that in the swing phase the foot trajectory draws a ellipse that has the radii, h in the vertical direction and β in the horizontal direction, respectively. For joints 2 and 4, which coincide with roll axis, the desired trajectory is determined so that the leg tilts from $-W$ to W relative to the vertical axis. The amplitude of the oscillation, α, determines the desired trajectory of joint 1. The desired trajectories are summarized as following functions,



(a) Joint angles around pitch axis

(b) Joint angles around roll axis

(c) Joint angle around yaw axis

Fig. 3. Joint angles

$$\theta_1 = \alpha \sin(\phi) \quad (1)$$

$$\theta_2 = W \sin(\phi) \quad (2)$$

$$\theta_i = f_i(\phi, h, \beta) \qquad (i = 3,4,5) \quad (3)$$

$$\theta_6 = -W \sin(\phi). \quad (4)$$

The detail of f_i is explained in the Appendix. Among the 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, those parameteres does not determine resultant walking so precisely because of slippage between the support leg and the ground. These parameters are learned in the upper layer learning module, explained in section **III**.

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

$$\dot{\phi}_L = \omega - K(\phi_L - \phi_R - \pi) + g_L \quad (5)$$

$$\dot{\phi}_R = \omega - K(\phi_R - \phi_L - \pi) + g_R. \quad (6)$$

The second term on the RHS in the above equations ensures that the oscillators have opposite phases. The third term, feedback signal from sensor information, is given as follows:

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

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 a certain threshold value, otherwise 0. The third term enables the mode switching between the free leg phase, and the support leg phase 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], $K = 15.7,\ K' = 1$.

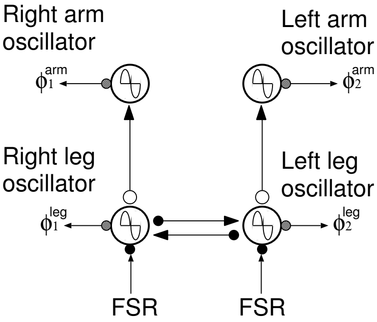


Fig. 4. Phase control system

III. Reinforcement learning with rhythmic walking parameters

A. The principle of reinforcement learning

Reinforcement learning has recently been receiving increased attention as a method of robot learning with little or no *a priori* knowledge and a higher capability for reactive and adaptive behaviors. Fig. 5 shows the basic model of robot-environment interaction [11], in which two synchronized finite state automatons interacting in a discrete time cyclical processes models a robot and environment. The robot senses the current state $s_t \in S$ 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 $s_{t+1} \in S$ 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. 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 action taken. A stochastic function T which maps a state-action pair to the next state ($T : S \times A \to S$) models the state transition. Using T, the state transition probability $P_{s_t,s_{t+1}}(a_t)$ is given by

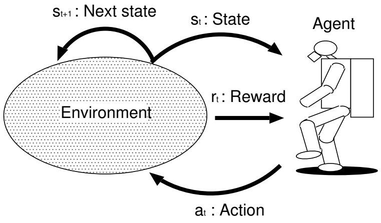


Fig. 5. Basic model of agent-environment interaction

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

The reward function gives the immediate reward, r_t, 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 $\mathscr{P}^a_{ss'}$) and $R(s_t)$ (hereafter $\mathscr{R}^a_{ss'}$) 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 that maximizes the expected *return* is given by finding the optimal value function $V^*(s)$ or the optimal action value function $Q^*(s,a)$ as follows. Their derivation can be found elsewhere [11].

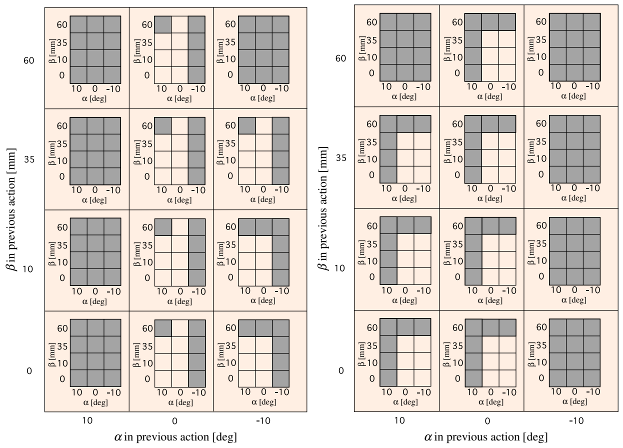
$$\begin{aligned} V^*(s) &= \max_a E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} \mathscr{P}^a_{ss'} \left[\mathscr{R}^a_{ss'} + \gamma V^*(s')\right] \end{aligned} \tag{10}$$

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

In this paper, the learning module examines the state transition when both feet contact with the ground. The state space, $\mathbf{S}$, consists of the visual information s_v and the robot posture s_p, and the action space consists of two parameters of rhythmic walking. Details are explained in the following subsections.

B. 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 $(\alpha,\ \beta)$. 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]), which are the previous action command and the leg side (left or right). Therefore, we have 24 kinds of postures. Firstly, we have excluded the infeasible combinations of (α, β), 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. Owing to the differences in physical properties between the two legs, the constructed action space was not symmetric, although theoretically it should be.



(a) left leg

(b) right leg

Fig. 6. Experimental result of action rule

C. 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 the dynamic programming (DP) method based on the rhythmic walking parameters (α, β). In the case of the ball shooting task, s_v consists of ball substates and goal substates, which are quantized as shown in Fig. 8. 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.

The learning module consists of a planner that determines an action a based on the current state s, a state transition model that estimates the state transition probability $\mathscr{P}^a_{ss'}$ 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 there are no more changes in the summation of action values.

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

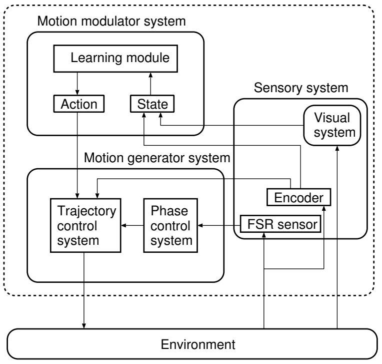
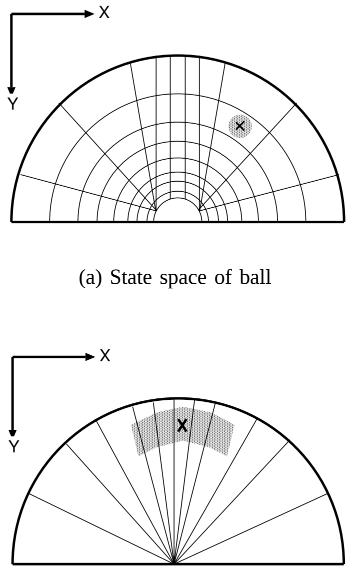


Fig. 7. Biped walking system with visual perception



(a) State space of ball

(b) State space of goal

Fig. 8. State space of ball and goal

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

IV. Experiments

A. Robot platform and environment set-up

We use a humanoid platform HOAP-1 by Fujitsu Automation Ltd. [9] 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 has six degrees-of-freecom and each arm has four. 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.

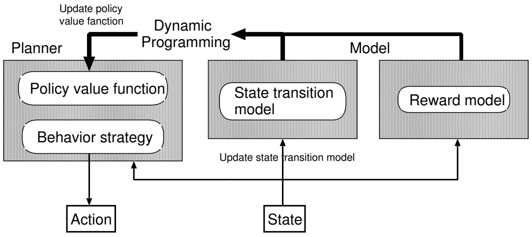


Fig. 9. Learning module

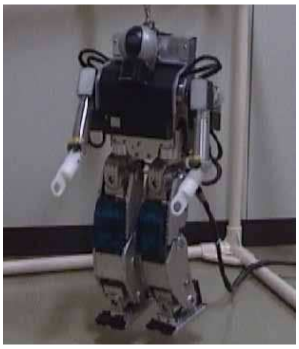


Fig. 10. HOAP-1

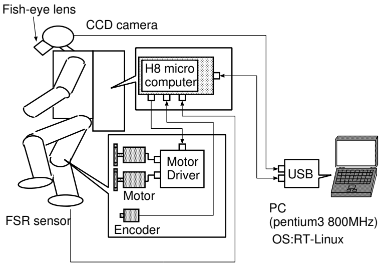


Fig. 11. Overview of robot system

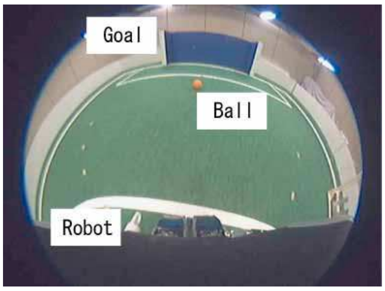


Fig. 12. Robot's view (CCD camera image through fish-lens)

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). The reward 1.0 is given to the robot when the robot reaches to the right position, otherwise 0.0.

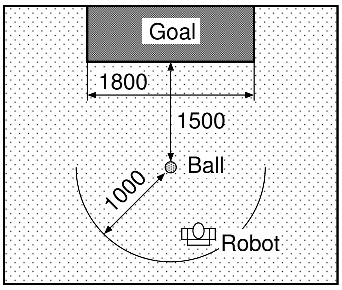


Fig. 13. Experimental environment

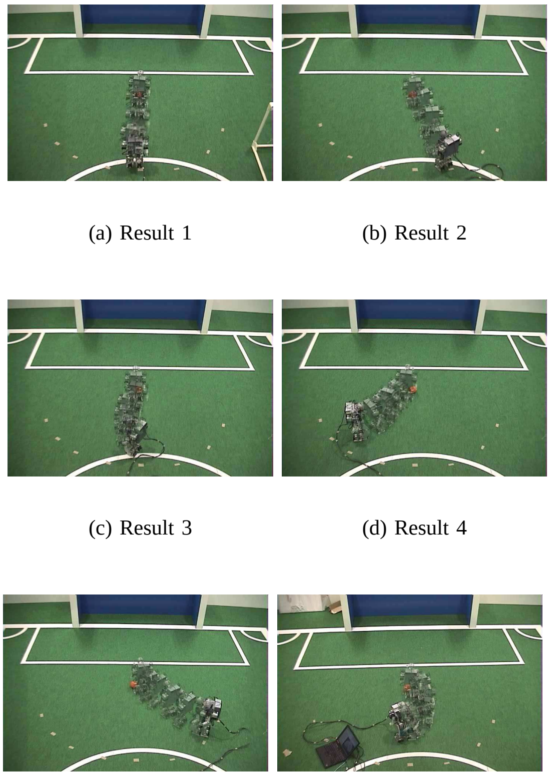
B. *Experimental results*

One of the most serious issues in applying the reinforcement learning method to real robot tasks is how 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 $\mathscr{P}^{a}_{ss'}$ that is obtained separately from the behavior learning itself. Further, we give the instructions to start up the learning: during the first 50 episodes (about a half hour), the human instructor avoids useless exploration by directly specifying the action command to the learner about 10 times per episode. After that, the learner experienced about 1500 episodes. Owing to the state transition model and initial instructions, 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.

V. Concluding remarks

Vision-based humanoid behavior 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 in the action space, we adopted rhythmic walking parameters, which drastically reduces the search space and,


(a) Result 1

(b) Result 2

(c) Result 3

(d) Result 4

(e) Result 5

(f) Result 6

Fig. 14. Experimental results

therefore, real robot learning was possible in a 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 of the Japanese Government.

VI. REFERENCES

[1] A. Fujii and A. Ishiguro, "Evolving a cpg controller for a biped robot with neuromodulation", in Proceedings of *5th International Conference on 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] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, "The development of honda humanoid robot" in *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, 1998, pp. 1321–1326.

[4] S. Kajita and K. Tani, "Adaptive gait control of a biped robot based on realtime sensing of the ground profile", in *Proceedings. of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996, pp. 570-577.

[5] H. Kimura, Y. Fukuoka, H. Nakamura, "Biologically inspired adaptive dynamic walking of the quadruped on irregular terrain", in *Proceedings of 9th International Symposium of Robotics Research*, 1999, pp. 271-278.

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

[7] M. Laurent and J. A. Thomson, "The role of visual information in control of a constrained locomotor task", *Journal of Motor Behavior*, Vol. 20, 1988, pp. 17–37.

[8] S. Miyakoshi, G. Taga, Y. Kuniyoshi, A. Nagakubo, "Three dimensional bipedal stepping motion using neural oscillators –towards humanoid motion in the real world–", in *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, pp. 84-89.

[9] Y. Murase, Y. Yasukawa, K. Sakai, "Design of a compact humanoid robot as a platform", in *Proceedings of 19th Conference of Robotics Society of Japan*, 2001, pp. 789-790.

[10] J. Pratt, "Exploiting Inherent robustness and natural dynamics in the control of bipedal walking robots", Doctor thesis, MIT, June. 2000.

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

[12] 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.

[13] 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.

[14] K. Tsuchiya, K. Tsujita, K. Manabu, S. Aoi, "An emergent control of gait patterns of legged locomotion robots", in Proceedings of the symposium on Intelligent Autonomous Vehicles 2001, pp. 271-276, 2001, pp. 271-276.

[15] J. Yamaguchi, N. Kinoshita, A. Takanishi, 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 *Proceedings of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996, pp. 232-239.

APPENDIX: PLANNING THE REFERENCE TRAJECTORY AROUND THE PITCH AXIS

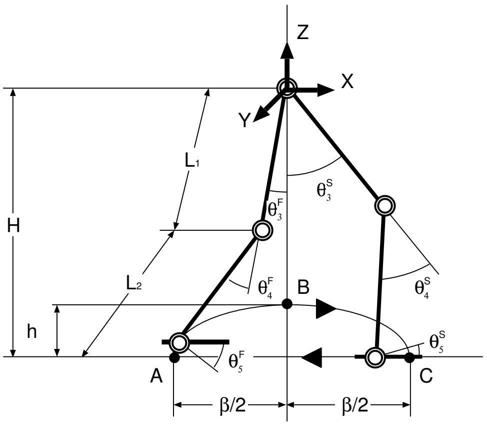


Fig. 15. Joint angles and the reference trajectory of the foot

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].