Artif Life Robotics (2002) 6:44–51

ORIGINAL ARTICLE

Tadashi Odashima · Zhiwei Luo · Shigeyuki Hosoe

Hierarchical control structure of a multilegged robot for environmental adaptive locomotion

Received and accepted: September 27, 2002

Abstract We propose a biomimetic, two-layered, hierarchical control structure for adaptive locomotion of a hexapod robot. In this structure, the lower layer consists of six uniform subsystems. Each subsystem interacts locally with its neighboring subsystems, and autonomously controls its own leg movements according to the weighted sum of three basic vector fields that represent the three basic motion patterns of the robot body. The upper-layer controller decides the intended body movement, and sends the lower-layer controllers three variables as the weights of each basic vector field. This approach greatly reduces the communication between the two layers, and contributes to real-time adaptive locomotion. 3D dynamic simulations, as well as experiments with a real modularized hexapod robot, show the effectiveness of this hierarchical structure.

Key words Multilegged robot · Hierarchical structure · Locomotion pattern · Distributed cooperation · Vector fields

1 Introduction

The ultimate purpose of our work is to build a multilegged robot that can achieve real-time autonomous adaptive locomotion in a complex environment. For a complex system with many degrees of freedom, such as a multilegged robot, the computational load for its control tends to increase, which poses a serious problem in realizing real-time adaptation to the environment. Considering that an autonomous robot not only has to control many actuators, but also has to collect and process enormous quantities of spatial–temporal environmental information and then plan its motion strategy, it is important to reduce the computational load as much as possible. In this article, a distributed system in which the entire system is controlled through cooperation between several processors is discussed as one method for reducing the control load on each processor. Such cooperation through local interactions between processors that handle only local information allows globally balanced behavior from the system. However, it is difficult for a distributed structure to rapidly plan the behavior of the entire system based on each subcontroller, because the communication time for local interactions tends to increase. In order to achieve real-time adaptive locomotion, it is desirable for the robot to possess the benefits of both distributed control and centralized control.

Nature provides us with clues on how to realize such a structure. For example, animals have quite elaborate systems to control their multiple degrees of freedom in motions. They appear to be very skillful at performing environmentally adaptive movements. How do they manage to overcome such knotty problems? One possible answer to this question is as follows. Animals may solve the problem by using the locomotion control structure shown schematically in Fig. 1, which is a control system that can be regarded as a hierarchical structure consisting of two layers,[1] i.e., the upper brain layer and the lower central pattern generator (CPG) layer. The brain layer performs a global motion plan for the body, while the CPG layer realizes the coordinated control of every leg. The assumption that the CPG consists of several nervous oscillators[2] is widely accepted. The oscillators interact with each other locally, and regulate the periodic movement of the legs. In this sense, the CPG layer can be regarded as a distributed control system consisting of the subsystems enclosed by the dotted lines in Fig. 1.

Gister et al.[3] and Mussa-Ivaldi and Gister[4] investigated the organization of the motor output using spinally dissected frogs. They stimulated the spinal cord with microelectrodes, and observed the isometric forces produced by

T. Odashima (✉) · Z. Luo · S. Hosoe
Biomimetic Control Research Center, Riken, Nagoya Anagahora, Shimoshidami, Moriyama-ku, Nagoya 463-0003, Japan
Tel. +81-52-736-5870; Fax +81-52-736-5871
e-mail: odashima@bmc.riken.go.jp

S. Hosoe
Graduate School of Engineering, Nagoya University, Nagoya, Japan

This work was presented, in part, at the Sixth International Symposium on Artificial Life and Robotics, Tokyo, Japan, January 15–17, 2001

the muscles of the legs. M ıssa-Ivaldi and Gister[4] reported that when elicited by a sir gle stimulation, the ankle position converged to a single e quilibrium point of a force vector field, and the point of convergence was shifted by superimposing several vector fields resulting from multiple stimulations. These studies show that complex movements can be achieved by the sum of several basic movements. This means that the upper layer can describe a complex movement with a small number of parameters, such as the weight of the basic vector fields.

Inspired by such studies on motion control in animals, we have applied a hierarchical control structure to the control of a multilegged robot. This robot consists of two control layers. The upper-layer controller plans the body movements and broadcasts three variables only to the lower-layer controllers. In the lower layer, six homogeneous controllers construct a distributed system such as the CPG. They receive the variables from the upper layer and control the leg movements. We focus on the highly effective coding method between the two layers in order to translate a designated trunk movement smoothly into individual leg movements.

This article is organized as follows. In Sect. 2, the hierarchical control structure as well as the functions of the two layers are described. A leg movement is characterized by the swing timing between the legs and a trajectory. Section 3 considers how to coordinate the swing timing. A cording method to calculate the trajectory of each leg is presented in Sect. 4. We evaluate our approach with 3D dynamic simulations, and in experiments with an actual multilegged robot named "Caterpillar" in Sect. 5. Finally, some conclusions are given in Sect. 6.

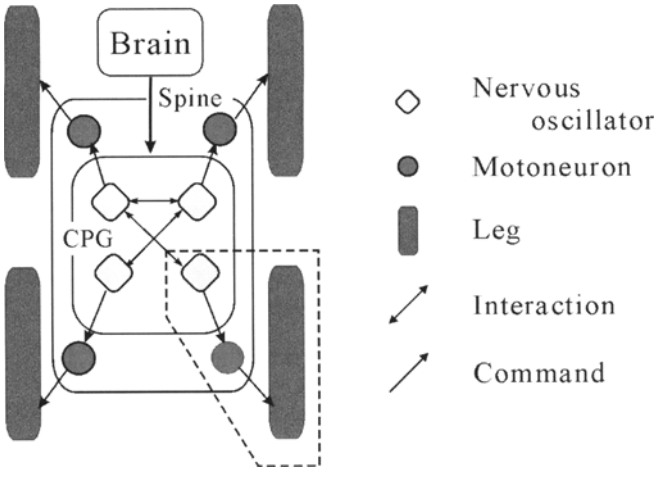

Fig. 1. Hierarchical control structure in animals

2 Synthesis of a hierarchical control structure

The hierarchical control structure for Caterpillar is shown in Fig. 2. Caterpillar consists of two control layers, each playing a different role. The upper layer is a centralized control part which plans and decides the intended body movement in rapid response to environmental changes. Meanwhile, the lower layer works as a distributed system consisting of uniform subsystems. The whole robot movement is realized by the autonomous and cooperative behavior of each subsystem. The details of both layers are presented below.

2.1 Upper layer

The upper layer consists of a processor, the external sensors, and communication channels, and plays the role of a brain. In order to achieve real-time adaptive movement, it is reasonable to set a central controller. The role of this layer is to make plans for the robot body based on environmental information. The plans are translated to three variables, which are sent to all the subcontrollers in the lower layer. It is important to note that the upper-layer controller broadcasts the same set of variables to all lower controllers for each specific robot movement. This is the key to reducing the load in leg control. The details of this coding method are described in Sect. 4.

2.2 Lower layer

The lower layer works as a CPG. It consists of six uniform subsystems. Each subsystem has one leg with three degrees

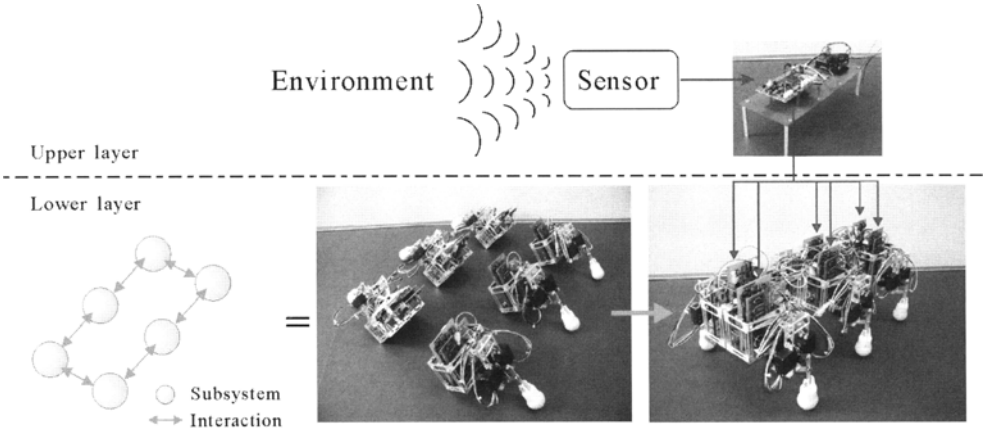

Fig. 2. Hierarchical control structure of the six-legged robot "Caterpillar"

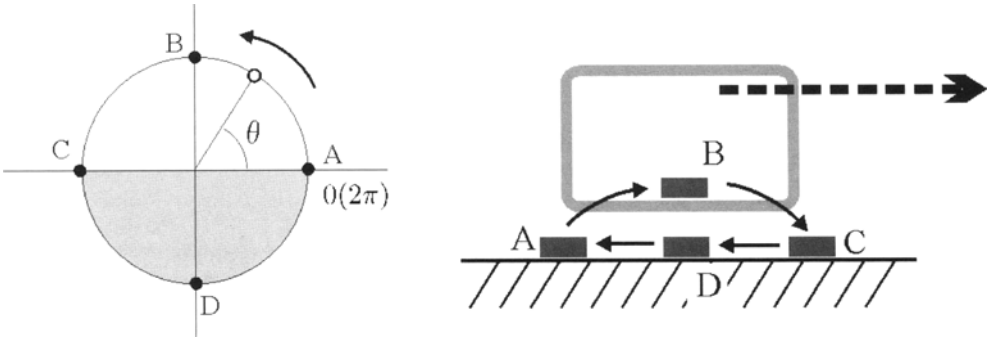

Fig. 3. Time series of the relation between leg state and phase of oscillator

of freedom (d.o.f.), one processor, three touch sensors, three communication channels, and one toggle switch. These are arranged in two lines along the rostral–caudal line (see Fig. 4, right). By recognizing this arrangement, all subsystems can process using the same program even though they are set in different positions. If one subsystem is connected to another, a touch sensor is turned on and the leg detects its neighbor. Switching a toggle switch tells a subsystem on which side (left or right) it is positioned. This information allows the subsystems to recognize their location within the robot body, as well as the direction of the robot's head. Two of the communication channels are connected to neighbors, and the other is corrected to the upper-layer controller. The former are arranged as shown in the bottom left-hand panel of Fig. 2, and send data bidirectionally. Through interaction via these channels, all the lower-layer controllers decide their own swing timing, which means that this interaction guarantees a continuous stable gait pattern. The details of the gait-pattern generations are described in Sect. 3, and more details of the subsystem's structure can be found in in the literature.[5] The third communication channel transfers a command from the upper layer to the lower layer. Currently, the lower-layer controllers use this channel for reception only. Based on this information, each lower-layer processor determines an adequate trajectory for its leg, the detailed algorithm of which is given in Sect. 4.

Several studies have considered the application of a distributed control structure for generating the gait patterns of multilegged robot.[6–8] These studies focus on the distribution in the algorithm and/or the software level. However, there is no study that tried to expand the distribution to the hardware level. In our control structure, the subsystems remain highly independent from each other, not only on the software level, but also on the hardware level. This structure brings three advantages from the engineering point of view. First, since one controller controls only one leg, a system covered by one controller is quite small, which means that the program code development is simple. Second, the structure of each subsystem is also not complex, and therefore the robustness of the hardware is increased. The last advantage is the ease of repairs, i.e., by replacing the defective subsystem with a new one, the whole robot system can be recovered rapidly.

3 Leg-phase control by interaction between oscillators

During walking, since leg motion is a periodic movement, the state of a leg can be associated with the phase of a periodic oscillator. Figure 3 represents the relationship between the foot position and the oscillator phase. In this figure, the left-hand and right-hand panels show the periodic oscillator and the side view of one subsystem, respectively. In the left-hand panel, the small circle indicates the phase of the oscillator, which is represented by θ. The four points marked A, B, C, and D on the circle are the points corresponding to the four small squares representing the foot positions in the right-hand panel. The dotted arrow in the right-hand panel indicates the rostral direction. The actual positions of A, B, and C are changed according to the desired trajectory, and that of D is fixed relative to the coxa, i.e., the joint between the body and the leg. The lower-layer processors interact locally with each other and adjust their own oscillator phase according to phase differences between neighbors based on the following gradient dynamics:

$$\frac{d\theta_i}{dt} = \omega - \sum_p \frac{\partial W(\psi_{ip})}{\partial \psi_{ip}} \tag{1}$$

where θ_i is the phase of the oscillator in the i-th subsystem, ω is the constant angular velocity, and $W(\psi_{ip})$ is the potential function defined as

$$W(\psi_{ip}) = -h\exp\left[-T\left\{1 - \cos(\psi_{ip} - \psi_{ipd})\right\}\right] \tag{2}$$

where T and h are constant values. ψ_{ip} is the phase difference between the i-th subsystem and its neighbor. Subscript d indicates the direction where neighbors exist (forward, backward, or side), and ψ_{ipd} is the target value of ψ_{ip}. In a gradient system in which the potential is given by Eq. 2, ψ_{ip} converges to ψ_{ipd}.[9] Therefore, the phase of each oscillator

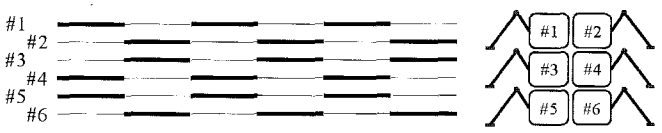

Fig. 4. Gait diagram for tripod gait

and the state of each leg are determined relative to each other. After convergence, the second term on the right-hand side of Eq. 1 is 0, and θ_i moves with a constant angular velocity. This means that if the reference clocks of all subsystems remain synchronized with each other, communication between the lower controllers is not needed after convergence. Under these conditions, a reduction in communication costs can be expected.

Any gait can be described by ψ_{ipd}. Variations in walking speed within a small range are realized by a change in stance length, i.e., leg trajectory. However, drastic changes cannot happen without a gait shift. It is easy for the proposed structure to introduce a gait shift. In previous work,[5,10] other gait types have been generated by controlling ψ_{ipd} and ω. Since we focus on how to generate leg trajectories in order to realize arbitrary locomotion of a robot body, we work with a tripod gait only. This gait allows the hexapod to progress with the fastest possible walking speed. Figure 4 shows a gait diagram of a tripod gait. This represents the state of each leg as a time series. The bold and fine lines represent the stance and swing phases, respectively. It is clear from this figure that each leg is in antiphase against its neighbors in this gait, and setting all the ψ_{ipd} to π guarantees this gait. ω is set to a constant value to maintain the gait, and in practice this value is decided from the values derived from the actuators used for the robot legs.

4 Trajectory control based on a vector field

The interaction described in Sect. 3 can generate a steady gait pattern which prevents the robot falling down. However, this gait is not sufficient to drive the robot along arbitrary paths. In this section, a gait cording method to help each lower-layer controller to estimate its leg trajectory is discussed. For simplicity, the movement of the robot is restricted to a flat plane. Under these condition, any robot movement can be described by two basic motions: translation and rotation. In a translation, all the points on the robot body progress in the same direction for the same distance (Fig. 5). In Fig. 5, a rectangle and six large circles indicate the robot body and the coxae of the legs, respectively, while the small circles indicate the center of the robot body. It is also possible to decompose the translation movement into two elements, i.e., in the rostral and the lateral directions. Each motion element is represented by one vector field, $\boldsymbol{A}$ or $\boldsymbol{B}$, in Fig. 6. These fields are

$$\boldsymbol{A} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{B} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3}$$

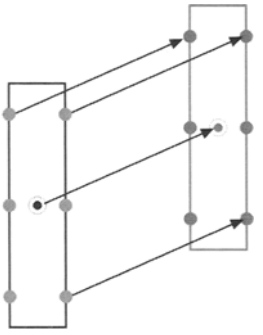

Fig. 5. Translational movement

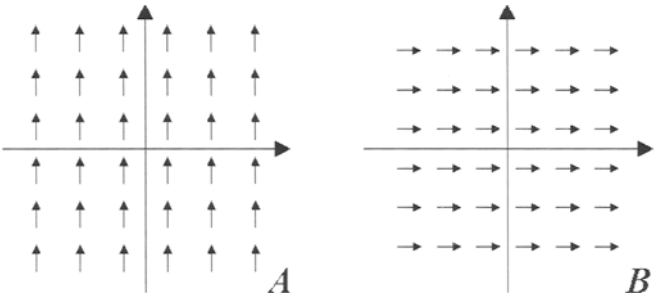

Fig. 6. The two basic vector fields for translation

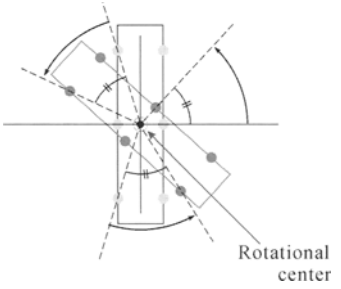

Fig. 7. Rotational movement

A coordinate system for these fields is fixed on the robot body, whose vertical and horizontal axes correspond to the rostral axis (caudal to rostral) and the lateral axis (medial to lateral), respectively. The positive directions are the rostral and right-hand side directions, and the origin is set at the center of the robot body. A vector in this field indicates the desired direction at the corresponding point, and the desired distance is proportional to its norm.

The second basic motion is rotation. In rotational movements, all the points on the robot rotate by about the same angle around the rotational center, and the distance moved is proportional to the distance from the rotational center (Fig. 7). A rotational movement is also represented by one vector field, as shown in Fig. 8. Its coordinate system is the same as in the case of translation. This field is

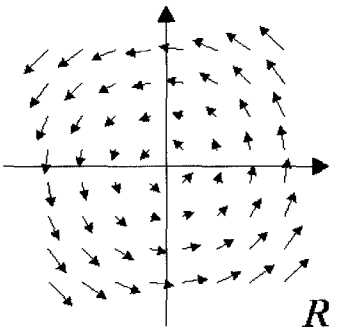

Fig. 8. Basic vector field for rotation

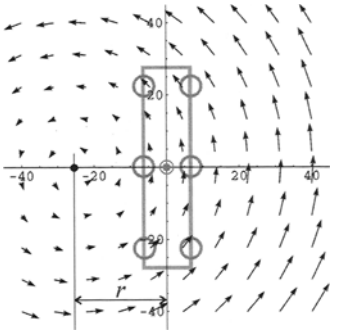

Fig. 9. Vector field generated, resulting in a gentle curving motion

$$\boldsymbol{R} = \begin{pmatrix} -y \\ x \end{pmatrix} \tag{4}$$

where x and y are horizontal and vertical variables, respectively. One can obtain any movement from the weighted sum of these three basic vector fields, as follows:

$$\boldsymbol{V} = a_1\boldsymbol{A} + a_2\boldsymbol{B} + a_3\boldsymbol{R} \tag{5}$$

If $a_3 \neq 0$, the position of the rotational center $(X_C Y_C)$ is given by

$$X_C = -\frac{a_1}{a_3}, \qquad Y_C = \frac{a_2}{a_3} \tag{6}$$

For example, a given parameter set $(a_1,a_2,a_3) = (2.0, 0.0, 0.08)$ produces a vector field such as the one shown in Fig. 9. A set of vectors is generated around a rotational center, which is indicated by a black dot on the horizontal axis. This field represents a gentle curving motion, as in Fig. 10. Since each subsystem knows the position of its coxa in the vector field, the leg trajectory for moving the coxa can be calculated based on the vector at the point corresponding to the position of the coxa in the vector field.

The sequence of information and command from the upper layer to lower-layer subsystems is concluded as follows. Based on the environmental constraints, the upper-layer processor generates a vector field $\boldsymbol{V}$ for the desired movement, and sends weights (a_1,a_2,a_3) to all the lower-layer processors. Each lower-layer processor reconstructs $\boldsymbol{V}$ and refers to a vector, $v \in \boldsymbol{V}$, at a position corresponding to its coxa. Each leg trajectory is derived from v by indicating the desired coxa trajectory.

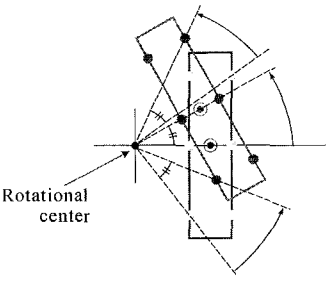

Fig. 10. Gentle curving motion generated by the vector field in Fig. 9

There are four points which characterize the leg trajectories. These points correspond to the phase of the oscillator (see Fig. 3). Point D is a home position of one leg, and the tip of the leg reaches at this specific point when the phase of the oscillator becomes $3\pi/2$. When the phase of the oscillator becomes 0, the desired trajectory in this step is calculated. At first, two specific points in the trajectory are decided: the posterior extreme position (PEP) and the anterior extreme position (AEP). The PEP is a point where the state of the leg changes from the stance phase to the swing phase (in Fig. 3, this is point A). Meanwhile, the AEP is a point which is changing from the swing phase to the stance phase (Fig. 3 point C). Note that the AEP in a step is identical to the PEP in the next step, since walking is a periodic motion. The AEP and PEP are derived as follows. When $a_3 = 0$, i.e., translational motion, the AEP and the PEP are set a relative to the point D. Point D is the middle point between the AEP and the PEP. The direction from the AEP to the PEP is opposite to the direction of v. The distance between the AEP and the PEP, i.e., the stride (defined as S), is calculated by the product of the norm of v and a constant value set by the size of the robot. We call the state that the tip of the leg reaches at point A (= PEP) condition A, and the other states are labeled in the same way. In translational movement, the coxa moves in the first half of S from condition C to condition D. Then in the second half of S, the coxa moves from condition D to condition A. In total, each coxa moves by S in the direction v with one step. Figure 11 illustrates the relations between the motion of a subsystem and its leg conditions in the stance phase. This is a top view of the subsystem working as the right middle leg. Figure 11a shows the body movement relative to the grounding point. On the global coordinates, points A, C, and D are at the same point in one stance phase. The dashed arrow indicates the direction of progress, and the subscripts n and $n + 1$ indicate the number of steps. From the n-th condition C to the $(n + 1)$-th condition A, the body is moved through S. Figure 11b shows the relations of the points A, C, and D to the coordinates fixed on the coxa. Point D is the middle point between points A and C. At any moment in the stance phase, the desired joint angles are set to values which will mean that the coxa moves by S in the same direction as v. In the swing phase, the lateral and caudal elements of the trajectories are set at a linearly inter-

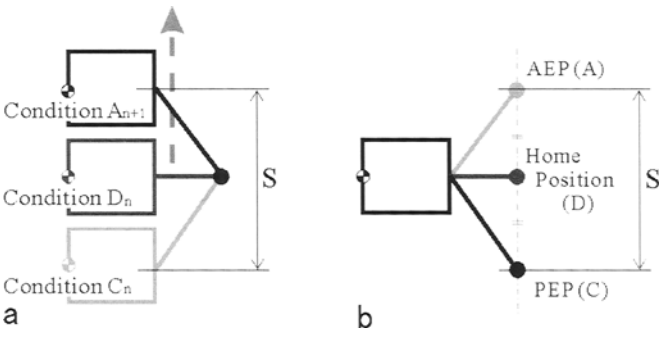

Fig. 11. The relations of leg conditions and body positions in the stance phase of translational movement. **a** The grounding position is the reference point. **b** The coxa is the reference point

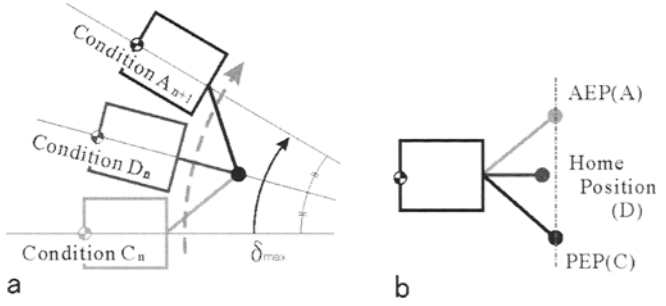

Fig. 12. The relations of leg conditions and body positions in the stance phase of rotational movement. **a** The grounding position is the reference point. **b** The coxa is the reference point

polated position between the AEP and the PEP, and the swing height is set by a function that makes the trajectory form an arch shape. When the phase of the oscillator is $\pi/2$, the tip reaches at the highest point, and the leg is in condition B. The height in condition B is constant, and this is decided by the size of the leg. On the other hand, in rotational movement, the maximum rotating angle with one step, δ_{max}, is given by

$$\begin{cases} \delta_{max} = ca_3 & (\text{if } a_a = a_2 = 0) \\ \delta_{max} = \text{sig}(a_3) \cdot \dfrac{S}{R} & (\text{if } a_1 \neq 0 \text{ and/or } a_2 \neq 0) \end{cases} \quad (7)$$

where c is a constant value, and $r = \sqrt{x_c^2 + Y_c^2}$ is the rotational radius from Eq. 6. The function sig(*) means the sign of the variable. The desired angle of each joint is set to the value that the coxa rotates around the rotational center. Note that the rotational radius of the coxa is not r, because r is the distance from the center of the robot body to the rotational center (see Fig. 9). The coxa rotates through half of δ_{max} from condition C to condition D, and then rotates through the remaining half between condition D and condition A (Fig. 12a). In rotational movement, point D is not always the middle point between A and C (Fig. 12b). The desired leg trajectories in the stance phase, which are made by each subsystem with the parameter set $(a_1,a_2,a_3) = (2.0, 0.0, 0.08)$, are compared in Fig. 13. The directions of the

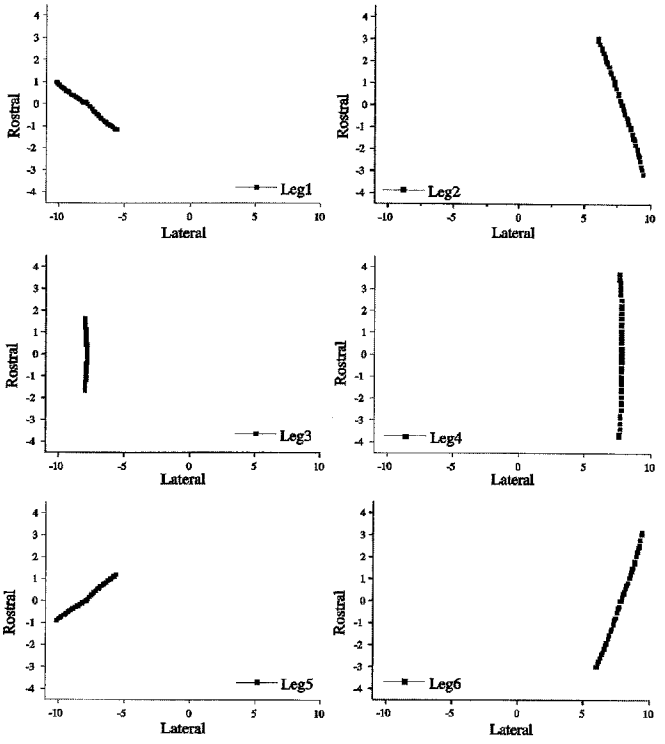

Fig. 13. Trajectory for each leg generated by the command for a gentle curve

axes are identical to the robot body coordinates. The position of each panel corresponds to the arranged position of each subsystem in the robot (see Fig. 4, right). The values in Fig. 13 mean the displacement from the coxa to the tip of the leg (cm). It is clear that each subsystem calculates a different trajectory according to its position in the robot body.

5 Simulation and experimental results

A simulator for studying dynamic gait generation has been developed using the 3D dynamic simulation library "Vortex".[11] Figure 14a shows the graphic results. In order to confirm the effectiveness of our approach, the track of the robot resulting from given parameter sets is displayed in Fig. 15. The bold line represents the track of the center of the body, and the direction of each arrow shows the posture of the body every second. The robot starts from (0,0) and makes five motions according to five parameter sets which vary with time, as shown in Table 1. The coordinates of this graph are fixed on the ground, and the unit of length is centimetres.

At the beginning of the simulation, the trunk trajectory does not follow the commanded straight line. This is caused by the transitional state between oscillators. The initial value for each oscillator takes a random value (−0.1, 0.1) (rad), and the resultant gait does not converge during the transitional period. After this period, a stable tripod gait is generated.

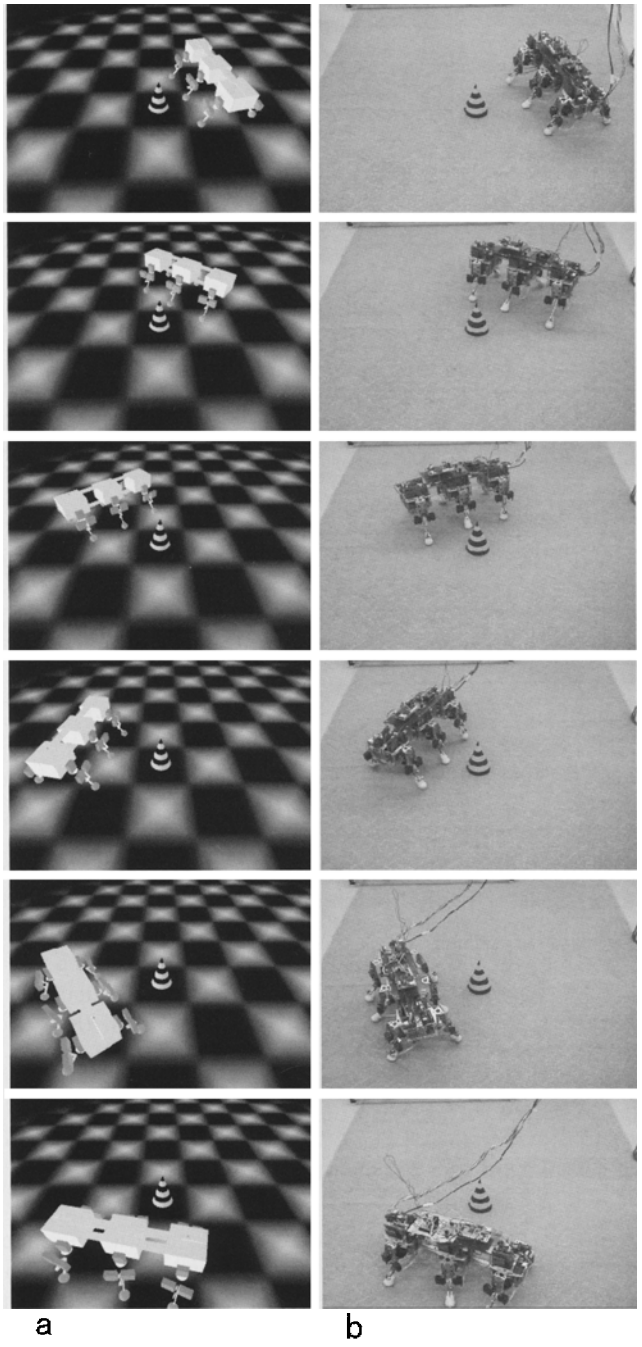

Fig. 14. 3D graphical simulation and experimental results during gentle curving motion

Table 1. Parameter sets with passing time

Time (s)	a_1	a_2	a_3
$0 \leq t \leq 5$	2.0	0.0	0.0
$5 < t \leq 15$	−1.0	1.0	0.0
$15 < t \leq 42$	2.0	0.0	−0.1
$42 < t \leq 48$	−1.5	0.0	0.0
$48 < t \leq 58$	0.0	0.0	−0.3

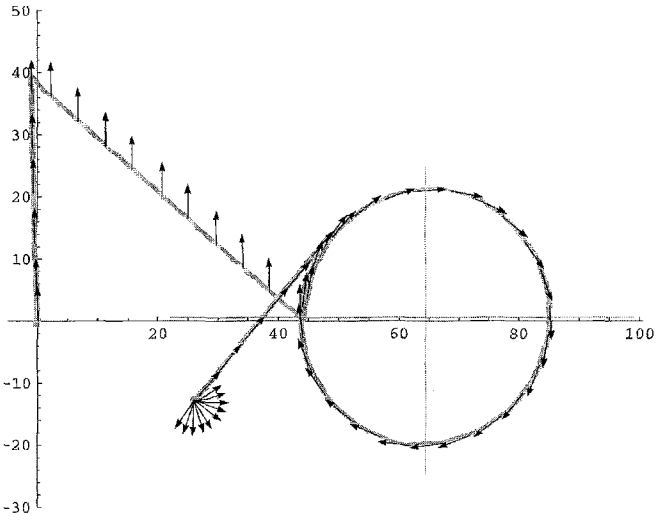

Fig. 15. Simulation results for a truck with varying commands

Table 2. Fundamental specifications of "Caterpillar"

Size (L,W,H)	57,35,35 (cm)
Weight (one module/total)	0.72/4.6 (kg)
Power supply (actuator/circuit)	External (DC8V/DC9V)
d.o.f. (one module/total)	3/18
MPU	*Hitachi* H8-539F
MPU frequency	10 (MHz)
Actuator	KO PROPO PDS-2144FET
Maximum speed	12 (cm/s)

In the third period, the rotational center is at (64.5, 0.54). At the beginning of the third period, the robot is at (43.9, 0.16), and the unit vector representing its orientation is (−0.01, 0.999). Considering that the rotational center is expressed as a value of the coordinates fixed on the robot body, it is clear that the center of the circular trajectory of the trunk matches the result derived from Eq. 6.

The proposed approach has also been implemented with Caterpillar. The fundamental specifications of Caterpillar are given in Table 2. Figure 14b shows an experimental scene which confirms that Caterpillar can walk along the correct trajectory generated by a given parameter set. This parameter set makes the robot perform a left-hand curving motion with a 25-cm radius. Figures 9 and 13 show the vector field generated and the trajectories of each leg, respectively. The results shown in Figs. 9, 13, and 14a use the same parameter set. Considering the size of the robot body, the displacement from the rotational center (indicated by a striped cone) is kept approximately within the desired length.

Figure 16 shows the second experiment to confirm the response from varying commands. Using a wireless controller, a human operator sends three weights of basic vector fields as commands. The task in this experiment is for the robot to pass through the narrow path shown in Fig. 16. Caterpillar rapidly generates its gait in response to the com-

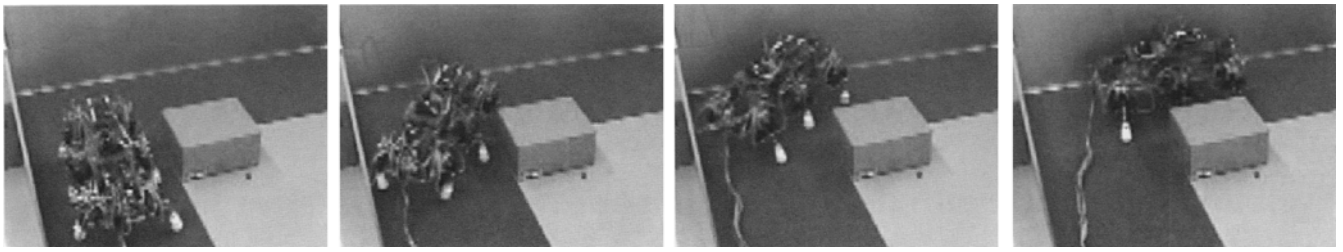

Fig. 16. An experiment to pass through a narrow pathway

mand, and smooth movements were observed. The film files of these experimental results can be found on our web site.[12]

6 Conclusions

It is well known that to control a large-scale complex system, it is better to design a decentralized control structure in order to distribute the computational load. However, if we only use a distributed structure, there will be an enormous amount of information passing between each subsystem, which makes it difficult for the system to realize a fast response to environmental changes. In other words, there is a trade-off between centralized control and distributed control.

In this article, which was inspired by the motion control structures of animals, we propose a hierarchical control structure for a multilegged robot as a solution to the above trade-off. The effectiveness of our approach was confirmed by 3D simulations and experiments with a hexapod robot. The simulation result showed that precise body trajectories are generated according to several parameter sets. The experimental results showed that a smooth adaptive movement was realized by switched commands, despite the use of slow processors. This result indicates that our hierarchical control structure succeeded in reducing the total computational cost.

We have already combined Caterpillar with other devices, and reported on gait generation by visual information received via a fixed CCD camera,[13] and voice recognition used for getting the weights of the vector field.[14] In both studies, an external computer decided the desired paths, and the upper-layer controller on Caterpillar worked only as a channel connecting the external computer and the lower-layer controllers. Strictly speaking, in these studies Caterpillar is not an autonomous robot. Currently, we are also developing a camera module and force sensors in order to achieve an autonomous adaptive locomotion robot. After considering the computational load as well as the robot's weight, we used an artificial retina chip as a camera module to allow Caterpillar to receive external information. In addition, force sensors were installed at the tip of every leg to measure the slight roughness of the terrain. The sensors attached to the lower subsystems will play a particularly important role in our future work. We will extend our approach to introduce different types of basic vector field, which will include the desired force information in order to compensate for the dynamics of body movements. The force vector field is created not only by the upper-layer controller, but also by the lower-layer controllers based on the information measured by the sensors of the lower subsystems.

References

1. Pearson K (1976) The control of walking. Sci Am 235:72–74, 79–86
2. Grillner S (1975) Locomotion in vertebrates: central mechanisms and reflex interaction. Physiol Rev 55:247–304
3. Gister SF, Loeb E, Mussa-Ivaldi FA, et al. (2000) Repeatable spatial maps of a few force and joint torque patterns elicited by micro-stimulation applied throughout the lumber spinal cord of the spinal frog. Hum Movement Sci 19:597–626
4. Mussa-Ivaldi FA, Gister SF (1992) Vector field approximation: a computational paradigm for motor control and learning. Biol Cybern 67:491–500
5. Odashima T, Yuasa H, Ito M, et al. (1998) The application of an autonomous decentralized system to a myriapod locomotion robot. Proceedings of the 3rd AROB, vol 1, p 114–118
6. Cruse H, Bartling C, Cymbalyuk G, et al. (1995) A modular artificial neural net for controlling a six-legged walking system. Biol Cybern 72:421–430
7. Kawabata K, Kobayashi H (1996) Gait generation for a walking robot by a distributed intelligent control structure (in Japanese). J Robotics Soc Jpn 14:977–985
8. Tsujita K, Tsuchiya K, Onat A (2000) Decentralized autonomous control of a quadruped locomotion robot. Proceedings of the International Symposium on Adaptive Motion of Animal and Machines WeA-I-2
9. Yuasa H, Ito Y, Ito M (1991) Autonomous distributed systems which generate various patterns using bifurcation (in Japanese). Trans SICE 27:1307–1314
10. Odashima T, Yuasa H, Luo ZW (1999) Emergent generation of gait pattern for a myriapod robot system based on energy consumption. Proceedings of IEEE Hong Kong Symposium on Robotics and Control, vol 2, p 387–392
11. http://www.cm-labs.com/
12. http://www.bmc.riken.go.jp/~odashima/Movie.htm
13. Odashima T, Luo ZW, Kishi Y, et al. (2001) Hierarchical control structure of a multi-legged robot for environmental adaptive locomotion. Proceedings of the 4th International Conference on Climbing and Walking Robots, p 105–112
14. Tanaka H, Luo ZW, Odashima T (2002) Teaching of locomotion for a multi-legged robot based on sound perception. Proceedings of SICE 2002 WA17-3