Learning algorithm for a brachiating robot

Hideki Kajima,[1] Yasuhisa Hasegawa,[2] Toshio Fukuda[1]

[1]Department of Micro System Engineering, Nagoya University, Nagoya, Japan; [2]Department of Mechanical and Systems Engineering, Gifu University, Gifu, Japan

Abstract: This paper introduces a new concept of multi-locomotion robot inspired by an animal. The robot, 'Gorilla Robot II', can select the appropriate locomotion (from biped locomotion, quadruped locomotion and brachiation) according to an environment or task. We consider 'brachiation' to be one of the most dynamic of animal motions. To develop a brachiation controller, architecture of the hierarchical behaviour-based controller, which consists of behaviour controllers and behaviour coordinators, was used. To achieve better brachiation, an enhanced learning method for motion control, adjusting the timing of the behaviour coordination, is proposed. Finally, it is shown that the developed robot successfully performs two types of brachiation and continuous locomotion.

Keywords: multi-locomotion robot, gorilla, brachiation, behaviour-based control, motion learning, continuous locomotion

Introduction

Our goal is to realise a robot that can intelligently assist and perform a heavy, boring and time-consuming task in place of a human. This useful robot needs to have good hardware (ie sensors, actuators and a processor) and good software (ie human-like intelligence information processing). Until now, very few of these have been realised. In the robotics field, the achievement of the dynamic and skilful motion of animals has been discussed in many research works, but it is still a challenge because of multiple degrees of freedom (DOF) and nonlinearity. 'Brachiation' is an interesting form of long-armed apes' locomotion using the arms to swing from branch to branch as shown in Figure 1. This motion is the kind of dynamic and dexterous action that apes perform as a lifework (Simons 1972). Figures 2 and 3 show a white-handed gibbon and its locomotion, respectively.

The focus of this paper is on behaviour generation architecture and its modification algorithm to realise a robot that can perform brachiation. The issue of the brachiation robot is closely related to dynamical tasks such as manipulation and underactuated systems (Paul 1981; Yamafuji et al 1992; Spong 1995; Arai et al 1998).
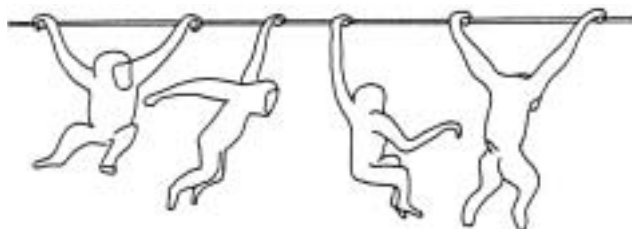

Figure 1 Gibbon locomotion.

Figure 2 The white-handed gibbon (*Hylobates lar*).

Figure 3 Brachiation of the white-handed gibbon.

Correspondence: Hideki Kajima, Department of Micro System Engineering, Nagoya University, Furo-cho 1, Chikusa-ku, Nagoya 464-8603, Japan; tel +81 52 789 4481; fax +81 52 789 3909; email kajima@robo.mein.nagoya-u.ac.jp

Figure 4 Brachiator II.

Several kinds of brachiation robot have been developed to research the brachiation controller. In pioneering research, the dynamics of locomotion were analysed and locomotion types proposed using a six-link brachiation robot (Fukuda, Hosokai et al 1991). Following this research was a two-link brachiation robot, 'Brachiator II', shown in Figure 4. This robot has one actuator at the elbow connecting two links, each of which has a gripper. Because the gripper cannot impose torque on the handhold directly, this is an underactuated mechanical system. A two-link brachiation robot was developed, and a heuristic method proposed, to find feasible motions (Fukuda, Saito et al 1991; Saito et al 1993, 1994). A self-scaling reinforcement learning algorithm was also proposed to achieve the robustness against some disturbances (Hasegawa et al 1999). As an analytical approach, a target dynamics method was proposed to solve the 'ladder', 'swing up', 'rope' and 'leap' problems, where the task was encoded as the output of the target dynamical system inspired by the pendulum-like motion of an ape's brachiation (Nakanishi et al 1998, 2000; Nakanishi and Fukuda 2000). In those works, a brachiating robot with

a dynamic multibody has not been explored. Considering the control of the higher degrees of freedom, the third brachiation robot, 'Brachiator III', was then developed (see Figure 5). Brachiator III consists of 13 links, 12 degrees of freedom and 14 motors, including two motors for a gripper control. The dimension and location of the joints are based on those of a long-armed ape. Hasegawa et al (2000), proposed an adaptation method that measures the effects of each local behaviour on the total behaviour and determines the activation level of each behaviour controller. The controller can adjust the amplitude of the behaviour controller. This method is effective for the online learning of a real robot, because the number of learning iterations for this method is less than that of other learning algorithms. However, it is important to adjust the amplitude and timing of the behaviour controller for realising more flexible coordination. Hence, the structure of the hierarchical behaviour controller was improved by adding timing parameters as well as amplitude.

In this paper, a new concept of the multi-locomotion robot inspired by an animal and a developed robot 'Gorilla Robot II', which can select the appropriate locomotion type according to an environment, are first introduced. Second, we propose an enhanced control method for dynamic motion control to achieve brachiation by adjusting the activation level and timing of the local behaviours. Finally, we show that the developed robot successfully improves two types of the brachiation (over-hand and side-hand).

Multi-locomotion robot

Many works on humanoid robots have been reported recently. Most of them focus on the achievement of a single type of locomotion, such as biped locomotion (Kajita and
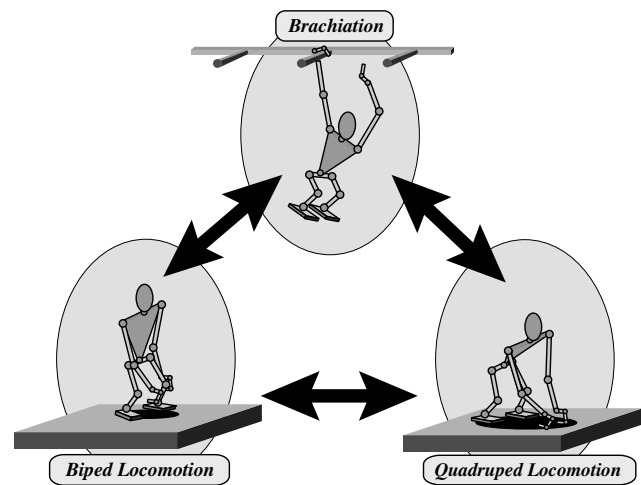

Figure 5 Brachiator III.

Figure 6 Concept of the multi-locomotion robot.

Tani 1996; Huang et al 1999; Kuffner et al 2001; Kanehira et al 2002). However, anthropoids, including humans, do various types of locomotion in daily life. In the same way, a multi-locomotion robot that can perform several types of locomotion is highly desirable for adaptive functioning in various situations. Figure 6 shows the concept of the multi-locomotion robot, which can choose the appropriate locomotion according to the situation. For example, it can brachiate across a river, biped walk in narrow space and quadruped walk in rough terrain. Such a robot is challenging and attractive.

'Gorilla Robot II' was developed as the most advanced multi-locomotion robot. It was designed to perform brachiation, biped locomotion and quadruped locomotion ('knuckle walking' using two arms and two feet). Figures 7 and 8 show an overview of Gorilla Robot II and its structure, respectively. Figure 9 shows the control system. This robot is about 1 m high, weighs 20 kg, and consists of 19 links and 20 AC motors. The real-time operating system VxWorks (Wind River Systems Inc, Alameda, CA, USA) runs on a
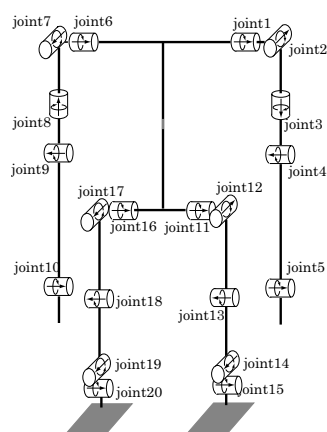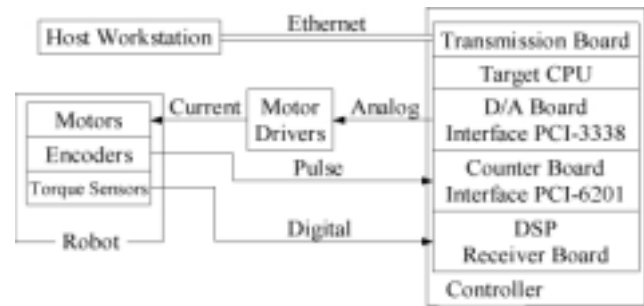

Figure 8 Structure of Gorilla Robot II.

Figure 9 Control system of Gorilla Robot II.

Pentium® III PC for processing sensory data and generating behaviours. Each arm has a torque sensor to recognise whether or not the robot catches a target branch during brachiation. Also, each leg has a torque sensor to measure the ground reaction force during biped and quadruped locomotion.

Motion learning

The focus of this research is on behaviour generation architecture and its adaptation algorithm to obtain dexterous animal-like motion. The model-based approach is such that the target dynamics can be one of the feasible approaches for design of the motion controller, but there are two major problems. The first is a modelling problem of the robot's dynamics: it is difficult to acquire a precise dynamical model of the multibody robot; as a result, much time is spent modelling and adjusting its parameters. The second is a design problem of the target dynamics: the target motion design of some representative parts in a multibody robot becomes complex, because the motion is constrained by a robot mechanism and its kinematics. Conversely, there have been some learning approaches introduced for motion control such as evolutionary algorithms, reinforcement learning algorithms and backpropagation methods for neural networks. These approaches can generate a robot controller from an evaluation index of the robot motion through an iterative searching process, but they have two limitations: learning time and controller size. They take many iteration processes for controller design and apply only to simple robot dynamics with a few DOF. Therefore, an online learning algorithm was proposed for the hierarchical behaviour controller, so that a multibody robot can adjust dynamically dexterous motion to the desired motion based only on some evaluation indices. The hierarchical behaviour controller with two layers is designed based on the behaviour-based approach. A control output for the dexterous motion is generated on the upper layer

coordinating some simple behaviours on the lower layer. The structure is explained in the next subsection.

Structure of hierarchical behaviour controller

We discuss brachiation as an objective task of Gorilla Robot II. The hierarchical behaviour controller is designed based on a behaviour-based approach, since the robot has multiple degrees of freedom and the objective task is complex. In the designing process, the brachiation behaviour is firstly divided into two actions: (1) 'preliminary swing', which stores sufficient energy prior to locomotion; and (2) 'locomotion', which is movement towards the next branch. These actions are composed of several local behaviours. 'Preliminary swing' consists of two behaviour controllers: 'leg swing' and 'body rotation 1'. 'Leg swing' is the up and down swinging motion of the legs. This behaviour controller drives all of the joints in both legs. 'Body rotation 1' is the rotational motion of the body to take suitable posture for the swing action, especially in the control of the direction of movement, controlling the yaw axis joints of both arms. Additionally, 'locomotion' consists of four behaviour controllers: 'leg stretch', 'body rotation 2', 'body lift' and 'arm reaching'. 'Leg stretch' is the motion of extending both legs. This behaviour controller, like 'leg swing', drives the leg joints. 'Body rotation 2' is the rotational motion of the body controlling the yaw axis joint of the supporting arm. 'Body lift' is a lift-up motion of the supporting arm, controlling shoulder and elbow joints of the supporting arm.

'Arm reaching' is the swinging motion of the free arm to get at the target branch and grasp it.

The behaviour controllers are feedforward controllers, which output the desired trajectories expressed by the cubic spline function to the feedback controllers. The feedback controller is a proportional-derivative (PD) controller and makes a corresponding actuator follow the desired trajectory. The video tracking system measures representative points to evaluate the performance of the robot. The evaluator then adjusts the desired trajectories according to a learning algorithm. The hierarchical behaviour controller for Gorilla Robot II is shown in Figure 10.

Desired trajectory adjustment

Firstly, we design the trajectory $y_k^m(t)$ and timing parameter t_k^m as a primitive input of the behaviour coordinator m based on a target motion. The behaviour coordinator m outputs the coefficients r_k^m and t_k^m to the behaviour controller k. In the lower layer, the behaviour controller k outputs the desired trajectory $yd_i^m(t)$ modified by the coefficients r_k^m and t_k^m to the actuator i. The desired trajectory $yd_i^m(t)$ for the actuator i included in the behaviour controller k is rescaled by two coefficients r_k^m and t_k^m from the behaviour coordinator m as follows:

$$yd_i^m(t) = r_k^m\left(\tilde{y}_k^m(t) - b_k^m(t)\right) \tag{1}$$

$$\tilde{y}_k^m(t) = y_k^m\left(\frac{\left(t_{st}^{m+1} + t_k^{m+1}\right) - \left(t_{st}^m + t_k^m\right)}{t_{st}^{m+1} - t_{st}^m}t\right) \tag{2}$$
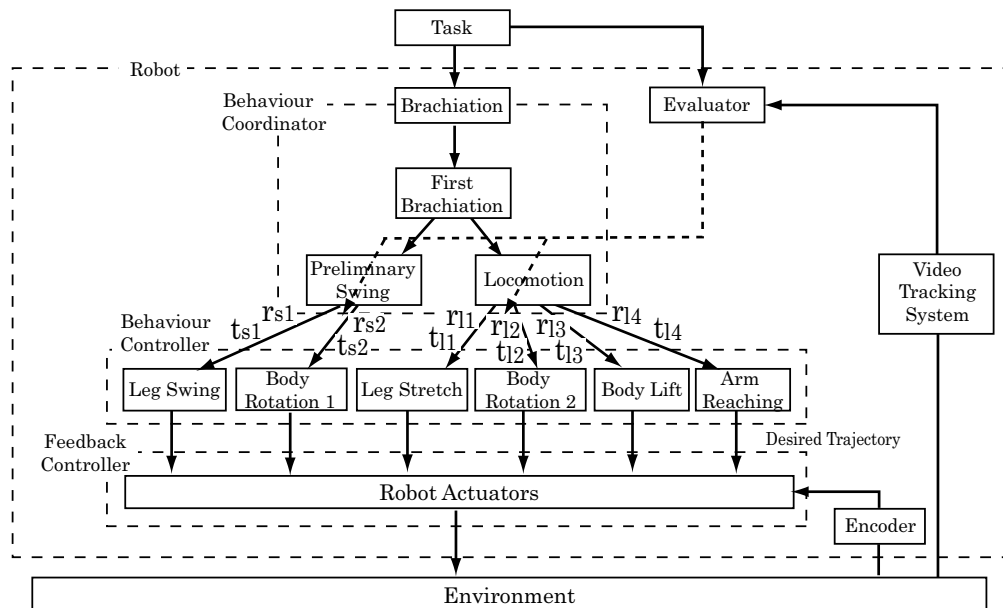

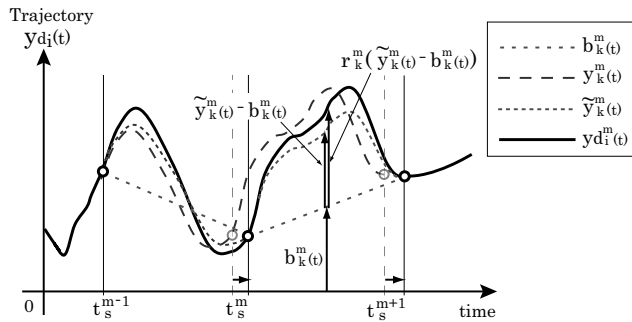**Figure 10** Hierarchical behaviour controller for Gorilla Robot II.

Figure 11 Adaptation of desired trajectory.

$$b_k^m(t) = y_k^m\left(t_{st}^m\right)\frac{\left(t_{st}^{m+1} + t_k^{m+1}\right) - t}{\left(t_{st}^{m+1} + t_k^{m+1}\right) - \left(t_{st}^m + t_k^m\right)}$$

$$+ y_k^m\left(t_{st}^{m+1}\right)\frac{t - \left(t_{st}^m + t_k^m\right)}{\left(t_{st}^{m+1} + t_k^{m+1}\right) - \left(t_{st}^m + t_k^m\right)} \tag{3}$$

where $y_k^m(t)$ is the primitive trajectory from the behaviour coordinator m to the behaviour controller k, $\tilde{y}_k^m(t)$ is the trajectory modified by the timing t_k^m, $b_k^m(t)$ is the base line connecting an initial point with an end point of $\tilde{y}_k^m(t)$, and t_{st}^m is the primitive timing parameter of the behaviour coordinator m that represents when it gets the motion started. These modifications are shown in Figure 11. If multiple behaviour coordinators indicate coefficients to one behaviour controller, multiplication of these values becomes a new one as follows:

$$r_k = \prod_{i \in I} r_i \tag{4}$$

where I is a group that indicates coefficients to the behaviour controller k. Finally, the feedback controller on the bottom layer of the hierarchical behaviour controller makes the corresponding actuator follow the desired trajectory.

Learning algorithm for behaviour coordinator

Utilisation of the hierarchical behaviour controller makes the controller designing process easier and more comprehensive. However, in this approach there is a critical issue of how to adjust the behaviour coordinator in case the objective behaviour or robot parameters are changed. The Newton-Raphson method is used to adjust the behaviour coordinator for these issues. This method measures the effects of the local behaviour controllers on the global behaviour, and changes the coefficients for them by trials.

The global behaviour could be adjustable to some extent only by adjusting the activation coefficients. The relation between the change of the activation coefficient and the resultant behaviour is strongly nonlinear. However, we assume that the relations could be expressed as the multiplication of the degree of contributions and activation coefficients only in limited neighbourhood of the current state. Therefore, the performance vector p at step s, which has elements as indices of the evaluation for the total behaviour, is written by:

$$p(s) = \mathbf{W}(s) \cdot r(s) \tag{5}$$

where $\mathbf{W}$ is a gradient matrix, and r is an activation vector.

The error vector e from the desired performance p^* and the current performance vector p are written by:

$$\begin{aligned} e(s) &= p^* - p(s) \\ &= \mathbf{W}(s) \cdot r^*(s) - \mathbf{W}(s) \cdot r(s) \\ &= \mathbf{W}(s) \cdot (r^*(s) - r(s)) \end{aligned} \tag{6}$$

This calculated activation coefficient r^* is not the desired one, because a linearised equation is adopted for the nonlinear system. Therefore, the target activation vectors are searched iteratively by using the Newton-Raphson method. The procedure is as follows:

Step 1: Evaluate total behaviour performance $p(s)$ by means of tuning with the activation vector $r(s)$.

Step 2: Explore the performance $p'(s)$, $p''(s)$ and $p^{n-1}(s)$ around neighbourhood areas, $r'(s)$, $r''(s)$ and $r^{n-1}(s)$. These $r(s)$ are linearly independent.

Step 3: Update gradient matrix $\mathbf{W}(s)$ using equation (9), and calculate new activation coefficients using equation (10) and equation (11).

$$R(s) = \left(r(s), r'(s), r''(s), \dots, r^{n-1}(s)\right) \tag{7}$$

$$P(s) = \left(p(s), p'(s), p''(s), \dots, p^{n-1}(s)\right) \tag{8}$$

$$\mathbf{W}(s) = P(s) \cdot \mathbf{R}^{-1}(s) \tag{9}$$

$$\Delta r(s) = \mathbf{W}^{-1}(s) \cdot e(s) \tag{10}$$

$$r(s+1) = r(s) + \Delta r(s) \tag{11}$$

Step 4: Evaluate performance vector $p(s+1)$ at next step s with new activation vector $r(s)$. If the behaviour is insufficient, go back to Step 2.

Experiments

In this section, the learning algorithm explained above is applied to the control problem of three-dimensional brachiation using Gorilla Robot II.

Generally, apes do 'slow brachiation', moving at almost the same speed as a human walks. However, when excited or frightened, apes can plunge through the forest canopy at astonishing speeds, sometimes covering 30 feet ($\sim$9.5 m)
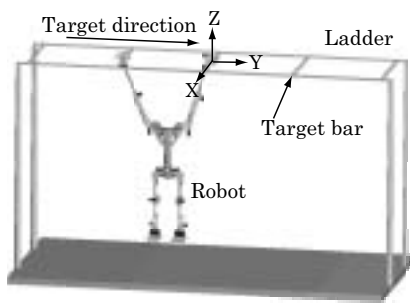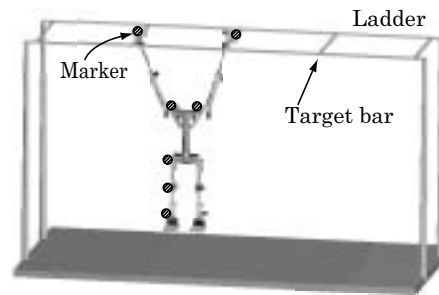
Figure 12 Coordinate system.

Figure 13 Measuring points.

or more in a single jump without a break in 'stride' ('fast brachiation', ricocheting) (Eimerl and DeVore 1966). In this section, we adapt the proposed learning algorithm to two types of slow brachiation: over-hand brachiation and side-hand brachiation. Figure 12 shows the coordinate system.

Motion measurement using real-time tracking system

A vision sensor is very useful to measure a dynamical motion without constant constraints, because the constrained points are switched in accordance with the body posture. During brachiation, it is almost impossible to measure the body position, eg the tip of the free arm or the centre of gravity of the robot, because the slip angle at the catching grip is not directly measurable using a potentiometer or rotary encoder. We therefore use the real-time tracking system Quick MAG System IV, which measures the three-

dimensional locations of the eight points at 60 Hz sampling frequency, using two CCD cameras and coloured markers.

The seven measuring positions shown in Figure 13 are chosen to approximately calculate the centre of gravity of the robot based on the following assumptions: (1) elbow of the grasping arm is kept straight; (2) both legs are controlled synchronously; and (3) two joints on the shoulder are adjoining and attached at almost the same position.

Experiment setting

The adaptation algorithm was applied to adjust loco-motion, which indicates six activation coefficients $\left(r_3^2, r_4^2, r_5^2, r_6^2, t_3^2, t_4^2\right)$ to the corresponding four local behaviour controllers: leg stretch, body rotation 2, body lift and arm reaching. To apply the adaptation algorithm to these six activation coefficients, six performance indices were chosen to represent performance of the total behaviour as follows:
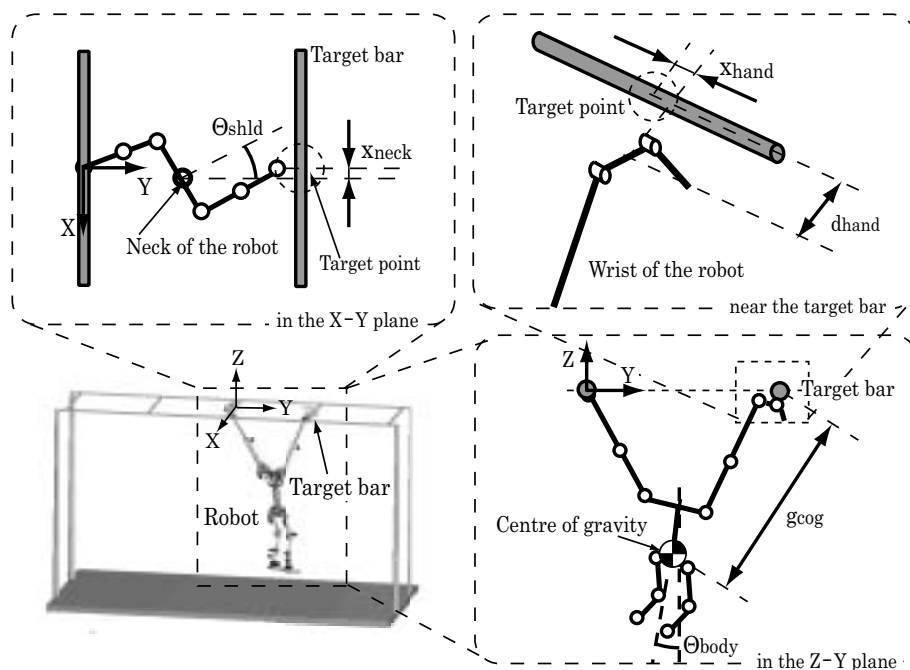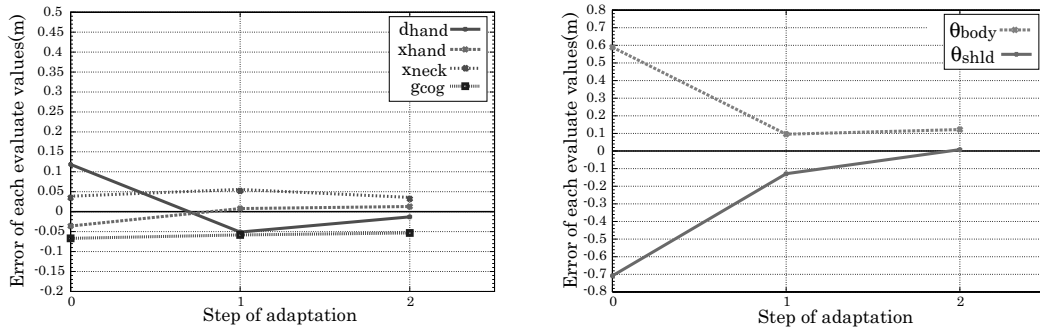

Figure 14 Elements of the performance vector p(s).

Figure 15 Transitions of evaluation values (over-hand brachiation).

- minimum distance d_{hand} in the yz plane;
- minimum distance x_{hand} in the x-axis between a free hand and a target point;
- minimum distance g_{cog} between centre of gravity and a target branch;
- tilt angle θ_{body} from vertical direction at a grasping moment;
- rotating angle θ_{shld} from moving direction at the grasping moment;
- deviated position of the neck x_{neck} in the x direction from a target point.

In this case, the performance vector $\boldsymbol{p}$(s) and the activation coefficients $\boldsymbol{r}$(s) at step s are therefore expressed by:

$$\boldsymbol{p}(s) = \left(d_{hand}(s), x_{hand}(s), g_{cog}(s), \theta_{body}(s), \theta_{shld}(s), \right.$$
$$\left. x_{neck}(s) \right)^{\mathrm{T}} \qquad (12)$$

$$\boldsymbol{r}(s) = \left(r_3^2, r_4^2, r_5^2, r_6^2, t_3^2, t_4^2 \right) \qquad (13)$$

To sustain linear independence, small perturbations α were added to the activation coefficients $\boldsymbol{r}'$(s), $\boldsymbol{r}''$(s) and $\boldsymbol{r}^{n-1}$(s) as follows:

$$\boldsymbol{r}'(s) = \left(r_3^2 + \alpha, r_4^2, r_5^2, r_6^2, t_3^2, t_4^2 \right) \qquad (14)$$

$$\boldsymbol{r}''(s) = \left(r_3^2, r_4^2 + \alpha, r_5^2, r_6^2, t_3^2, t_4^2 \right) \qquad (15)$$

$$\boldsymbol{r}'''(s) = \left(r_3^2, r_4^2, r_5^2 + \alpha, r_6^2, t_3^2, t_4^2 \right) \qquad (16)$$

$$\boldsymbol{r}''''(s) = \left(r_3^2, r_4^2, r_5^2, r_6^2 + \alpha, t_3^2, t_4^2 \right) \qquad (17)$$

$$\boldsymbol{r}'''''(s) = \left(r_3^2, r_4^2, r_5^2, r_6^2, t_3^2 + \alpha, t_4^2 \right) \qquad (18)$$

For elements of the performance vector, see Figure 14.

Over-hand brachiation

Six local behaviours for preliminary swing and locomotion were initially designed. The behaviour coordinator for preliminary swing is intuitively designed based on the parametric excitation. The learning algorithm was applied
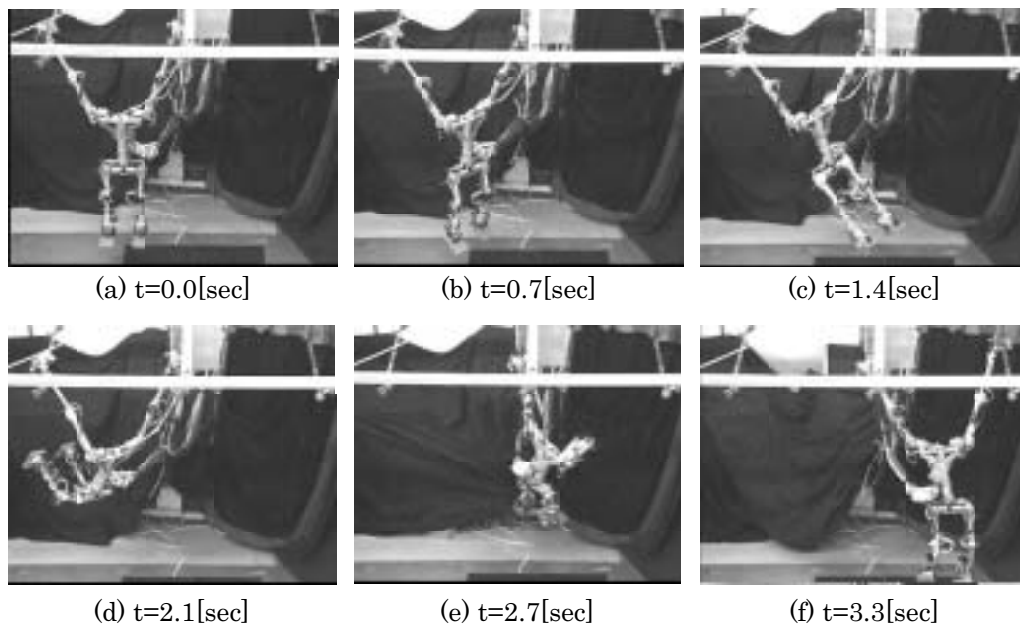

(a) t=0.0[sec]   (b) t=0.7[sec]   (c) t=1.4[sec]

(d) t=2.1[sec]   (e) t=2.7[sec]   (f) t=3.3[sec]

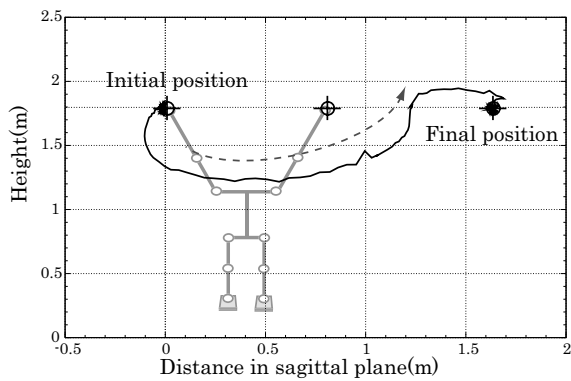**Figure 16** Stroboscopic photography of over-hand brachiation.

Figure 17 Trajectory of free hand during over-hand brachiation.

to obtain the behaviour coordinator for locomotion. As an initial setting, the behaviour coordinator outputs one as an activation level to each local behaviour. In this experiment, $\alpha = 0.05$. In each step, the robot makes six trials with different activation coefficients. Through the learning process, each performance error goes to zero as shown in Figure 15, which shows transitions of each element in error vector e. Over-hand brachiation is achieved at the second step. In each step the robot makes six trials with different activation coefficients for local search. Figure 16 shows the achieved over-hand brachiation, and Figure 17 shows a trajectory of the free hand. The locomotion time from one branch to another is 1.20 s.
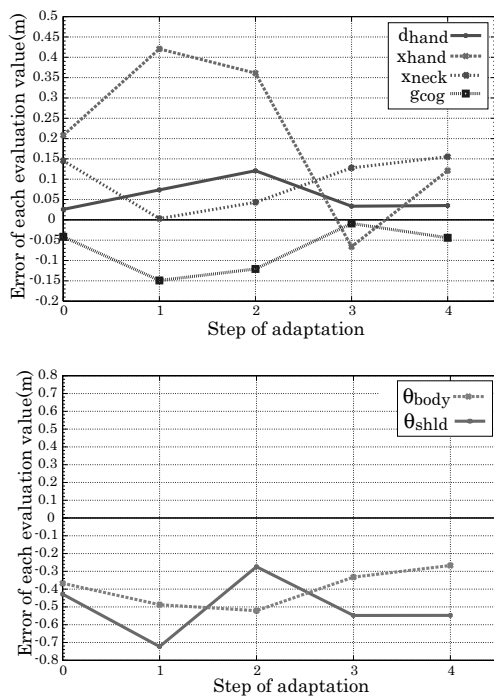
Side-hand brachiation

To achieve side-hand brachiation, two local behaviours, body rotation 2 and arm reaching, were redesigned. The learning algorithm was then applied to obtain the behaviour coordinator for locomotion as well as over-hand brachiation. Through four trials, the minimum distance x_{hand} and θ_{body} is especially decreasing. At the fourth step, the robot grasped the target branch. The target motion is generated in four steps by the same learning algorithm. Transitions of each element in error vector e and a trajectory of the free hand are shown in Figures 18 and 19, respectively. Figure 20 shows the achieved side-hand brachiation. It takes a little longer, 1.53 s, for this locomotion.

Continuous locomotion

Now we consider achievement of continuous locomotion without suspending. Continuous locomotion generally consists of a periodic repetition of one cycle motion, although there is a difference between the start state of preliminary swing and secondary swing with regard to brachiation: with the former, the legs are just under the body; but with the latter, the legs are in front of the body, because they are swung up during the first locomotion. Taking this difference into consideration, it was decided that the latter half of preliminary swing be used as secondary swing. A controller was made that can periodically repeat side-hand brachiation. Gorilla Robot II could achieve dynamic continuous locomotion moving twice without suspending. Figure 21 shows the continuous locomotion achieved.
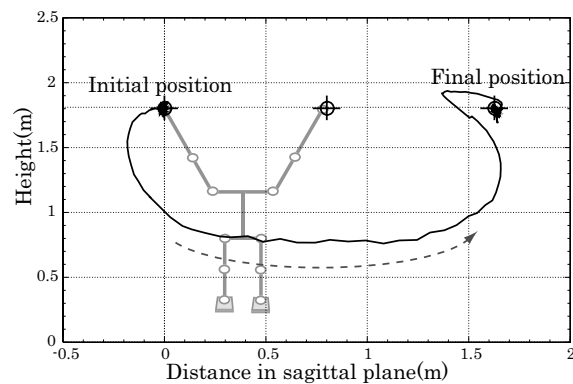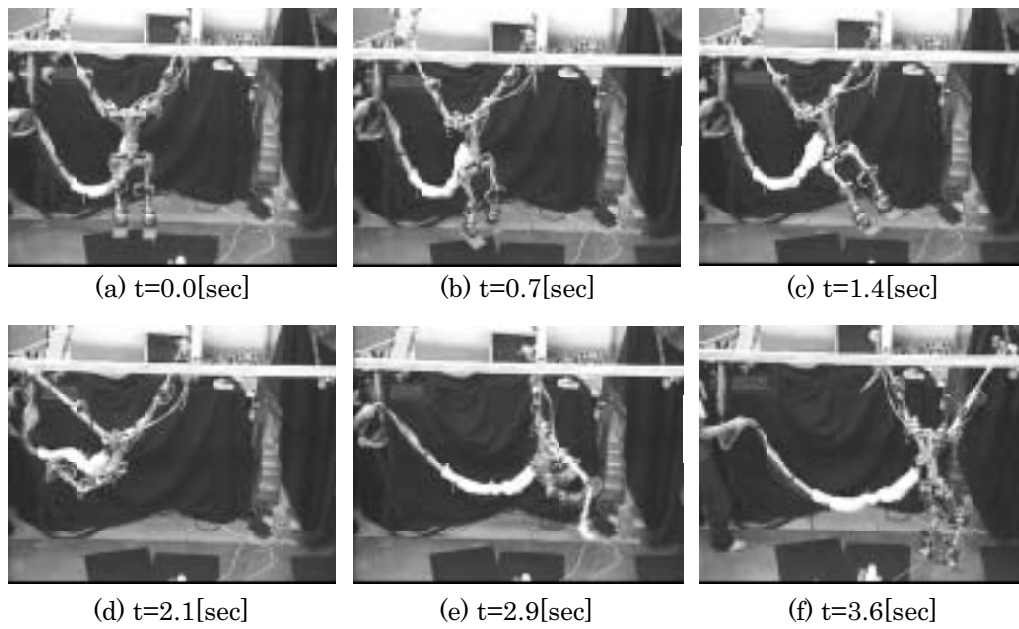

Figure 18 Transitions of evaluation values (side-hand brachiation).

Figure 19 Trajectory of free hand during side-hand brachiation.

(a) t=0.0[sec]   (b) t=0.7[sec]   (c) t=1.4[sec]

(d) t=2.1[sec]   (e) t=2.9[sec]   (f) t=3.6[sec]

Figure 20 Stroboscopic photography of side-hand brachiation.

(a) t=0.0[sec]   (b) t=0.8[sec]   (c) t=1.7[sec]

(d) t=2.9[sec]   (e) t=3.6[sec]   (f) t=4.5[sec]

(g) t=6.8[sec]   (h) t=7.2[sec]   (i) t=7.5[sec]
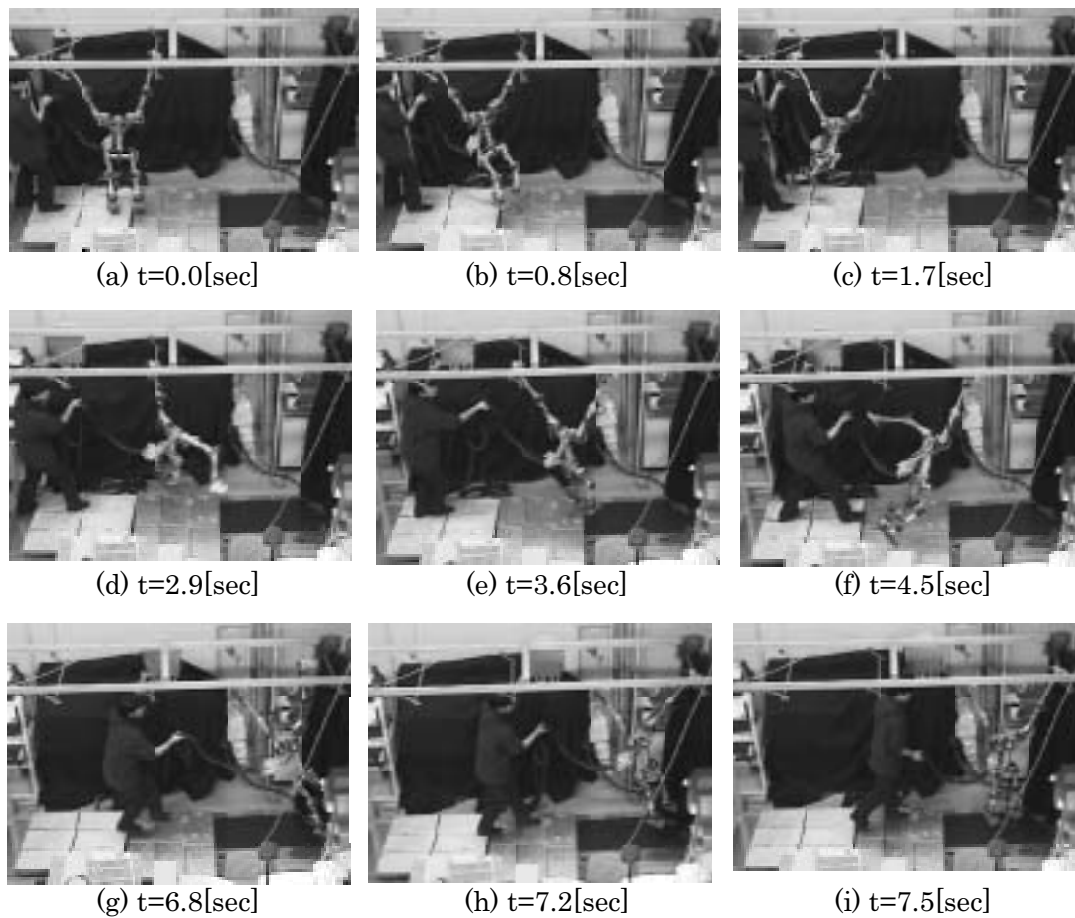
Figure 21 Stroboscopic photography of continuous locomotion.

Conclusions

An unsupervised learning algorithm such as reinforcement learning has generally poor ability to find optimal solutions; therefore, task decomposition or hierarchical structure are required to obtain a controller for a robot with multiple degrees of freedom. In this paper a solution for a robot to achieve such dexterous behaviour was explained.

The concept of a multi-locomotion robot that can select appropriate locomotion types was proposed, and Gorilla Robot II was developed as a prototype. We focused on brachiation in several locomotion types. We showed the algorithm for generation of dexterous behaviour, which includes the architecture for behaviour coordination and adjustment. The learning algorithm was improved so that the timing of each behaviour as well as their activation levels could be properly adjusted. The learning algorithm was applied to achieve two types of brachiation, over-hand and side-hand, by Gorilla Robot II. Finally, a controller was made that can periodically repeat side-hand brachiation; Gorilla Robot II was able to achieve continuous locomotion moving twice without suspending.

References

Arai H, Tanie K, Shiroma N. 1998. Time-scaling control of an underactuated manipulator. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98). 1998 May 16–20; Leuven, Belgium. Monterey, CA: IEEE Pr. p 2619–26.

Eimerl S, DeVore I. 1966. The primates. New York: Time-Life Books.

Fukuda T, Hosokai H, Kondo Y. 1991. Brachiation type of mobile robot. In: Proceedings of the IEEE International Conference on Advanced Robotics. 1991 Jun 20–22; Pisa, Italy. Monterey, CA: IEEE Pr. p 915–20.

Fukuda T, Saito F, Arai F. 1991. A study on the brachiation type of mobile robot (heuristic creation of driving input and control using CMAC). In: Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems. 1991 Nov 3–5; Osaka, Japan. Monterey, CA: IEEE Pr. p 478–83.

Hasegawa Y, Fukuda T, Shimojima K. 1999. Self-scaling reinforcement learning for fuzzy logic controller – applications to motion control of two-link brachiation robot. *IEEE Trans Ind Electron*, 46:1123–31.

Hasegawa Y, Ito Y, Fukuda T. 2000. Behavior coordination and its modification on brachiation-type mobile robot. In: Proceedings of the IEEE International Conference on Robotics and Automation. 2000 Apr 24–28; San Francisco, CA. Monterey, CA: IEEE Pr. p 3984–9.

Huang Q, Kajita S, Koyachi X et al. 1999. A high stability, smooth walking pattern for a biped robot. In: Proceedings of the IEEE International Conference on Robotics and Automation. 1999 May 10–15; Detroit, MI. Tampa, FL: IEEE Robotics and Automation Society. p 65–71.

Kajima H, Hasegawa Y, Fukuda T. 2002. Study on brachiation controller – adjustment method of strength and timing parameters. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems. 2002 Sep 30–Oct 4; Switzerland. Monterey, CA: IEEE Pr. p 2497–502.

Kajita S, Tani K. 1996. Experimental study of biped dynamic walking. *IEEE Control Syst*, 16:13–19.

Kanehira N, Kawasaki T, Ohta S et al. 2002. Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems. 2002 Sep 30–Oct 4; Switzerland. Monterey, CA: IEEE Pr. p 2455–60.

Kuffner JJ, Nishiwaki K, Kagami S et al. 2001. Motion planning for humanoid robots under obstacle and dynamic balance constraints. In: Proceedings of the IEEE International Conference on Robotics and Automation. 2001 May 21–26; Seoul, Korea. Tampa, FL: IEEE Robotics and Automation Society. p 692–8.

Nakanishi J, Fukuda T. 2000. A leaping maneuver for a brachiating robot. In: Proceedings of the IEEE International Conference on Robotics and Automation. 2000 Apr 24–28; San Francisco, CA. Monterey, CA: IEEE Pr. p 2822–7.

Nakanishi J, Fukuda T, Koditschek DE. 1998. Experimental implementation of a 'target dynamics' controller on a two-link brachiating robot. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98). 1998 May 16–20; Leuven, Belgium. Monterey, CA: IEEE Pr. p 787–92.

Nakanishi J, Fukuda T, Koditschek DE. 2000. A brachiation robot controller. *IEEE Trans Robotics Automation*, 16:109–23.

Paul RP. 1981. Robot manipulators: mathematics, programming and control. Cambridge, MA: MIT Pr. p 157.

Saito F, Fukuda T, Arai F. 1993. Swing and locomotion control for two-link brachiation robot. In: Proceedings of the IEEE International Conference on Robotics and Automation. 1993 May 2–7; Atlanta, GA. Tampa, FL: IEEE Robotics and Automation Society. p 719–24.

Saito F, Fukuda T, Arai F. 1994. Swing and locomotion control for a two-link brachiation robot. *IEEE Control Syst*, 14:5–12.

Simons EL. 1972. Primate evolution: an introduction to man's place in nature. New York: Macmillan.

Spong M. 1995. The swing up control problem for the acrobot. *IEEE Control Syst*, 15:49–55.

Yamafuji K, Fukushima D, Maekawa K. 1992. Study of a mobile robot which can shift from one horizontal bar to another using vibratory excitation. *JSME Int J Ser III*, 35:456–61.