Research Subject

Liapunov Design of Dynamical Information Processing and Transition Control of Motion Patterns (Ushio Group)

(1) Goal and summary

Efficient representations of primitive motions of humanoid robots are important in order to realize various complicated motions. A graphical representation called a state net is very useful for describing the transitions between basic positions [1]. In the state net, each static position is described by a node in a sensory space, and transitions between positions are described by directed arcs from a current node to the next node. So, both addition of new motions and modifications of motions are very easy. However, it is not included to represent periodic motions such as walk.

Since every periodic motion forms a limit cycle in the sensory space, we can extend the state net where a limit cycle is described by a node. Thus, we can use the state net for transitions between periodic motions, too. In general, however, a dimension of the sensory space is very large and it is difficult to construct dynamics in the sensory space, which generates the limit cycle. In order to overcome the problem, Okada *et al.* reduced the dimension of data using the singular decomposition method and constructed dynamics for the reduced data [2]. This method needs much time for computing a vector field on the reduced data space.

On the other hand, studies on design of dynamical systems which have a specified stable limit cycles have been done in nonlinear system theory. Green derived a constraint function which specifies the limit cycle and proposed a Lyapunov function based method for design of a desired dynamical system from the constraint function [3].

In this research, applying Green's method, we show a method for representations of periodic motions of humanoid robots using dynamical systems. Especially, we propose a method for representation by nonsmooth limit cycles using hybrid dynamical systems. Moreover, we extend the state net so that transitions between periodic motions are realized. This extension will be called a hybrid state net.

First, We consider a humanoid robot with n degrees-of-freedom. A periodic motion is given by the discrete-time data sequence M:

$$M = \begin{bmatrix} \theta_1[t_1] & \theta_1[t_2] & \cdots & \theta_1[t_T] \\ \theta_2[t_1] & \theta_2[t_2] & \cdots & \theta_2[t_T] \\ \vdots & \vdots & & \vdots \\ \theta_n[t_1] & \theta_n[t_2] & \cdots & \theta_n[t_T] \end{bmatrix}, \tag{1}$$

where $\theta_i[t_j]$ is the ith joint angle at the time t_j. We reduce the motion data M to a lower-dimensional data. By applying a reduction method using the singular value decomposition

[2], the reduced data is obtained as follows:

$$
\begin{aligned}
M &= USV^T, \\
U &= \left[\ U_1 \ \middle| \ U_2\ \right], \\
S &= \mathrm{blockdiag}\{S_1\ S_2\}, \\
S_1 &= \mathrm{diag}\{s_1\ s_2\ \ldots\ s_m\}, \\
S_2 &= \mathrm{diag}\{s_{m+1}\ s_{m+2}\ \ldots\ s_n\}, \\
V &= \left[\ V_1 \ \middle| \ V_2\ \right],
\end{aligned}
$$

if $s_m \gg s_{m+1}$ is satisfied, the motion data M is reduced to m-dimensional data V_1^T as follows:

$$M = U_1 S_1 V_1^T. \tag{2}$$

The vector V_1^T is called a reduced motion data Y and $\Psi = U_1 S_1$ is a reconstruction function from the reduced motion data space to the original one.

We consider the following two cases:

(a) The motion data Y is approximated by a **smooth** closed curve.

(b) The motion data Y is approximated by a **nonsmooth** closed curve.

In the case (a), we use a method proposed by Green [3]. Any closed curve in an n-dimensional space can be expressed by using Fourier series and Chebyshev polynomials as follows :

$$
\begin{aligned}
y(t) &= \sum_{k=0}^{\infty} \Big\{ \alpha_k \cos(k\omega t) + \beta_k \sin(k\omega t) \Big\} \\
&= \sum_{k=0}^{\infty} \Big\{ a_k T_k(\cos(\omega t)) + b_k \sin(\omega t) U_k(\cos(\omega t)) \Big\} \\
&= F_1(\cos(\omega t)) + \sin(\omega t) F_2(\cos(\omega t)),
\end{aligned} \tag{3}
$$

where $y(t) \in \mathbf{R}^n$ is a state vector, both α_k and $\beta_k \in \mathbf{R}^n$ are coefficient vectors, both a_k and b_k are constants depending on α_k and β_k, T_k and U_k are the kth Chebyshev polynomials of the first and second kind, respectively, F_1, $F_2 : [-1\ 1] \to \mathbf{R}^n$. We construct the following differential equation:

$$
\begin{aligned}
\dot{\mathbf{y}} &= f(\mathbf{y}) + g(\mathbf{y}) \\
&= \omega \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ \dfrac{-\partial F_1(x_2)}{\partial x_2} - x_1 \dfrac{\partial F_2(x_2)}{\partial x_2} & F_2(x_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y \end{bmatrix} \\
&\qquad\qquad + \alpha \begin{bmatrix} x_1(1 - x_1^2 - x_2^2) \\ x_2(1 - x_1^2 - x_2^2) \\ F_1(x_2) + x_1 F_2(x_2) - y \end{bmatrix},
\end{aligned} \tag{4}
$$

where $\mathbf{y} = [x^T \ y^T]^T$, $x \in \mathbf{R}^2$, $y \in \mathbf{R}^n$, and $\alpha > 0$ represents the convergence rate. Then there exists the following constraint function $V : \mathbf{R}^{n+2} \rightarrow \mathbf{R}^{n+1}$:

$$V(\mathbf{y}) = \begin{bmatrix} x_1^2 + x_2^2 - 1 \\ F_1(x_2) + x_1 F_2(x_2) - y \end{bmatrix}. \tag{5}$$

All trajectories of (4) converge to the hypersurface $V(\mathbf{y}) = 0$, i.e., $x_1(t) = \sin(\omega t)$, $x_2(t) = \cos(\omega t)$ and $y(t) = F_1(x_2(t)) + x_1(t)F_2(x_2(t))$, respectively as $t \rightarrow \infty$. This result shows that this system can generate an arbitrary asymptotically stable limit cycle. So, the vector x is an augmented state vector which is used for convergence of the original state vector y to the limit cycle represented by Eq.(3).

From the practical point of view, Eq.(3) can be approximated as follows:

$$y(t) = \sum_{k=0}^{l} \Big\{ \alpha_k \cos(k\omega t) + \beta_k \sin(k\omega t) \Big\}, \tag{6}$$

where l is a sufficiently large integer number such that Eq.(6) is a good approximation of Y. Let

$$\Lambda = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_l & \beta_1 & \cdots & \beta_l \end{bmatrix}^T$$

be a coefficient vector of Eq.(6). Then, using the least square method, we have

$$\Lambda = (\Phi^T \Phi)^{-1} \Phi^T Y^T, \tag{7}$$

where

$$\Phi = \begin{bmatrix} 1 & \cos(t_1) & \cdots & \cos(lt_1) & \sin(t_1) & \cdots & \sin(lt_1) \\ 1 & \cos(t_2) & \cdots & \cos(lt_2) & \sin(t_2) & \cdots & \sin(lt_2) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(t_T) & \cdots & \cos(lt_T) & \sin(t_T) & \cdots & \sin(lt_T) \end{bmatrix}. \tag{8}$$

Thus, we get the function $y(t)$ which is an approximation of Y as follows:

$$y(t) = F_1(\cos(\omega t)) + \sin(\omega t) F_2(\cos(\omega t)). \tag{9}$$

By substituting both functions F_1 and F_2 into (4), we obtain the desired system.

Figures 1 and 2 show a time sequence of reduced “walk” motion data $Y_{\mathrm{w}} \in \mathbf{R}^3$ and its closed curve, respectively.

Figure 3 shows a trajectory of the designed system where $l = 9$. When l is sufficiently large, this system generates an asymptotically stable limit cycle corresponding to the closed curve of the reduced motion data Y.

We summarize the proposed synthesis procedure of smooth motion patterns as follows:

Step1: A given periodic motion data M which forms a closed curve in the n-dimensional space is transformed to $M = USV^T$ by the singular value decomposition.

Step2: By choosing an appropriate singular value s_m, we get $M = \Psi Y$, where Y is a reduced motion data of the original one.

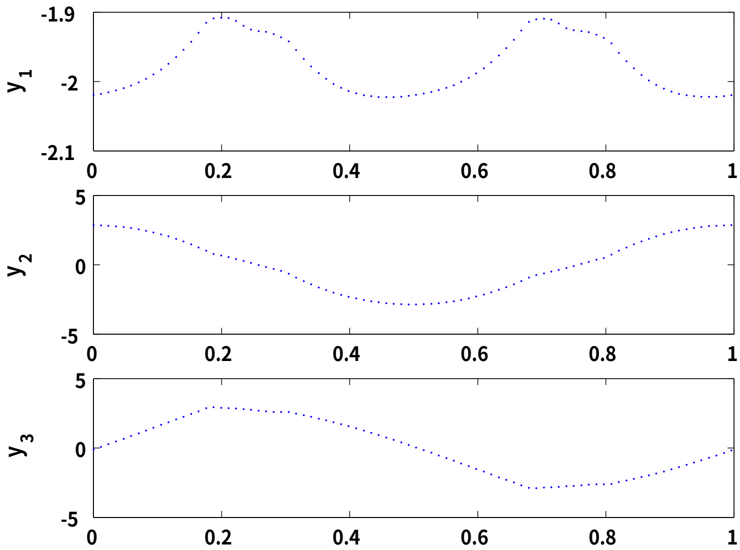

Figure 1: Time sequence of reduced "walk" motion.

Step3: We approximate the reduced motion data Y to a Fourier series expansion $y(t) = F_1(\cos(\omega t)) + \sin(\omega t)F_2(\cos(\omega t))$ by the least square method.

Step4: The desired system which has a periodic motion pattern as an asymptotically stable limit cycle is obtained from Eq.(4).

Step5: The sensor data in the n-dimensional space is restored by the reconstruction function Ψ from the reduced motion data space to the original one.

Next, we consider the case (b) and propose a synthesis method for nonsmooth motion generation. According to the nonsmoothness of motions, we synthesize switched systems or hybrid systems [4] whose vector fields are discontinuous. Furthermore, in order to guarantee the stability of nonsmooth limit cycle, we employ the concept of piecewise quadratic Lyapunov functions [5].

We partition the reduced motion data Y into several components as follows:

$$Y = \left[\begin{array}{c|c|c|c} Y_1 & Y_2 & \cdots & Y_Q \end{array}\right] = \left[\begin{array}{cc|c|cc} y_1[t_1] & \cdots & \cdots & \cdots & y_1[t_T] \\ y_2[t_1] & \cdots & \cdots & \cdots & y_2[t_T] \\ \vdots & & & & \vdots \\ y_m[t_1] & \cdots & \cdots & \cdots & y_m[t_T] \end{array}\right]. \tag{10}$$

The reduced space is partitioned into several polyhedral regions whose boundaries are given by hyperplanes, and the periodic motions are represented by an ellipsoidal curve in each region $\mathcal{R}_q(q = 1,\ 2,\ \ldots,\ Q)$ as shown in Figure 4.

By constructing subsystems corresponding to each ellipsoidal curve and switching those subsystems, the nonsmooth motion pattern can be realized. Assume that each component $Y_q(q = 1,\ 2,\ \ldots,\ Q)$ can be approximated by an elliptic cylinder and a hyperplane (Figure 5). Then, the approximated ellipsoidal curve is given by $\mathcal{V}_q(y) = 0$, where

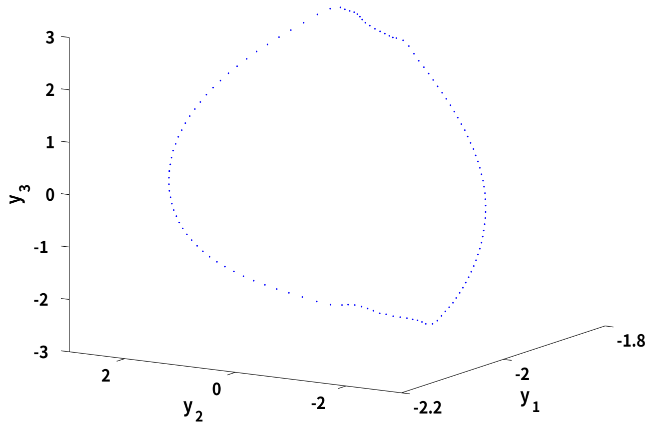

Figure 2: Closed curve of the reduced “walk” motion.

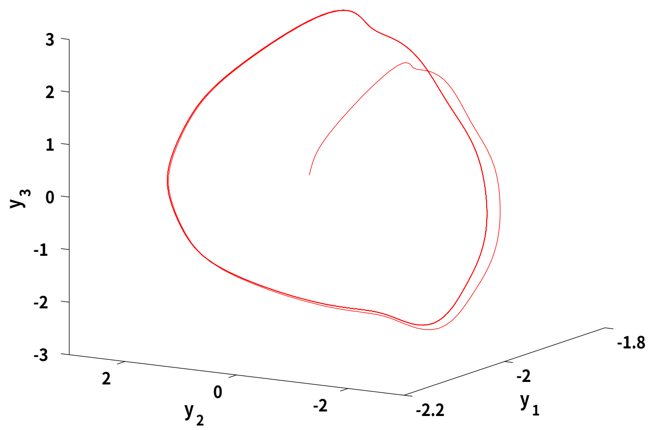

Figure 3: A trajectory of the designed system with reduced “walk” motion.

$\mathcal{V}_q : \mathbf{R}^m \to \mathbf{R}^{m-1}$:

$$\mathcal{V}_q = \begin{bmatrix} \mathcal{V}_{q1} \\ \mathcal{V}_{q2} \\ \vdots \\ \mathcal{V}_{q_{m-1}} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} y_1 & y_2 & 1 \end{bmatrix} \begin{bmatrix} P_q & p_q \\ p_q & \pi \end{bmatrix} \begin{bmatrix} y_1 & y_2 & 1 \end{bmatrix}^T \\ \zeta_{q1} y_1 + \eta_{q1} y_2 + \lambda_{q1} - y_3 \\ \vdots \\ \zeta_{q_{m-2}} y_1 + \eta_{q_{m-2}} y_2 + \lambda_{q_{m-2}} - y_m \end{bmatrix}, \tag{11}$$

where $P_q \in \mathbf{R}^{2\times 2}$ is a positive definite matrix, and π, p_q, ζ_{q_i}, η_{q_i}, and λ_{q_i} are real parameters. Here, in order to guarantee the stability of limit cycles of this form, $\mathcal{V}_{q1}(y)$ is not allowed to choose freely when subsystems are combined. The requirement is the continuity of constraint functions $\mathcal{V}_{q1}(y)$ on all region boundaries (Figure 6). The boundary

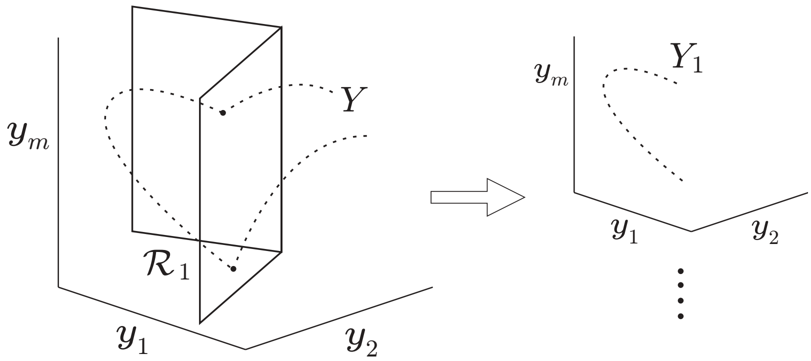

Figure 4: Partition of a nonsmooth closed curve into several ellipsoidal curves.

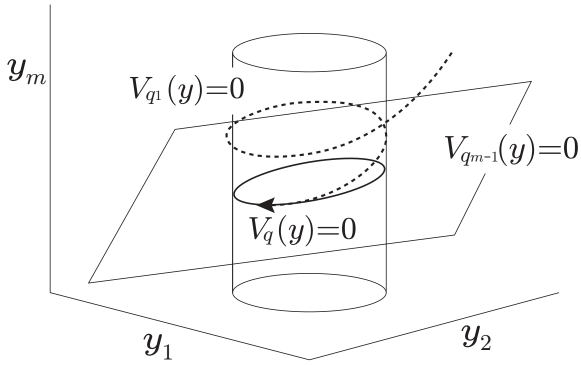

Figure 5: An ellipsoidal curve defined by an elliptic cylinder and a hyperplane.

between $\mathcal{R}_q$ and $\mathcal{R}_r$ is given by

$$\tilde{c}_{q,r}^T \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = 0, \tag{12}$$

where $\tilde{c}_{q,r} = [c_{q,r}^T \; d_{q,r}]^T$, and $c_{q,r} \in \mathbf{R}^2$, $d_{q,r} \in \mathbf{R}$. The constraint function $\mathcal{V}_{q_1}(y)$ is continuous on all boundaries if and only if, for some $\tilde{t}_{q,r} \in \mathbf{R}^3$,

$$\tilde{P}_r = \tilde{P}_q + \tilde{t}_{q,r}^T \tilde{c}_{q,r} + \tilde{c}_{q,r}^T \tilde{t}_{q,r}, \tag{13}$$

where $\tilde{P}_q = \begin{bmatrix} P_q & p_q \\ p_q^T & \pi \end{bmatrix}$. If each region forms a polyhedron with pairwise disjoint interior, there exist the following matrices for each region $\mathcal{R}_q$ [5]:

$$\tilde{F}_q = [F_q \; f_q],$$

where, for any $y \in \mathcal{R}_q \cap \mathcal{R}_r$ $(q, r \in \{1, 2, \ldots, Q\})$,

$$\tilde{F}_q \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = \tilde{F}_r \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix}. \tag{14}$$

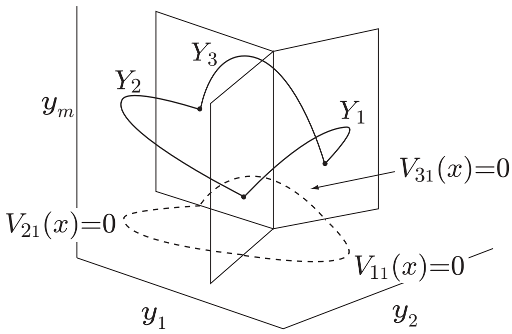

Figure 6: Continuity of $\mathcal{V}_{q_1}$ on all region boundaries.

By using this representation, the requirement that a constraint function is continuous at every point on the region boundaries can be written as

$$\tilde{P}_q = \tilde{F}_q T \tilde{F}_q, \tag{15}$$

where T is a symmetric matrix.

For ellipsoidal curves $\mathcal{V}_q(y) = 0 (q = 1,\ 2,\ \ldots,\ Q)$ approximated under the condition (13) or (15), we can synthesize the following switched system which generates a stable nonsmooth limit cycle:

$$\begin{aligned}\dot{y} &= f(y) + g(y)\\ &= \begin{bmatrix} A_q & a_q \\ \Xi_{q_1} & \xi_{q_1} \\ \vdots & \vdots \\ \Xi_{q_{m-2}} & \xi_{q_{m-2}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} + \alpha_q \begin{bmatrix} \mathcal{V}_{q_1} \begin{bmatrix} B_q & b_q \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} \\ \mathcal{V}_{q_2} \\ \vdots \\ \mathcal{V}_{q_{m-1}} \end{bmatrix},\ \text{if } y \in \mathcal{R}_q,\end{aligned} \tag{16}$$

where A_q, $B_q \in \mathbf{R}^{2\times 2}$, a_q, $b_q \in \mathbf{R}^{2\times 1}$, $\Xi_{q_i} \in \mathbf{R}^{1\times 2}$, and $\xi_{q_i} \in \mathbf{R}$ $(i = 1,\ 2,\ \ldots,\ m-2)$. Here A_q, a_q, B_q, and b_q are given such that the following conditions hold:

$$\begin{aligned}\tilde{A}_q \tilde{P}_q + \tilde{P}_q \tilde{A}_q = 0,\\ \tilde{B}_q \tilde{P}_q + \tilde{P}_q \tilde{B}_q < 0,\end{aligned}$$

where $\tilde{A}_q = \begin{bmatrix} A_q & a_q \\ 0 & 0 \end{bmatrix}$ and $\tilde{B}_q = \begin{bmatrix} B_q & b_q \\ 0 & 0 \end{bmatrix}$. We can simply choose these matrices as follows:

$$\tilde{A}_q = \tilde{G}_{A_q} \tilde{P}_q, \quad \tilde{G}_{A_q} = \begin{bmatrix} G_{A_q} & 0 \\ 0 & 0 \end{bmatrix}, \tag{17}$$

where G_{A_q} is an arbitrary skew-symmetric matrix. The matrix $\tilde{B}_q$ can be chosen as follows:

$$\tilde{B}_q = \tilde{G}_{B_q} \tilde{P}_q, \quad \tilde{G}_{B_q} = \begin{bmatrix} G_{B_q} & 0 \\ 0 & 0 \end{bmatrix}, \tag{18}$$

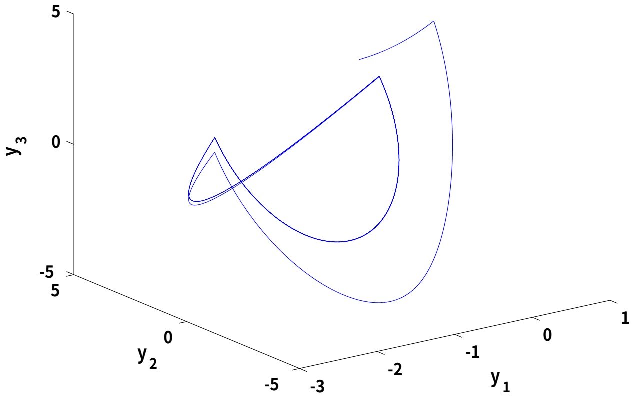

Figure 7: A trajectory of the designed system with reduced “squat” motion.

where G_{B_q} is a matrix which satisfies $G_{B_q}^T + G_{B_q} < 0$. Furthermore, Ξ_{q_i} and ξ_{q_i} are determined by

$$\left[\begin{array}{cc} \Xi_{q_i} & \xi_{q_i} \end{array}\right] = \left[\zeta_{q_i} \quad \eta_{q_i}\right] \left[A_q \quad a_q\right]. \tag{19}$$

Figure 7 shows a trajectory of the designed system corresponding to the reduced “squat” motion $Y_s \in \mathbf{R}^3$. Before we apply the proposed method, we have to partition the motion data Y into several components. In Figure 7, we simply partitioned motion data “squat” into two components. It is clear that the approximation becomes accurate when we split the motion data into a large number of components. The restriction (13) or (15), however, becomes tight.

We summarize the proposed synthesis procedure of nonsmooth motion patterns as follows:

Step1: A given periodic motion data M which forms a closed curve in the n-dimensional space is transformed $M = USV^T$ by the singular value decomposition.

Step2: By choosing an appropriate singular value s_m, we get $M = \Psi Y$, where Y is a reduced motion data of the original one.

Step3: The obtained reduced periodic motion data Y is partitioned into several components (this partition is up to the designer).

Step4: Each ellipsoidal curve Y_q corresponding to a component of Y is approximated by an elliptic cylinder $\mathcal{V}_{q_1}$ and hyperplanes $\mathcal{V}_{q_i}$ $(i = 2, 3, \ldots, n-1)$. Here, $\mathcal{V}_{q_1}$ is chosen such that they are continuous on all region boundaries .

Step5: The desired system which has a periodic motion pattern as an asymptotically stable limit cycle is obtained by (16).

Step6: The sensor data in the n-dimensional space is restored by the reconstruction function Ψ from the reduced motion data space to the original one.

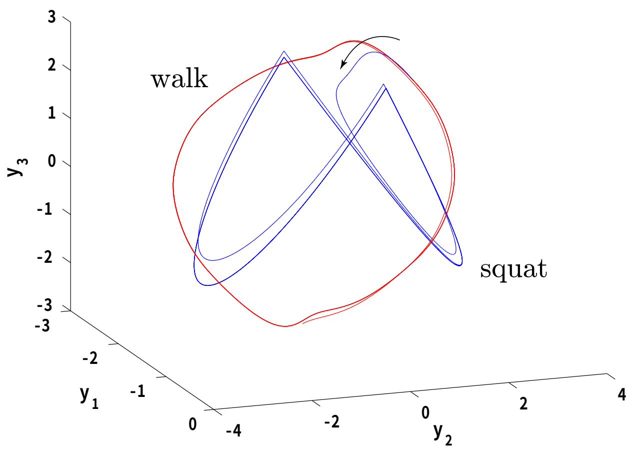

Figure 8: Trajectory in transition from "walk" motion to "squat" motion ($\varepsilon_{\rm ws} = 5$).

Finally, we consider a method for generating a transient behavior when the humanoid robot changes his periodic motions. It is clear that the designed system $\dot{y} = f(y) + g(y)$ can be easily destabilized by replacing $g(x)$ to $-g(x)$. In other words, using a pair of functions $(f,\ g)$ given by Eq.(4) or (16), we can construct the following two systems Σ^+ and Σ^- with a stable and an unstable limit cycle, respectively:

$$\Sigma^+ : \dot{y} = f(y) + g(y), \tag{20}$$

$$\Sigma^- : \dot{y} = f(y) - g(y). \tag{21}$$

By changing dynamics from Eq.(20) to (21), we can make the transition from a stable limit cycle to another one, and there are many methods for the transition. We illustrate a simple method using the transition from "walk" motion to "squat" motion as example. The transient dynamics is described by

$$\dot{y} = w_{\rm ws}(\tau)(f_{\rm w}(y) - g_{\rm w}(y)) + (1 - w_{\rm ws}(\tau))(f_{\rm s}(y) + g_{\rm s}(y)), \tag{22}$$

where the subscript w (*resp.* s) means a function related to "walk" (*resp.* "squat"), and τ is set to be 0 when the transition starts, and $w_{\rm ws}$ is a weight function given by

$$w_{\rm ws}(\tau) = \frac{1}{\exp(\varepsilon_{\rm ws}\tau)}, \tag{23}$$

where $\varepsilon_{\rm ws}$ represents the changing rate of vector fields. When the transition occurs, the trajectory starts to leave from the current motion and converge to the next motion, and after some time elapsed, the vector fields of $\Sigma_{\rm w}^-$ almost disappear and those of $\Sigma_{\rm s}^+$ are dominant. This transient dynamics enable trajectories to transfer between motions smoothly. Figure 8 shows a trajectory in the transition from "walk" motion to "squat" motion, where $\varepsilon_{\rm ws}$ equals 5.

Here, we use a reconstruction function $\Psi_{\rm ws}$ in the transient dynamics is given by an additive combination of two reconstruction functions $\Psi_{\rm w}$ and $\Psi_{\rm s}$ as follows:

$$\Psi_{\rm ws} = w_{\rm ws}(\tau)\Psi_{\rm w} + (1 - w_{\rm ws}(\tau))\Psi_{\rm s}, \tag{24}$$

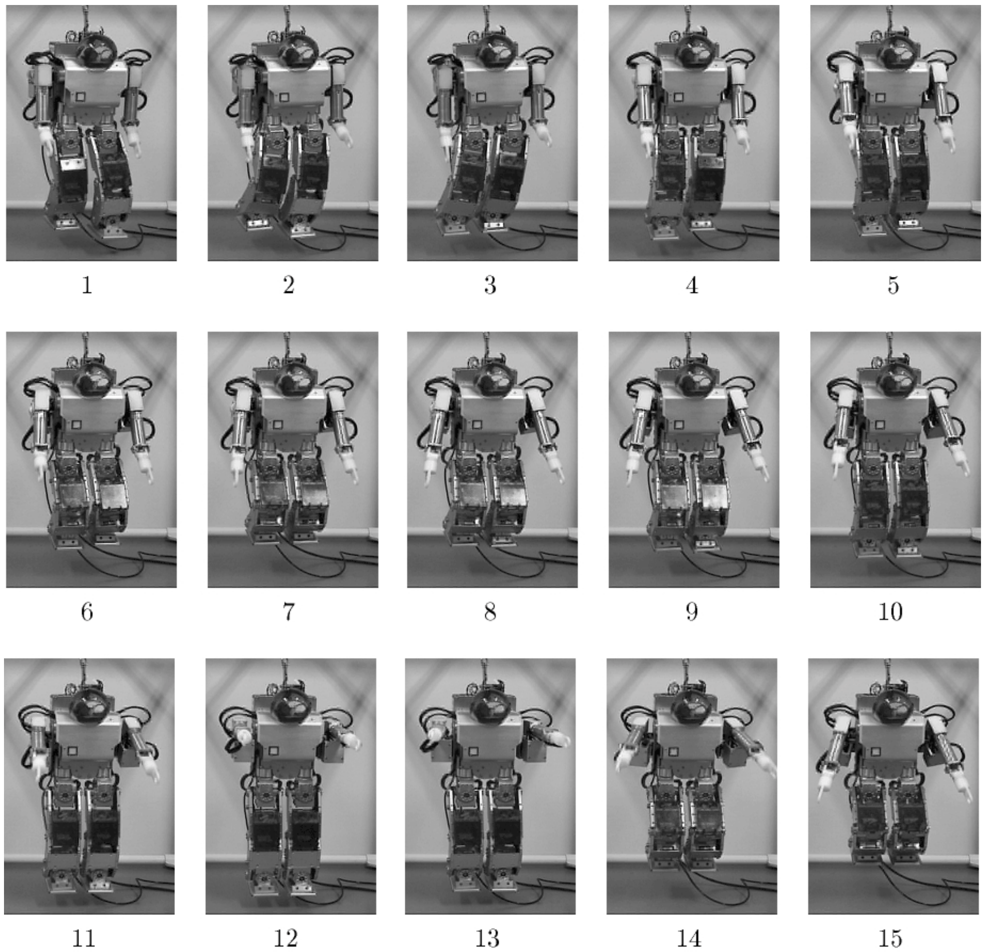

Figure 9: Experiment of transition of periodic motion from “walk” to “squat” by using HOAP-I($\varepsilon_{\mathrm{ws}} = 5$).

Transition of humanoid robot’s motion from “walk” to “squat” given by (24) with $\varepsilon_{\mathrm{ws}} = 5$ is shown in Figure 9.

We propose a new graphical method for representation of transitions between motions which may be periodic or static, called a hybrid state net, in order for humanoid robots to change from a motion to another smoothly. The hybrid state net is an extension of the state net[1], which presents transitions between static positions, in order to present both periodic motions and static positions in a unified way. Here, for simplicity, we deal with only smooth periodic motions, but it is easy to extend it to nonsmooth ones. Formally, the hybrid state net is described by 9-tuples $(Q,\ X,\ D,\ R,\ L_t,\ \Phi,\ A,\ G,\ A_s)$ and its graphical representation is shown in Figure 10, where each symbol in the hybrid state net is summarized in Table 1.

$Q = \{q_1,\ q_2,\ \cdots\}$ is a set of nodes which corresponds to periodic motions of humanoid robots. $X = \{X_1,\ X_2,\ \cdots\}$ is a set of continuous variables, and the continuous variable X_i at node q_i is described by $X_i = [\ x_i^T \quad y^T\ ]^T$, where $x_i = [\ x_{i1} \quad x_{i2}\ ]^T$ represents a phase of node q_i and $y = [\ y_1 \quad y_2 \quad \cdots \quad y_r\ ]^T$ is a r-dimensional vector which

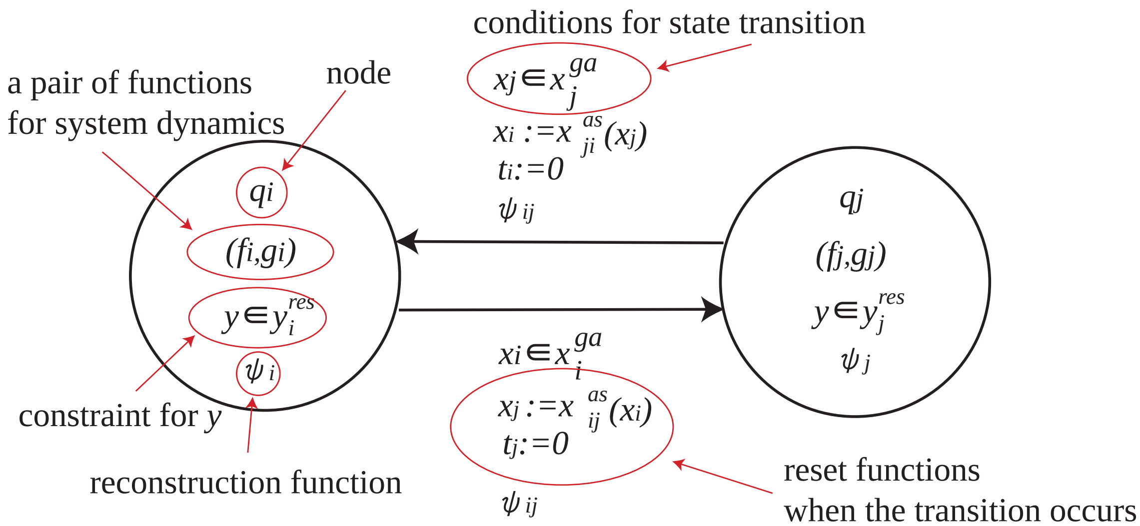

Figure 10: Illustration of hybrid state net.

Table 1: Meaning of each symbol in hybrid state net

Q	Set of nodes
X	Set of continuous variables
D	Set of a pair of functions $(f_i,\ g_i)$ by which the vector field in node q_i is described
R	Set of a constraint condition for y at each node
L_T	Set of local time at each node
Φ	Set of reconstruction functions at each node or arc
A	Set of directed arcs between nodes
G	Set of phase conditions on transition at each directed arc
A_s	Set of functions which calculates a phase at the new node after transition

represents a reduced data. D is a set of a pair $(f_i,\ g_i)$ of functions by which the vector field at node q_i is given, and a reference trajectory of the humanoid robot is generated by the vector field. $R = \{y_1^{res},\ y_2^{res},\ \cdots\}$ is a set of subsets $y_i^{res} \in \mathbf{R}^r$ which indicates a constraint condition for the r-dimensional vector y at node q_i such that $y \in y_i^{res}$ holds whenever the humanoid robot exhibits the motion corresponding to node q_i. Note that y_i^{res} is obtained by a projection of a feasible region in the n-dimensional sensory space to the r-dimensional reduced space. $L_T = \{t_1,\ t_2,\ \cdots\}$ is a set of local time and t_i is local time at node q_i. $A = \{a_{12},\ a_{21},\ \cdots\}$ is a set of directed arcs, where a_{12} is a directed arc from node q_i to node q_j, and each arc indicates a transient motion. It takes ϵ_{ij} time to complete the transition from q_i to q_j. $\Phi = \{\Phi_1,\ \Phi_2,\ \cdots,\ \Phi_{12},\ \cdots\}$ is a set of reconstruction functions from the reduced vector y to its reconstructed data in n-dimensional sensory space, where Φ_i(*resp.* Φ_{ij}) is a reconstruction function at node q_i(*resp.* directed arc a_{ij}). x_{ij}^{gu} represents a phase condition under which transition from node q_i to node q_j is enabled. If $x_i \notin x_{ij}^{gu}$, then the transition is disabled. Let G be a set of the phase conditions x_{ij}^{gu}. x_{ij}^{as} is a function which determines the initial phase at node q_j after the transition corresponding to arc a_{ij} occurs, and let A_s be a set of all the functions x_{ij}^{as}.

We will show a procedure for the transition from node q_i to q_j based on the hybrid state net. We construct the following dynamical system with a stable limit cycle at node q_i from a pair of functions $(f_i,\ g_i)$:

$$\Sigma^i : \dot{X}_i = f_i(X_i) + g_i(X_i), \tag{25}$$

where

$$X_i = \begin{bmatrix} x_i \\ y \end{bmatrix}.$$

Now, suppose that the command " start the transition from q_i to q_j" is given when local time t_i of node q_i equals t_{i1} as shown in Figure 11. If the phase $x_i(t_{i1})$ satisfies $x_i \notin x_{ij}^{gu}$, then the transition starts at local time $t_i = t_{i2}$ when $x_i(t_{i2}) \in x_{ij}^{gu}$. When the transition

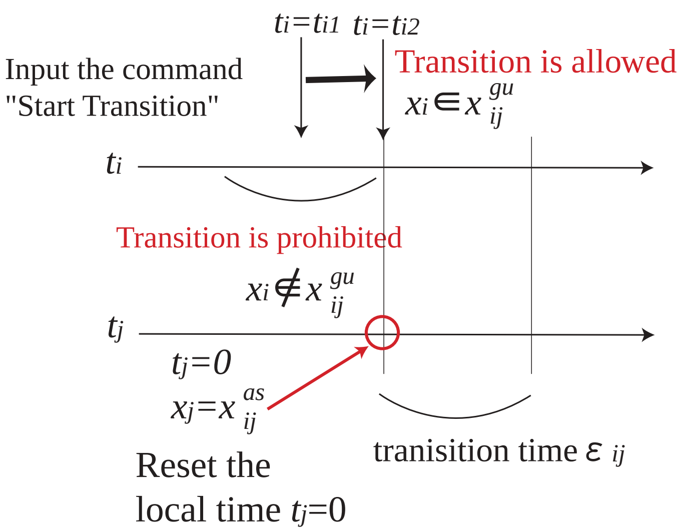

Figure 11: Illustration of state transition

starts, we introduce new continuous variables:

$$X^i = \begin{bmatrix} x_i \\ y^i \end{bmatrix}, \quad X^j = \begin{bmatrix} x_j \\ y^j \end{bmatrix},$$

and the local time t_j at node q_j is set to be zero, and we initialize the continuous variables such as $x_j(0) = x_{ij}^{as}$, and $y^i(t_{i2}) = y^j(0) = y(t_{i2})$. Moreover, we use the following two dynamical systems in order to change the steady state from the limit cycle at node q_i to that at node q_j:

$$\acute{\Sigma}^i : \dot{X}^i = f_i(X^i) - g_i(X^i), \tag{26}$$

$$\acute{\Sigma}^j : \dot{X}^j = f_j(X^j) + g_j(X^j), \tag{27}$$

Note that the limit cycle at node q_i is destabilized by the dynamical system $\acute{\Sigma}^i$, and that at node q_j is stabilized by the dynamical system $\acute{\Sigma}^j$. For simplicity, using a weight function $w_{ij}(t_j)$, a reconstruction function Ψ_{ij} during the transition from q_i to q_j is given as follows:

$$\Psi_{ij}(y^i(t_i),\ y^j(t_j)) = \omega_{ij}(t_j)\Psi_i(y^i(t_i)) + (1 - \omega_{ij}(t_j))\Psi_j(y^j(t_j)). \tag{28}$$

The transition ends at $t_j = \epsilon_{ij}$, and, by setting $y(\epsilon_{ij}) = y^j(\epsilon_{ij})$, the humanoid robot exhibits a periodic motion corresponding to the limit cycle in the dynamical system Σ^j at node q_j. Note that w_{ij} is determined such that the ZMP of the humanoid robot is always stable during the transition. In experiment using the humanoid robot HOAP-I, we constructed a hybrid state net shown in Figure 12, consisting of four primitive motions: "static position", "walk", "squat", and "footing". In the figure, the nodes $q_o,\ q_w,\ q_s$, and q_l mean "static position", "walk", "squat", and "footing", respectively. It is shown from the figure that a direct transition from "walk" to "footing" is possible but its reverse transition is impossible. So, in order to change from "footing" to "walk", we have to change "walk" to "static position" first, and "static position" to "walk" immediately. Thus, hybrid state net shows possibility of transitions. Moreover, when we find a new feasible transient trajectory, all we have to do is to insert a new arc corresponding to the trajectory. So, the hybrid state net is suitable for learning motions.

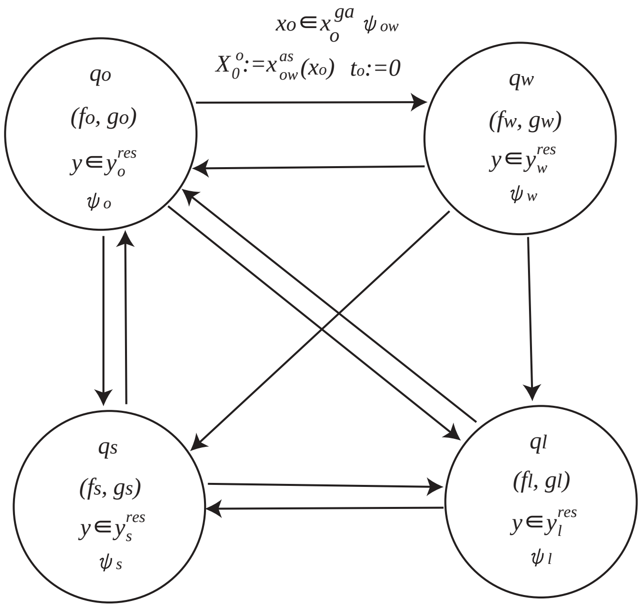

Figure 12: Hybrid state net consisting of four primitive motion: "static position", "walk", "squat", and "footing"

(2) Results and their importance

In this study, we proposed a dynamics based presentation of periodic motions of humanoid robots using a Lyapunov function based method for construction of nonlinear dynamical systems with specified limit cycles. First, we collect periodic sequence data in a sensory space of several primitive motions with a cut-and-try approach. Next, the data is reduced to a lower-dimensional data using the singular decomposition method. Finally, we construct a nonlinear dynamical system on the reduced space where the reduced periodic motion is represented by a stable limit cycle. In our experiment, the periodic data was reduced to three-dimensional data so that the proposed method was used effectively. Then, we introduced a hybrid state net, which is an extension of the state net, in order to present transitions between the primitive motions.

We also proposed a supervisory control system for motion planning of humanoid robots. The proposed system shown in Fig. 13 is hierarchically structured into two levels. The lower level controls and monitors the robots using modular state nets. A modular state net is a state net representing motions of parts of the robots such as arms, legs, and so on, and whole body motions of the robots are represented by a combination of modular state nets for the parts. The upper level generates an optimal sequence of motions for user's requirements using timed Petri nets. A timed Petri net is used as an abstracted model of the set of all modular state nets, and using discrete event systems theory and optimal path searching algorithms, we find an optimal motion sequence. Moreover, applying a reinforcement learning method, we proposed a novel approach to the fast convergence to an optimal supervisor.

Using the proposed methods, we can store many primitive motions of humanoid robots which lead to realization of very complicated behavior.

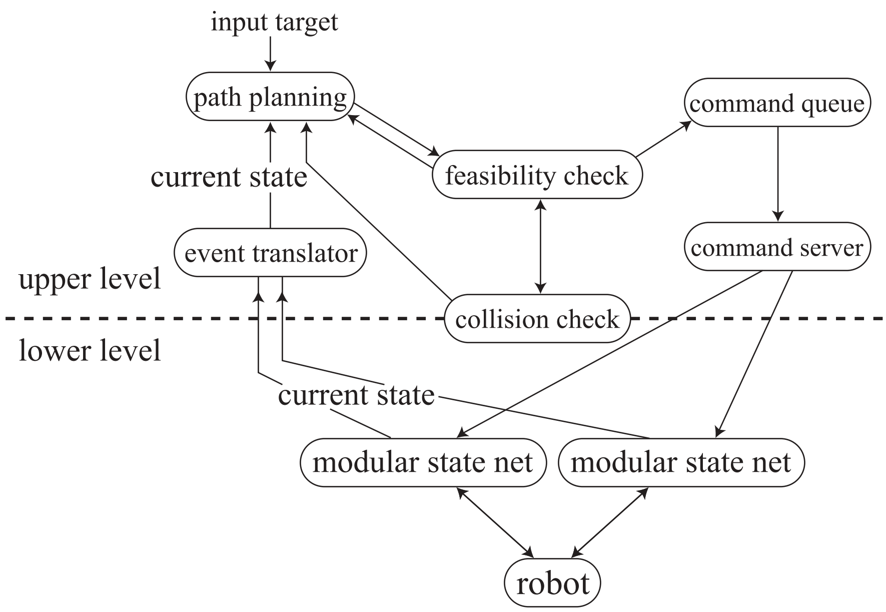

Figure 13: Hierarchical supervisory control system with two levels for motion planning

References

[1] F. Kanehiro, M. Inaba, H. Inoue, and S. Hirai: Developmental Realization of Whole-Body Humanoid Behaviors Based on State Net Architecture Containing Error Re-

covery Functions, Proc. of the First IEEE-RAS International Conference on Humanoid Robots, 2000.

[2] M. Okada, K. Tatani, and Y. Nakamura: Polynomial Design of the Nonlinear Dynamics for the Brain-Like Information Processing of Whole Body Motion, Proc. of IEEE International Conference on Robotics and Automation, pp. 1410–1415, 2002.

[3] D. N. Green: Synthesis of Systems with Periodic Solutions Satisfying $V(x) = 0$, IEEE Transactions on Circuits and Systems, Vol. 31, No. 4, pp. 317–326, 1984.

[4] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel (eds.), Hybrid Systems, Lecture Notes in Computer Science, Vol. 736, Springer Verlag, 1993.

[5] M. Johansson, A. Rantzer: Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems, IEEE Transactions on Automatic Control, Vol. 43, No. 4, pp. 555–559, 1998.