Proceedings of the 2nd International Symposium
on Adaptive Motion of Animals and Machines,
Kyoto, March.4-8, 2003

Hierarchical Design of Dynamics Based Information Processing System for Humanoid Motion Generation

Masafumi OKADA[1], Daisuke NAKAMURA[1] and Yoshihiko NAKAMURA[1,2]

[1]Univ. of Tokyo, Dept. of Mechano-Informatics, Hongo Bunkyo-ku Tokyo 113-8656, Japan
okada@ynl.t.u-tokyo.ac.jp
[2]Japan Science and Technology Corporation

Abstract

The robot motion is an interaction between the robot body and its environment. Because this interaction is continuous phenomenon, the use of dynamics as an information processing system will be a new approach for the robot intelligence. In this paper, we develop the on-line design method and the hierarchical design method of dynamics based information processing system. By using the oblivious parameter, dynamics memorizes a new robot motion forgetting an old motion, which means the plasticity of the system. The hierarchical structure enables information processing for complex and continuous environment. We implement the proposed method to a humanoid robot and realize the motion generation and transition.

1. Introduction

The conventional approach for an artificial intelligence is a symbol manipulation in a virtual world. Each symbol is written by an algorithm and it gives an answer to other symbols. The information processing is represented by the sequence of symbols and it needs so many symbols to treat the environment with a lot of information. Because it needs more than enough processing to deal with the varying environment, robots sometimes do not start to move until completing all the processing, which is so to say the flame problem.

The plastic property is effective for the information processing to make its choice to a lot of information. Because of the plastic property, the neural networks are expected to be a powerful tool for the symbol manipulation based on learning strategy. This method is effective for the functional approximation and make achievements in the fundamental intelligence such as pattern recognition. However, a macro scale network is not realized because of the instability and the hugeness of calculations.

On the other hand, some researchers regard the human brain as a dynamical system and proposed some design methods of the brain function using dynamics. The dynamical phenomenon in organismic brain was shown by Freeman[1, 2, 3]. The rabbit olfactory build has dynamical phenomena such as order for known smell and chaos for unknown smell. On the other hand, Tsuda shows the effectiveness of the chaotic dynamics for learning and calls it 'chaotic itinerancy' for the phenomenon that the human brain transits some attractors [4]. These results show the close relationship between the dynamics and intelligence.

From these results, some researchers tried to design the human brain function using a nonlinear dynamics. Nakamura proposed the motion control method for a mobile robot using chaotic dynamics[5, 6]. Kotosaka generated rhythmic motion using central pattern generator[7]. Matsuyama used dynamical system for the association of a time sequence data[8]. We proposed the dynamics-based information processing system using nonlinear dynamics that has some attractors with a polynomial configuration and realized a motion generation and transition for a humanoid robot[9]. In this method,

- Embedding attractors (closed curved lines) to the dynamics = memorization of the motion
- Autonomous motion of the dynamics (entrainment to an attractor) = production of the motion
- Entrainment of the dynamics based on sensor signals = recognition of the motion

are achieved. This system is expected to be a new approach for robot intelligence.

On the other hand, human brain cortex totalizes the sensory information and recognizes it based on the experience and memory. This feature is classified into frontal cortex, lobus temporalis, parietal lobe and lobus occipitailis cerebri. Lobus temporal processes degustation, acoustic sense, olfaction and language, parietal lobe processes balance control, lobus occipitailis cerebri processes optical sensation. Frontal cortex accretes this information and makes decision. Cerebella controls the tonicity of the muscle and generates the human motion. These parts go to cyclic stability (attractor) by input signal and transmit the information each other, which shows the hierarchical structure. By this architecture, the human brain is able to process a lot of information in the real world, and adapt to the changing environment with learning effects.

In this paper, based on the dynamics based information processing system,

1. We propose on-line embedding method of attractor to dynamics.
2. By setting the forgetting parameter, we realize plastic property for dynamics.
3. We propose the hierarchical design method for dynamics based information processing method for the motion generation based on the input signal.

The proposed method is implemented to humanoid robot and realizes the motion generation and transition based on input sensor signal.

2. Dynamics-based information processing system

2.1. Dynamics and robot whole body motion

In this section, we will talk about the relationship between a dynamics and robot whole body motions. Consider robot motion $\mathcal{M}$. The time sequence data of this motion is assumed to be $\boldsymbol{\xi}[k]$ (for example, joint angles) that composes M as follows.

$$M = \left[\begin{array}{cccc} \boldsymbol{\xi}[1] & \boldsymbol{\xi}[2] & \cdots & \boldsymbol{\xi}[m] \end{array} \right] \tag{1}$$

$$\boldsymbol{\xi}[k] = \left[\begin{array}{cccc} \xi_1[k] & \xi_2[k] & \cdots & \xi_N[k] \end{array} \right]^T \tag{2}$$

m means a number of data and N means the number of degrees of freedom of the robot. Because M composes a carved line in N dimensional space, when $\mathcal{M}$ is a cyclic motion, M shows the closed curved line.

On the other hand, consider the dynamics represented by the following difference equation.

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{f}(\boldsymbol{x}[k]) \tag{3}$$

$$\boldsymbol{x}[k] = \left[\begin{array}{cccc} x_1[k] & x_2[k] & \cdots & x_N[k] \end{array} \right]^T \tag{4}$$

If M is an attractor of this dynamics, the state vector $\boldsymbol{x}[k]$ converges to $\boldsymbol{\xi}[k]$ with initial state $\boldsymbol{x}[0]$, which means that the dynamics memorizes the motion M. And by picking up $\boldsymbol{x}[k]$ $(k = 0, 1, 2, \cdots)$, we can obtain $\boldsymbol{\xi}[k]$ $(k = 0, 1, 2, \cdots)$, which means that the dynamics reproduces time sequence data M of motion $\mathcal{M}$.

2.2. Design algorithm of the dynamics

In equation (3), we can consider that $\boldsymbol{f}(\boldsymbol{x}[k])$ defines the vector field in N dimensional space. By using the polynomial functional approximation of the vector field, the dynamics in equation (3) can be calculated[9]. The design algorithm of the dynamics is as follows.

Step 1 Draw the closed curved line M in N dimensional space.

Step 2 Set the basin D of attractor and define points $\boldsymbol{\eta}_i$ and vector of $\boldsymbol{f}(\boldsymbol{\eta}_i)$ so that the closed curved line M becomes attractor. Figure 1 shows the example of the definition of vector field

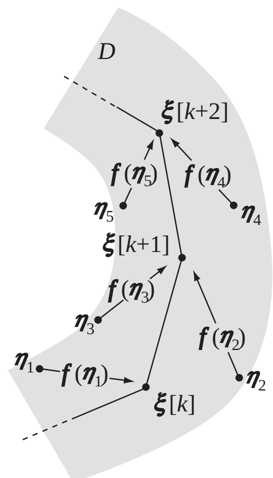


Figure 1: Definition of the vector field

Step 3 The defined vector $\boldsymbol{f}(\boldsymbol{\eta}_i)$ is approximated by the following equation by ℓ-th order poly-

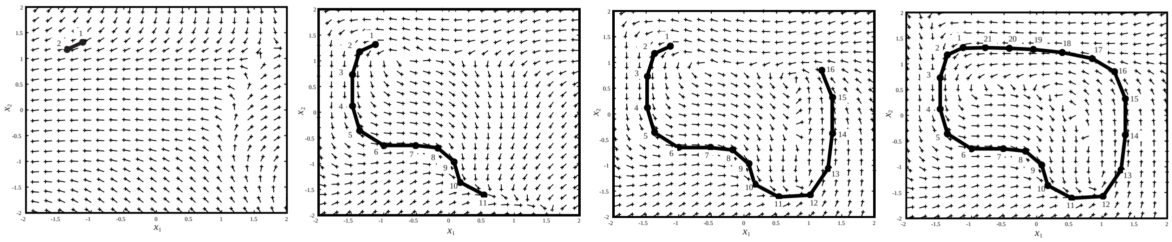


Figure 2: On-line design of dynamics

nomial of $\boldsymbol{x}[k]$.

$$\boldsymbol{f}(\boldsymbol{\eta}) = \sum_{P=0}^{\ell} \sum_{\substack{p_1,\cdots,p_n \\ \sum p_i = P \\ p_i\,:\,\text{positive integer}}}^{\ell} a_{(p_1 p_2 \cdots p_n)} \prod_{i=1}^{n} \eta_i^{p_i} \quad (5)$$

$$\boldsymbol{\eta} = \left[\begin{array}{cccc} \eta_1 & \eta_2 & \cdots & \eta_N \end{array} \right]^T \quad (6)$$

$a_{(p_1\ p_2\ \cdots p_N)}$ is a constant. Defining $\boldsymbol{f}(\boldsymbol{\eta})$ as

$$\boldsymbol{f}(\boldsymbol{\eta}) = \Phi(a_{(p_1\ p_2\ \cdots p_N)})\boldsymbol{\theta}(\boldsymbol{\eta}) \quad (7)$$

$$\boldsymbol{\theta}(\boldsymbol{\eta}) = \left[\begin{array}{cccccc} \eta_1^{\ell} & \cdots & \eta_N^{\ell} & \eta_1^{\ell-1}\eta_2 & \cdots & 1 \end{array} \right]^T \quad (8)$$

Φ is calculated by the least square method as follows.

$$\Phi(a_{(p_1\ p_2\ \cdots p_N)}) = F\Theta^{\#} \quad (9)$$

$$F = \left[\begin{array}{cccc} \boldsymbol{f}(\boldsymbol{\eta}_1) & \boldsymbol{f}(\boldsymbol{\eta}_2) & \cdots & \boldsymbol{f}(\boldsymbol{\eta}_m) \end{array} \right] \quad (10)$$

$$\Theta = \left[\begin{array}{cccc} \boldsymbol{\theta}(\boldsymbol{\eta}_1) & \boldsymbol{\theta}(\boldsymbol{\eta}_2) & \cdots & \boldsymbol{\theta}(\boldsymbol{\eta}_m) \end{array} \right] \quad (11)$$

Φ is constant parameter matrix that defines the dynamics in equation (3).

3. On-line design of the dynamics

3.1. On-line design method

In equation (9), Φ that defines $\boldsymbol{f}(\boldsymbol{x}[k])$ is designed based on the least square method. By using on-line least square method, the dynamics is on-line designed, which means the dynamics memorizes the humanoid motion successively. The parameter matrix Φ^m in time step m is calculated by the iteration of the following on-line least square algorithm using a non-singular matrix P_m.

$$P_{m+1} = P_m - \frac{P_m \boldsymbol{\theta}(\boldsymbol{\eta}_{m+1})\boldsymbol{\theta}^T(\boldsymbol{\eta}_{m+1})P_m}{1 + \boldsymbol{\theta}^T(\boldsymbol{\eta}_{m+1})P_m\boldsymbol{\theta}(\boldsymbol{\eta}_{m+1})} \quad (12)$$

$$\begin{aligned} \Phi^{m+1} &= \Phi^m + (\boldsymbol{f}(\boldsymbol{\eta}_{m+1}) \\ &\quad - \Phi^m \boldsymbol{\theta}(\boldsymbol{\eta}_{m+1}))\boldsymbol{\theta}^T(\boldsymbol{\eta}_{m+1})P_{m+1} \quad (13) \end{aligned}$$

P_m corresponds to

$$P_m = (\Theta\Theta^T)^{-1} \quad (14)$$

where Θ is defined in equation (11).

Figure 2 shows an example of the successive change of vector field $\boldsymbol{f}(\boldsymbol{x}[k])$ with $N = 2$. "+" means the arrow of the vector field. The length of vectors is normalized for easy view. The dynamics that has an attractor is gradually designed.

3.2. Set of the forgetting parameter

In this section, we set the forgetting parameter to the on-line least square method, which causes the plasticity of the dynamics. As the alternation of equation (9), we consider the following weighted equation.

$$\Phi = \widehat{F}\widehat{\Theta}^{\#} \quad (15)$$

$$\widehat{F} = \left[\begin{array}{cc} \alpha F & \boldsymbol{f}(\boldsymbol{\eta}_{m+1}) \end{array} \right] \quad (16)$$

$$\widehat{\Theta} = \left[\begin{array}{cc} \alpha\Theta & \boldsymbol{\theta}(\boldsymbol{\eta}_{m+1}) \end{array} \right] \quad (17)$$

α $(0 < \alpha < 1)$ is a constant parameter. Φ is calculated by the same way as equation (12) and (13) after P_m is replaced by

$$P_m = \alpha^{-2} P_m \quad (18)$$

This method means the on-line forgetting least square method. By using this method, the dynamics memorizes the newer time sequence data forgetting the older one. Figure 3 shows the motion of the designed dynamics. The attractor is embedded by the order of $1 \to 2 \to 3$. Some $\bullet$ mean some initial states $\boldsymbol{x}[0]$. The first embedded closed curved line is forgotten, a part of the attractor of the second closed curved line is remained. This means that the dynamics has plastic property.

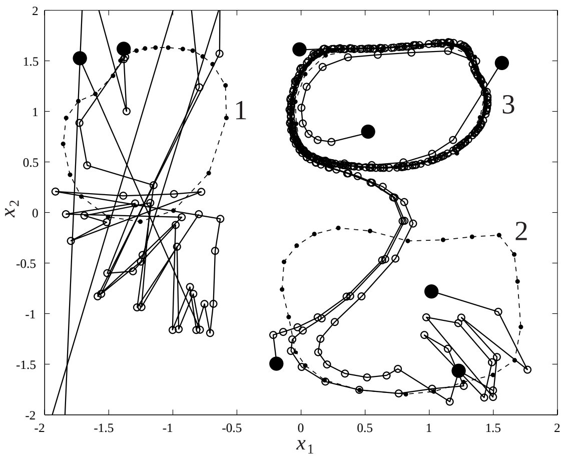


Figure 3: Motions of dynamics with weighted on-line algorithm

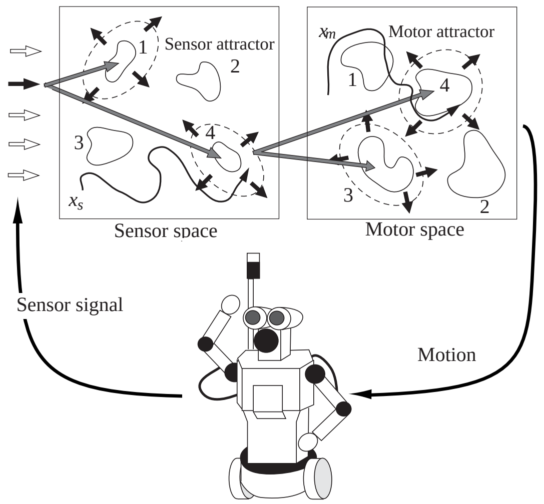


Figure 4: Hierarchical design of sensor space and motor space

4. Hierarchical design of the dynamics based information processing system

4.1. Hierarchical design

The dynamical system memorizes the time sequence data M as an attractor and the state vector entrained to the attractor autonomously. In this section, based on the human brain cortex model, we design the hierarchical model so that the dynamics transits the attractors using external inputs. Even if the external inputs are same, produced motions are different because of the difference of internal state of the dynamics.

The hierarchical structure means the increase of the dimension of the dynamics, which causes the increase of design parameters. Design of the large dimensional dynamics is not only difficult because of the small computer power but also causes unclearness of the structure of the system, which prevent heuristic design. On the other hand, the hierarchical structure enables the large dimensional dynamics with clear structure of the system and appropriated design for our objective.

4.2. Hierarchical structure

We set two spaces shown in Figure 4. One is the sensor space and the other is the motor space.

Sensor space : The sensor space is a virtual torus space. There are some sensor attractors in the sensor space. The state vector $\boldsymbol{x}_s[k]$ moves based on the vector field and the basins of each attractor are decided by the external sensor signal. When the vector field on $\boldsymbol{x}_s[k]$ is zero (undefined), it takes chaotic motion so that $\boldsymbol{x}_s[k]$ goes around and searches another basin. Once the state vector goes into the basin of the attractor, it is entrained to corresponding attractor. By the change of the sensor signal, $\boldsymbol{x}_s[k]$ transits to the other attractor.

Motor space : In the motor space, there are some motor attractors that define the humanoid motions directory. The entrainment of the dynamics in the sensor space decides the basin of the motor space. The state vector $\boldsymbol{x}_m[k]$ in motor space moves according to the vector field and produces the humanoid motion. The change of motion causes the change of the sensor signal that means the feedback of the signal through the environment.

4.3. Local entrainment area and global basin

For the hierarchical design of the dynamics based information processing system, it needs the design strategy of the dynamics that has some attractors whose basin is changeable. Because the dynamics is designed using polynomial function with locally defined vector field $\boldsymbol{f}(\boldsymbol{\eta}_i)$ in equation (10), $\boldsymbol{f}(\boldsymbol{x}[k])$ should be effective only in the neighborhood of the attractor.

Local entrainment area : The difference equation in equation (3) is changed as follows.

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + w_1 \boldsymbol{f}(\boldsymbol{x}[k]) \quad (19)$$

where w_1 is defined using a_1 as follows.

$$w_1 = 1 - \frac{1}{1+\exp\{a_1(\omega_1(\boldsymbol{x}[k]) - 1)\}} \quad (20)$$

$$\omega_1(\boldsymbol{x}[k]) = (\boldsymbol{x}^T[k] - \boldsymbol{X}_0^T)Q(\boldsymbol{x}[k] - \boldsymbol{X}_0) \quad (21)$$

where Q defines an ellipsoid that connotes closed curved line M in N dimensional space with the center $\boldsymbol{X}_0$. When $\boldsymbol{x}[k]$ is inside the ellipsoid, w_1 is 1 and otherwise 0.

Global basin : Global basin is defined by the following equation,

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + w_2 (w_1 \boldsymbol{f}(\boldsymbol{x}[k]) + (1-w_1)\delta(\boldsymbol{X}_c - \boldsymbol{x}[k])) \quad (22)$$

where δ $(0 < \delta < 1)$ and w_2 is defined as follows with a_2 and $\boldsymbol{X}_c$ the center of M.

$$w_2 = 1 - \frac{1}{1+\exp\{a_2(\omega_2(\boldsymbol{x}[k]) - 1)\}} \quad (23)$$

$$\omega_2(\boldsymbol{x}[k]) = K(\boldsymbol{x}^T[k] - \boldsymbol{X}_0^T)Q(\boldsymbol{x}[k] - \boldsymbol{X}_0) \quad (24)$$

This means that K defines the global basin. Figure 5 shows the pattern diagram of the dynamics. E_1 shows the ellipsoid that is defined

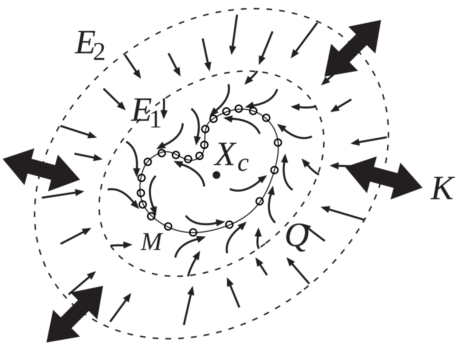


Figure 5: Change of basin of attractor

by the right side in equation (21) = 0 and E_2 is defined by the right side in equation (24) = 0.

w_1 is 1 when $\boldsymbol{x}[k]$ exists inside E_1 otherwise $w_1 = 0$. $\boldsymbol{f}(\boldsymbol{x}[k])$ is effective inside E_1 and inside E_2. Basin is defined outside E_1 and inside E_2. The attractor will disappear at $K \to \infty$.

4.4. Design of nonlinear dynamics with multi attractors

The design of the dynamics with multi attractors is designed by sum of vector fields as the following equation.

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \sum_i w_{2i}\{w_{1i}\boldsymbol{f}_i(\boldsymbol{x}[k]) + (1-w_{1i})\delta_i(\boldsymbol{X}_{ci} - \boldsymbol{x}[k])\} + \prod_i (1-w_{2i})\boldsymbol{x}^{\mathrm{chaos}}[k] \quad (25)$$

where $\boldsymbol{x}^{\mathrm{chaos}}[k]$ means the chaos term so that the state vector wonders in the space when the vector field is not defined because of no sensor signal.

5. Motion generation of the humanoid robot

5.1. Upper body humanoid robot

In this section, we implement the proposed system to the humanoid robot. Figure 6 shows the upper body humanoid robot ' Robovie '. This robot

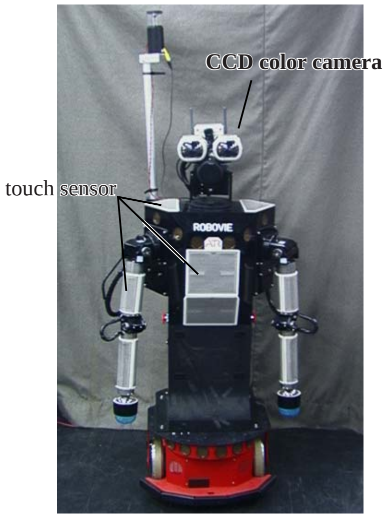
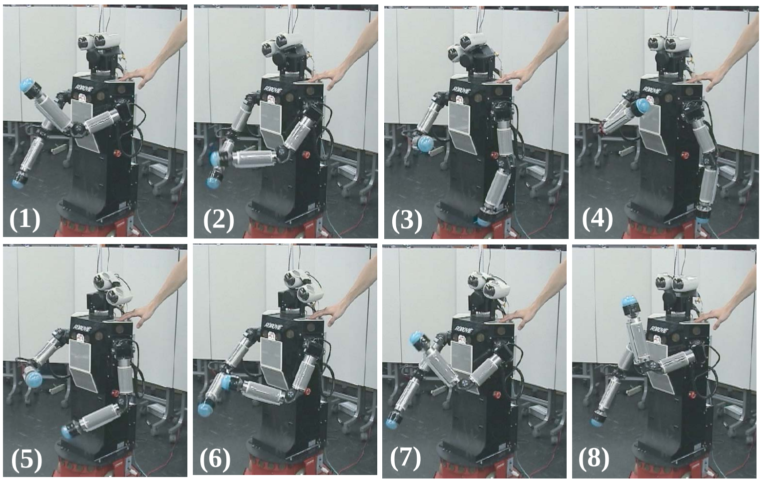
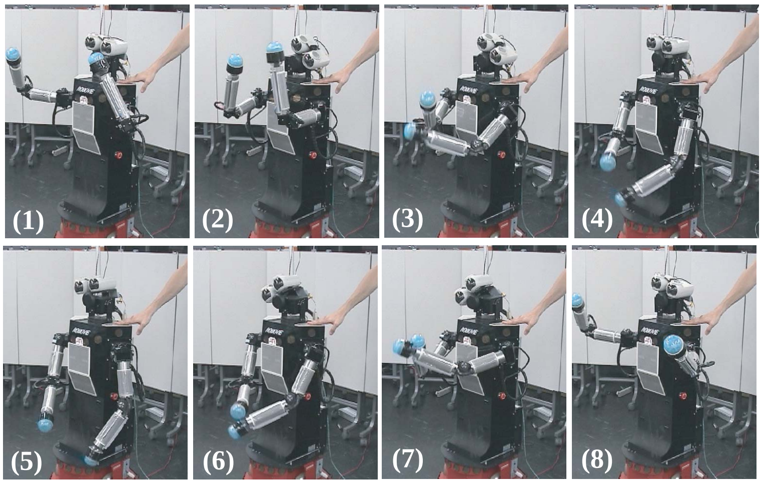


Figure 6: Humanoid robot Robovie

has 3 degrees-of-freedom on neck, 3 degrees-of-freedom on each shoulders and 1 degree-of-freedom on each elbows. The total degree-of-freedom is 11. This robot has 16 touch sensors on head, shoulder, breast, upper arm, lower arm and wrist. Each sensors yield on-off signal. The color CCD camera obtains color information of the environment (per-



Motion 1



Motion 2

Figure 7: Motions of the humanoid robot

centage of red, green and blue). Total number of sensor signals is 19. We design 10 motions to this robot. Figure 7 shows the sample of the motion (motion1 and motion 2).

5.2. Design of the dynamics and network

Because this robot has 11 degrees-of-freedom, motor space (joint angle space) has 11th order dimension. However to design the high dimensional dynamics is unrealistic because of the small computer power and numerical instability. In this paper, motor space is reduced to 3rd order dimension using principal component analysis reduction method [9] and the dynamics is designed as follows.

$$\boldsymbol{x}^m[k+1] = \boldsymbol{x}^m[k] + \sum_{i=1}^{10} w_{2i}^m \{ w_{1i}^m \boldsymbol{f}_i^m(\boldsymbol{x}^m[k])$$
$$+(1-w_{1i}^m)\delta_i^m(\boldsymbol{X}_{ci}^m - \boldsymbol{x}^m[k])\} \tag{26}$$
$$\boldsymbol{x}^m[k] \in \boldsymbol{R}^3 \tag{27}$$

$\boldsymbol{x}^m[k]$ means the state vector in motor space and the subscript m means the motor space. Using $F_i(\in \boldsymbol{R}^{11\times 3})$ that decompresses $\boldsymbol{x}^m[k]$ to 11th order vector, the robot joint angle $\boldsymbol{y}[k](\in \boldsymbol{R}^{11})$ is calculated by

$$\boldsymbol{y}[k] = \sum_{i=1}^{10} w_{2i}^m w_{1i}^m F_i \boldsymbol{x}^m[k] \tag{28}$$

where $w_{2i}^m w_{1i}^m$ is used for the index of entrainment to i-th attractor.

The parameter K_i^m in equation (24) that defines the basin of i-th attractor, is decided by the entrainment of $\boldsymbol{x}_i^s[k]$ that is the state vector in sensor space (subscript s means the sensor space). 10 attractors are embedded in the sensor space. The input vector from touch sensor space (0 or 1) u_{ti} $(i = 1,\ 2,\ \cdots, 16)$ is defined as

$$\boldsymbol{u}_t = \left[\begin{array}{cccc} u_{t1} & u_{t2} & \cdots & u_{t16} \end{array}\right]^T \tag{29}$$

and the input vector from visual sensor space (percentage of red, green and blue) is defined as

$$\boldsymbol{u}_i = \left[\begin{array}{ccc} u_{iR} & u_{iG} & u_{iB} \end{array}\right]^T \tag{30}$$

The parameter $\boldsymbol{K}$ that decides the basin of the sensor space

$$\boldsymbol{K}^s = \left[\begin{array}{cccc} K_1^s & K_2^s & \cdots & K_{10}^s \end{array}\right]^T \tag{31}$$

is represented as follows using the weighting matrix ${}^sW_s$

$$\boldsymbol{K}^s = {}^sW_s \left[\begin{array}{c} \boldsymbol{u}_t \\ \boldsymbol{u}_i \end{array}\right] \tag{32}$$

The parameter $\boldsymbol{K}^m$

$$\boldsymbol{K}^m = \left[\begin{array}{cccc} K_1^m & K_2^m & \cdots & K_{10}^m \end{array}\right] \tag{33}$$

that decides the basin of motor attractor is defined as follows.

$$\boldsymbol{K}^m = {}^mW_s \boldsymbol{w}_1^s \tag{34}$$

$$\boldsymbol{w}_1^s = \left[\begin{array}{cccc} w_{21}^s w_{11}^s & w_{22}^s w_{12}^s & \cdots & w_{2\cdot 10}^s w_{1\cdot 10}^s \end{array}\right]^T \tag{35}$$

$w_{2i}^s w_{1i}^s$ is used as the index of the entrainment of attractor that is same as equation (28).

5.3. Motion generation

Using the designed dynamics, we realize the humanoid motion generation and transition based on the sensor signal. The input of touch sensor is changed as right shoulder $(t_1 < t < t_2) \rightarrow$ right head $(t_3 < t < t_4) \rightarrow$ right arm $(t_5 < t)$ for time t. We make two experiments with different timing of touches. Figure 8 shows the motion of the state vector in motor space. By the difference of the timing of the touch sensor inputs, the different motion is generated, that means the humanoid motion depends on the internal space of the humanoid robot. Figure 9 shows the index of the entrainment of

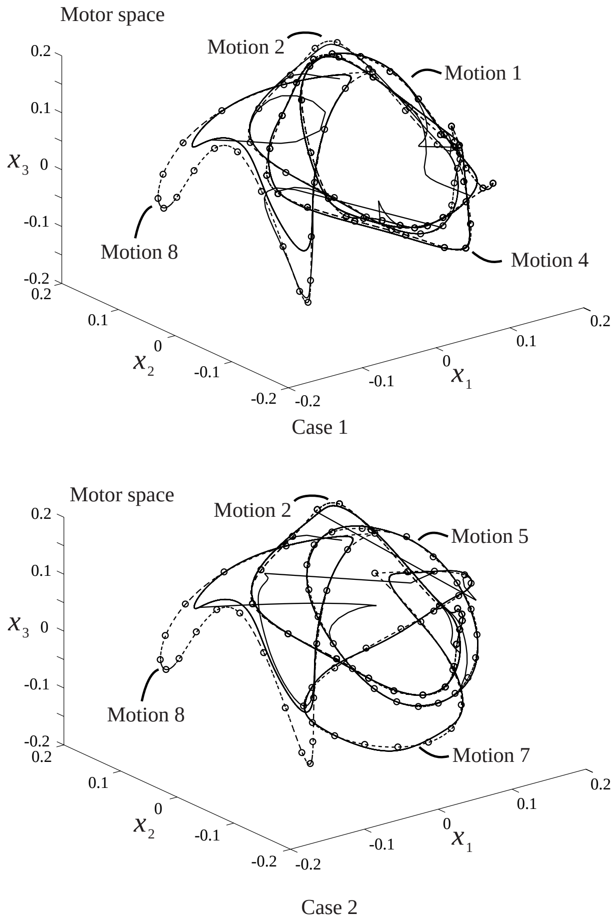


Figure 8: Motion of the dynamics in motor space

the attractor (fired attractor) in sensor space and generated motion. Density means the index value of the entrainment. Because the sequence of the touch sensor input is same, same attractors are fired in case 1 and 2, however, because of the difference of timing of input signal, different motions are generated.

6. Conclusions

In this paper, we proposed the hierarchical design method using the dynamics based information processing system with polynomial function. The results of this paper are as follows.

- By setting the input and output to nonlinear dynamics, we design the hierarchical structure of dynamics based information processing system.
- By the change of basins of attractors, the state

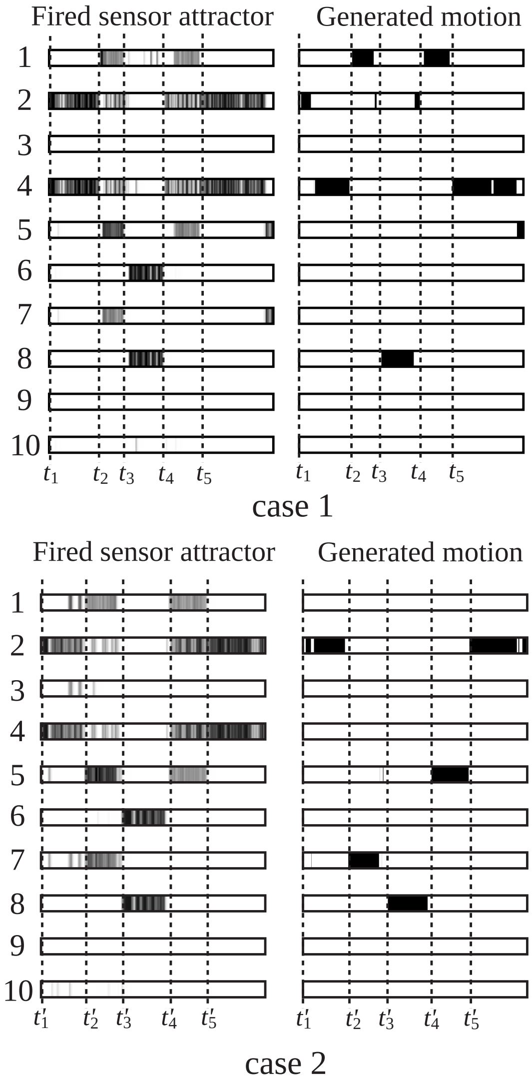


Figure 9: Fired attractors and generated motions

vector of the dynamics is entrained and transits to attractors.

- By the implementation of the designed dynamical system to the humanoid robot, we realize the motion generation and transition of the humanoid robot based on the sensor signal and the internal state.

Acknowledgments

This research is supported by the "Robot Brain Project" under the Core Research for Evolutional Science and Technology (CREST program) of the Japan Science and Technology Corporation.

References

[1] W.J.Freeman and W.Schneider : Changes in Spatial Patterns of Rabbit Olfactory EEG with Conditioning to Odors, Psychophysiology, Vol.19, pp.44–56, 1982.

[2] W.J.Freeman: Simulation of Chaotic EEG Patterns, Nonlinear Dynamic model of the Olfactory Systems, Biological Cybernetics, Vol.56, pp.139–150, 1987.

[3] Y.Yao and W.J.Freeman: Model of Biological Pattern Recognition with Spatially Chaotic Dynamics, Neural Networks, Vol.3 pp.153–160, 1990.

[4] I.Tsuda: Chaotic itinerancy as a dynamical basis of Hermeneutics in brain and mind, World Futures, 31 pp.105–122, 1991

[5] A.Sekiguchi and Y.Nakamura : The Chaotic Mobile Robot, Proc. of Systemics, Cybernetics and Informatics 2000 Vol.9, pp.463–468, 2000

[6] A.Sekiguchi and Y.Nakamura : Behavior Control of Robot Using Orbits of Nonlinear Dynamics, Proc. of IEEE International Conference on Robotics and Automation, pp.1647–1652, 2001

[7] S.Kotosaka and S.Schaal: Synchronized robot drumming by neural oscillators, Proc. of the International Symposium on Adaptive Motion of Animals and Machines, pp.8–12, 2000.

[8] H.Kawashima and T.Matsuyama: Integrated Event Recognition from Multiple Sources, Proc. of International Conference on Pattern Recognition, Vol.2, pp.785-789, 2002.

[9] M.Okada, K.Tatani and Y.Nakamura: Polynomial Design of the Nonlinear Dynamics for theBrain-Like Information Processing of Whole Body Motion, Proc. of IEEE International Conference on Robotics and Automation, pp1410-1415, 2002: