X

Biologically Inspired Robot Arm Control Using Neural Oscillators

Woosung Yang[1], Nak Young Chong[2] and Bum Jae You[1]
[1]Korea Institute of Science and Technology, Korea
[2]Japan Advanced Institute of Science and Technology, Japan

1. Introduction

Humans or animals exhibit natural adaptive motions against unexpected disturbances or environment changes. This is because that, in general, the neural oscillator based circuits on the spinal cord known as Central Pattern Generators (CPGs) might contribute to efficient motor movement and novel stability properties in biological motions of animal and human. Based on the CPGs, most animals locomote stably using inherent rhythmic movements adapted to the natural frequency of their body dynamics in spite of differences in their sensors and actuators.

For such reasons, studies on human-like movement of robot arms have been paid increasing attention. In particular, human rhythmic movements such as turning a steering wheel, rotating a crank, etc. are self-organized through the interaction of the musculoskeletal system and neural oscillators. In the musculoskeletal system, limb segments connected to each other with tendons are activated like a mechanical spring by neural signals. Thus neural oscillators may offer a reliable and cost efficient solution for rhythmic movement of robot arms. Incorporating a network of neural oscillators, we expect to realize human nervous and musculoskeletal systems in various types of robots.

The mathematical description of a neural oscillator was presented in Matsuoka's works (Matsuoka, 1985). He proved that neurons generate the rhythmic patterned output and analyzed the conditions necessary for the steady state oscillations. He also investigated the mutual inhibition networks to control the frequency and pattern (Matsuoka, 1987), but did not include the effect of the feedback on the neural oscillator performance. Employing Matsuoka's neural oscillator model, Taga *et al*. investigated the sensory signal from the joint angles of a biped robot as feedback signals (Taga et al., 1991), showing that neural oscillators made the robot robust to the perturbation through entrainment (Taga, 1995). This approach was applied later to various locomotion systems (Miyakoshi et al., 1998), (Fukuoka et al., 2003), (Endo et al., 2005), (Yang et al., 2008).

Besides the examples of locomotion, various efforts have been made to strengthen the capability of robots from biological inspiration. Williamson created a humanoid arm motion based on postural primitives. The spring-like joint actuators allowed the arm to safely deal with unexpected collisions sustaining cyclic motions (Williamson, 1996). And the neuro-mechanical system coupled with the neural oscillator for controlling rhythmic arm motions was proposed (Williamson, 1998). Arsenio suggested the multiple-input describing function technique to control multivariable systems connected to multiple neural oscillators (Arsenio, 2000).

Even though natural adaptive motions were accomplished by the coupling between the arm joints and neural oscillators, the correctness of the desired motion was not guaranteed. Specifically, robot arms are required to exhibit complex behaviors or to trace a trajectory for certain type of tasks, where the substantial difficulty of parameter tuning emerges. The authors have presented encouraging simulation results in controlling the arm trajectory incorporating neural oscillators (Yang et al., 2007 & 2008). This chapter addresses how to control the trajectory of a real robot arm whose joints are coupled to neural oscillators for a desired task. For achieving this, real-time feedback from sensory information is implemented to exploit the entrainment feature of neural oscillators against unknown disturbances.

In the following section, a neural controller is briefly explained. An optimization procedure is described in Section 3 to design the parameters of the neural oscillator for a desired task. Details of dynamic responses and simulation and experimental verification of the proposed method are discussed in Section 4 and 5, respectively. Finally, conclusions are drawn in Section 6.

2. Rhythmic Movement Using a Neural Oscillator

2.1 Matsuoka’s neural oscillator

Our work is motivated by studies and facts of biologically inspired locomotion control employing oscillators. Especially, the basic motor pattern generated by the CPG of inner body of human or animal is usually modified by sensory signals from motor information to deal with environmental disturbances. The CPGs drive the antagonistic muscles that are reciprocally innervated to form an intrinsic rhythm generating mechanism around each joint. Hence, adapting this mechanism actuated by the CPGs which consists of neural oscillator network, we can design a new type of biologically inspired robots that can accommodate unknown interactions with the environments by controlling internal loading (or force) of the body.

For implementing this, we use Matsuoka’s neural oscillator consisting of two simulated neurons arranged in mutual inhibition as shown in Fig. 1. If gains are properly tuned, the system exhibits limit cycle behaviors. Now we propose the control method for dynamic systems that closely interacts with the environment exploiting the natural dynamics of Matsuoka’s oscillator.

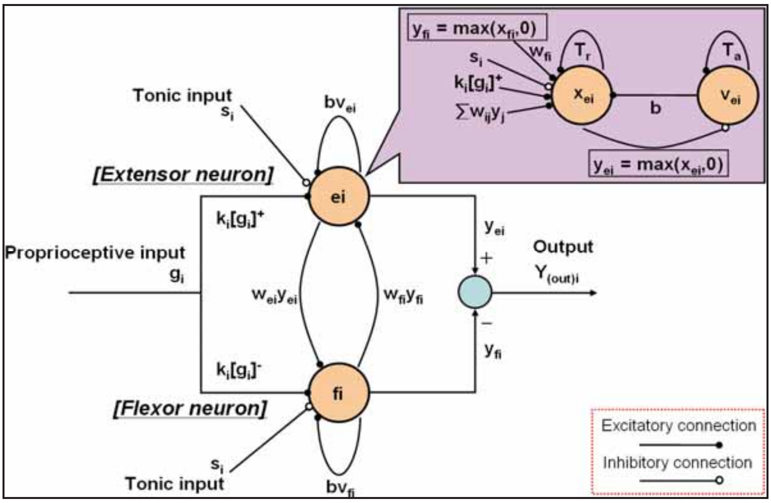


Fig. 1. Schematic diagram of Matsuoka Neural Oscillator

$$
\begin{aligned}
T_r\dot{x}_{ei} + x_{ei} &= -w_{fi}y_{fi} - \sum_{j=1}^{n} w_{ij}y_j - bv_{ei} - \sum k_i[g_i]^+ + s_i \\
T_a\dot{v}_{ei} + v_{ei} &= y_{ei} \\
y_{ei} &= [x_{ei}]^+ = \max(x_{ei}\,,\ 0) \\
T_r\dot{x}_{fi} + x_{fi} &= -w_{ei}y_{ei} - \sum_{j=1}^{n} w_{ij}y_j - bv_{fi} - \sum k_i[g_i]^- + s_i \\
T_a\dot{v}_{fi} + v_{fi} &= y_{fi} \\
y_{fi} &= [x_{fi}]^+ = \max(x_{fi}\,,\ 0)\,,\ \ (i = 1,\ 2,\ \cdots,\ n)
\end{aligned}
\tag{1}
$$

where x_{ei} and x_{fi} indicate the inner state of the i-th neuron for i=1~n, which represents the firing rate. Here, the subscripts 'e' and 'f' denote the extensor and flexor neurons, respectively. $v_{e(f)i}$ represents the degree of adaptation and b is the adaptation constant or self-inhibition effect of the i-th neuron. The output of each neuron $y_{e(f)i}$ is taken as the positive part of x_i and the output of the oscillator is the difference in the output between the extensor and flexor neurons. w_{ij} is a connecting weight from the j-th neuron to the i-th neuron: w_{ij} are 0 for $i \neq j$ and 1 for $i=j$. $w_{ij}y_i$ represents the total input from the neurons arranged to excite one neuron and to inhibit the other, respectively. Those inputs are scaled by the gain k_i. T_r and T_a are the time constants of the inner state and the adaptation effect, respectively, and s_i is an external input with a constant rate. $w_{e(f)i}$ is a weight of the extensor neuron or the flexor neuron and g_i indicates a sensory input from the coupled system.

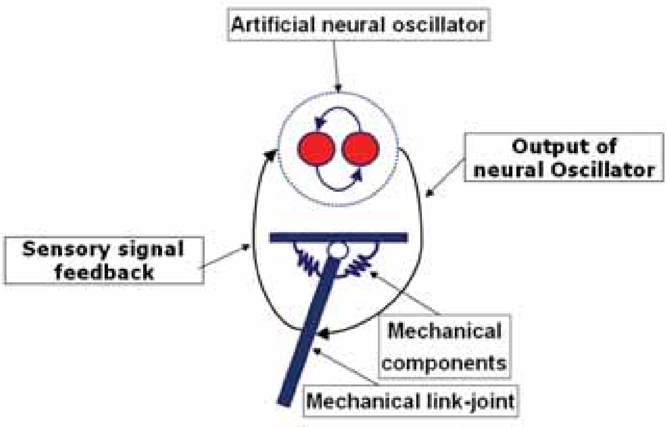


Fig. 2. Mechanical system coupled to the neural oscillator

Figure 2 conceptually shows the control method exploiting the natural dynamics of the oscillator coupled to the dynamic system that closely interacts with environments. This method enables a robot to adapt to changing conditions. For simplicity, we employ a general 2nd order mechanical system connected to the neural oscillator as seen in Fig. 4. The desired torque signal to the joint can be given by

$$\tau_i = k_i(\theta_{vi} - \theta_i) - b_i\dot{\theta}_i, \tag{2}$$

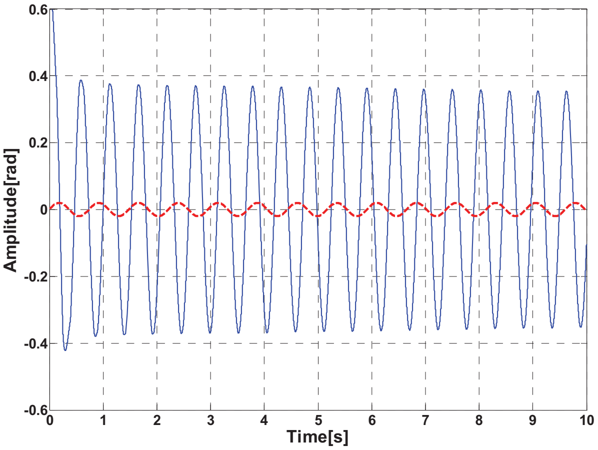
where k_i is the stiffness of the joint, b_i the damping coefficient, θ_i the joint angle, and θ_{vi} is the output of the neural oscillator that produces rhythmic commands of the *i*-th joint. The neural oscillator follows the sensory signal from the joints, thus the output of the neural oscillator may change corresponding to the sensory input. This is what is called "entrainment" that can be considered as the tracking of sensory feedback signals so that the mechanical system can exhibit adaptive behavior interacting with the environment.

2.2 Entrainment property of the neural oscillator

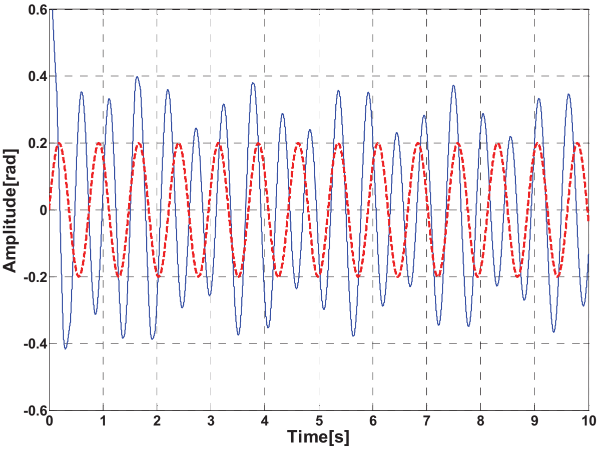
Generally, it has been known that the Matsuoka's neural oscillator exhibits the following properties: the natural frequency of the output signal increases in proportion to $1/T_r$. The magnitude of the output signal also increases as the tonic input increases. T_r and T_a have an effect on the control of the delay time and the adaptation time of the entrained signal, respectively. Thus, as these parameters decrease, the input signal is well entrained. And the minimum gain k_i of the input signal enlarges the entrainment capability, because the minimum input signal is needed to be entrained appropriately in the range of the natural frequency of an input signal. In this case, regardless of the generated natural frequency of the neural oscillator and the natural frequency of an input signal, the output signal of the neural oscillator locks onto an input signal well in a wide range.

Figure 3 illustrates the entrainment procedure of the neural oscillator. If we properly tune the parameters of the neural oscillator, the oscillator exhibits the stable limit cycle behaviors.

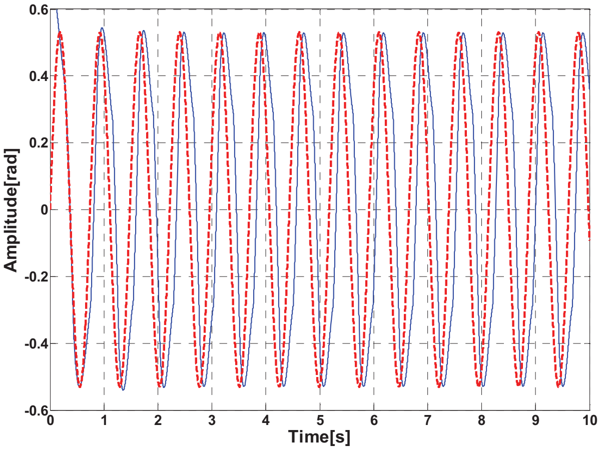
In Figure 1, the gain *k* of the sensory feedback was sequentially set as 0.02, 0.2 and 0.53 such as Figure 3 (a), (b) and (c). When *k* is 0.02, the output of the neural oscillator can't entrain the sensory signal input as shown in Figure 3 (a). The result of Figure 3 (b) indicates the signal partially entrained. If the gain *k* is properly set as 0.53, the neural oscillator produces the fully entrained signal as illustrated in Figure 3 (c) in contrast to the result of Figure 3 (b).



(a)



(b)



(c)

Fig. 3. Simulation results on the entrainment property of the neural oscillator. The solid line is the output of the neural oscillator and the dashed line indicates the sensory signal input.

3. Optimization of Neural Oscillator Parameters

The neural oscillator is a non-linear system, thus it is generally difficult to analyze the dynamic system when the oscillator is connected to it. Therefore a graphical approach known as the describing function analysis has been proposed earlier (Slotine & Li, 1991). The main idea is to plot the system response in the complex plane and find the intersection points between two Nyquist plots of the dynamic system and the neural oscillator. The intersection points indicate limit cycle solutions. However, even if a rhythmic motion of the dynamic system is generated by the neural oscillator, it is usually difficult to obtain the desired motion required by the task. This is because many oscillator parameters need to be tuned, and different responses occur according to the inter-oscillator network. Hence, we describe below how to determine the parameters of the neural oscillator using the *Metropolis* method (Yang et al., 2007 & 2008) based on simulated annealing (SA) (Kirkpatrick, 1983), which guarantees convergence to the global extremum (Geman & Geman, 1984).

For the process of minimizing some cost function E, $X=[T_r, T_a, w, s, \cdots]^T$ is selected as the parameters of the neural oscillator to be optimized; the initial temperature T_0 is the starting parameter; the learning rate ν is the step size for X. Specifically, the parameters are replaced by a random number N in the range [-1,1] given by;

$$X_i = X_{i-1} + \nu \cdot N \tag{3}$$

If the change in the cost function ΔE is less than zero, the new state X_i is accepted and stored at the i-th iteration. Otherwise, another state is drawn with the transition probability, $Prob_i(E)$ given by

$$\Pr ob_i(E) = (\frac{1}{Z(T)})\exp(-\frac{\Delta E}{c}) > \gamma, \tag{4}$$

where γ is a random value uniformly distributed between 0 and 1. The temperature cooling schedule is $c_i = k \cdot c_{i-1}$ (k is the Boltzmann constant or effective annealing gain) and $Z(T)$ is a temperature-dependant normalization factor. If ΔE is positive and $Prob_i(E)$ is less than γ or equal to zero, the new state X_i is rejected. Here the lower cost function value and large difference of ΔE indicate that X_i is the better solution. If temperature approaches zero, the optimization process terminates.

Even though SA has several potential advantages over conventional algorithms, it may be faced with a crucial problem. When searching for optimal parameters, it is not known whether the desired task is performed correctly with the selected parameters or not. We therefore added the task completion judgment and cost function comparison steps as shown in Fig. 4 by thick-lined boxes. If the desired task fails, the algorithm reloads previously stored parameters and selects the parameters that give the lowest cost function value. Then the optimization process is restarted with the selected parameters until it finds the parameters of the lowest cost function that allow the task to be done correctly.

4. Crank Rotation of Two-link Planar Arm

To validate the proposed control scheme, we evaluate the crank rotation task with a two-link planar arm whose joints are coupled to neural oscillators as shown in Fig. 5. The inter-oscillator network is not established, because the initial condition of the same sign will be equivalent to the excitatory connection between two oscillators. We focus on the entrainment property of the arm.

The crank rotation is modeled by generating kinematic constraints and an appropriate end-effector force. The crank has the moment of inertia I and the viscous friction at the joint connecting the crank and the base. If the arm end-effector position is defined as (x, y) in a Cartesian coordinate system whose origin is at the center of the crank denoted as (x_0, y_0), the coordinates x and y can be expressed as

$$\begin{gathered}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -r\sin\varphi + x_0 \\ r\cos\varphi + y_0 \end{pmatrix} = \begin{pmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{pmatrix} \\ \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} r\sin\varphi\dot{\varphi} - r\cos\varphi\ddot{\varphi} \\ -r\cos\varphi\dot{\varphi} - r\sin\varphi\ddot{\varphi} \end{pmatrix} = \frac{\partial J(\theta)^2}{\partial^2\theta}, \end{gathered} \tag{5}$$

where J is the Jacobian matrix of $[x, y]^\mathrm{T}$. φ and θ_i are the crank angle and the i-th joint angle, respectively. l_i is the length of the i-th link. c_1, c_{12}, s_1 and s_{12} denote $\cos\theta_1$, $\cos(\theta_1 + \theta_2)$, $\sin\theta_1$ and $\sin(\theta_1 + \theta_2)$, respectively. r is the radius of the crank. Eq. (5) can be rearranged as follows:

$$J(\theta)\ddot{\theta} + \dot{J}(\theta, \dot{\theta})\dot{\theta} = r(u(\phi)\ddot{\phi} - v(\phi)\dot{\phi}^2), \tag{6}$$

where u is the tangential unit vector and v is the normal unit vector at the outline of the crank as shown in Fig. 5, respectively.

Now the dynamic equations of the crank and the arm are given in the following form.

$$I\ddot{\varphi} + C\dot{\varphi} = ru(\varphi)^T F \tag{7}$$

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) = \tau - J(\theta)^T F \tag{8}$$

$$\tau = k(\theta_v - \theta) - b\dot{\theta}, \tag{9}$$

where M is the inertia matrix, V is the Coriolis/centripetal vector, and G is the gravity vector, k and b denotes the joint stiffness and viscosity matrixes, respectively (Gomi & Osu, 1998), θ_v is the output of the neural oscillator (see Eq. (2)), F is the contact force vector interacting between the crank and the end-efector. By solving Eqs. (7) and (8) simultaneously using Eq. (6), F is obtained as

$$F = \{J(\theta)M(\theta)^{-1}J(\theta)^T + r^2I^{-1}u(\phi)u(\phi)^T\}^{-1}\{J(\theta)M(\theta)^{-1}(\tau - V(\theta)) + \dot{J}(\theta,\dot{\theta})\dot{\theta} + r\dot{\phi}(v(\phi)\dot{\phi} + CI^{-1}u(\phi))\} \tag{10}$$

It is very hard to properly tune parameters of the neural oscillator for attaining the desired rotation task. Moreover, this dynamic model is tightly coupled to crank dynamics as described in Eq. (10). Thus, the proposed parameter tuning approach is divided into the following two steps:

1) Step 1: Find initial parameters of the neural oscillator corresponding to desired inputs of each joint using the cost function given by:

$$\phi = \left|\frac{T - T_G}{T_G}\right| + \nu \cdot \max\left(\frac{|A_d - C|}{B} - 1, 0\right) \tag{11}$$

subject to

$i)$ $A_{\min} \le A_d \le A_{\max}$

$ii)$ $|A_d - C| \le B$

where C=($A_{\max}$+$A_{\min}$)/2, B=($A_{\max}$-$A_{\min}$)/2; A_d is the desired amplitude of the neural oscillator for the rotation task, $A_{\max}$ and $A_{\min}$ are the maximum and minimum amplitude constraints, respectively; T and T_G denote the desired and measured natural frequencies of the output generated by the neural oscillator, respectively. v is the performance gain.

2) Step 2: Using the initial parameters obtained by Step1, run the proposed SA algorithm as illustrated in Fig. 4. The cost function for the crank rotation includes the velocity of the rotation, torque, and consumed energy.

Implementing Step 1 and Step 2 in sequence, we are able to acquire the appropriate initial and tuned parameters as seen in Table 1. Figure 6 (a) indicates a cooling state in terms of cooling schedule. Cooling or annealing gain K is set as 0.95. It can be observed in Fig. 6 (b) that the optimal process was well operated and a better solution at the lowest cost function was obtained iteratively. As expected, when the tuned parameters are employed to perform the given task, a stable motion could be accomplished as shown in Fig. 6. It is evident in Fig.

6 (c) that initial transient responses disappear due to the entrainment property of the neural oscillator. This property enables the arm to sustain the given task against changes in parameters of arm kinematics and dynamics as well as disturbances.

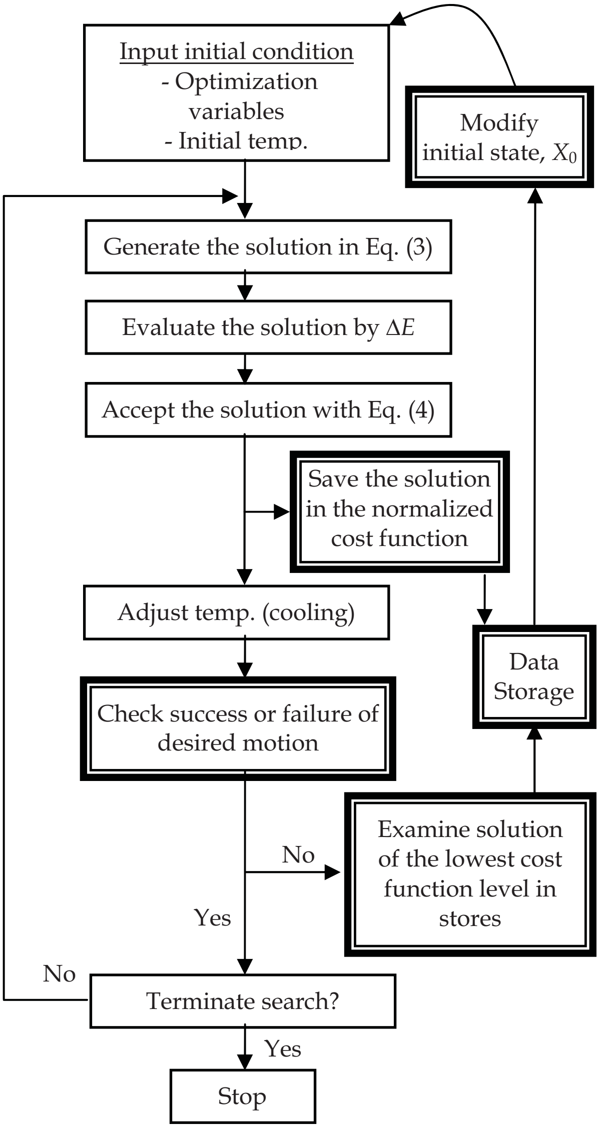
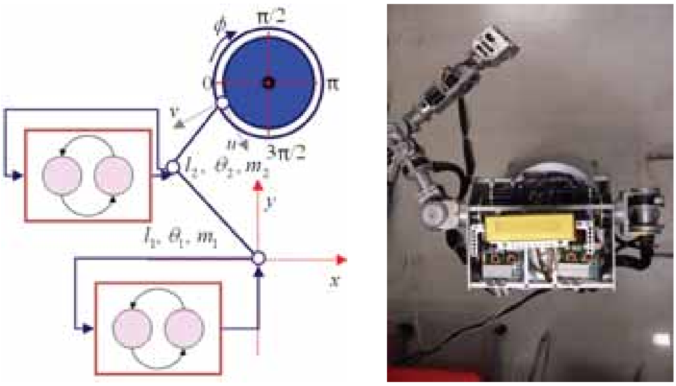
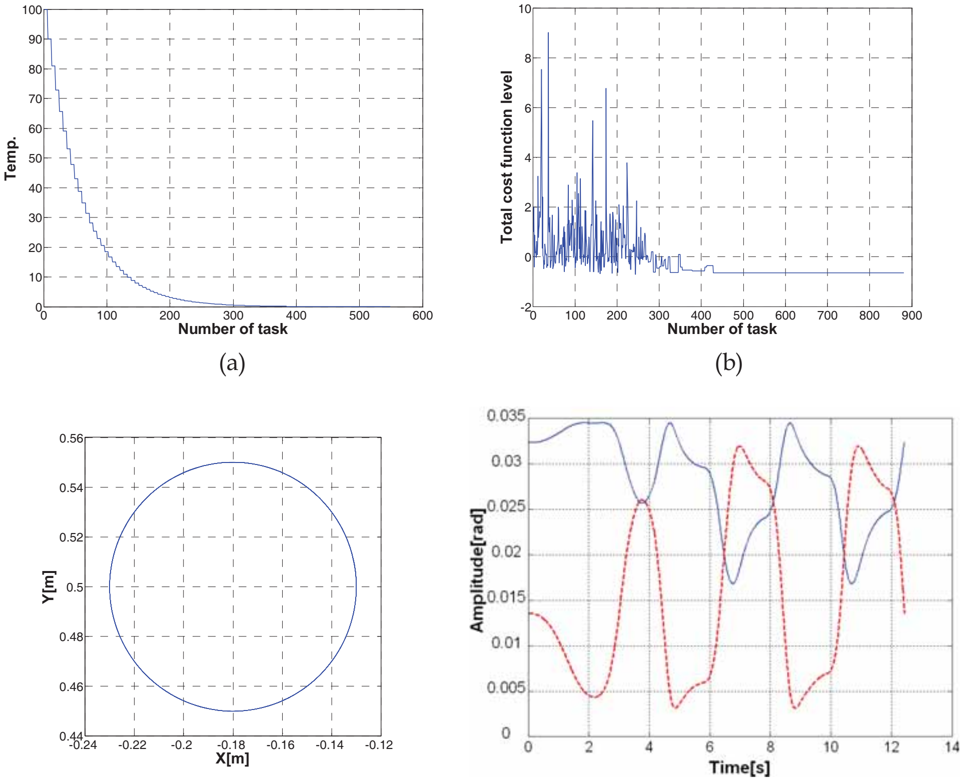

Input initial condition
- Optimization variables
- Initial temp.
Modify initial state, X_0
Generate the solution in Eq. (3)
Evaluate the solution by ΔE
Accept the solution with Eq. (4)
Save the solution in the normalized cost function
Adjust temp. (cooling)
Data Storage
Check success or failure of desired motion
No
Examine solution of the lowest cost function level in stores
Yes
No
Terminate search?
Yes
Stop

Fig. 4. Flowchart of the upgraded SA for task based parameter optimization.



(a) (b)

Fig. 5. (a) Schematic robot arm model and (b) real robot arm coupled with the neural oscillator for experimental test



(a) (b)

(c) (d)

Fig. 6. (a) Temperature transition for cooling schedule, (b) A transition of total cost function level, (c) The end-effector trajectory of two-link arm (d) The output of joint angle. The red dash line is the first joint angle and the second joint angle is drawn by the blue thin line

Initial parameters		**Optimized parameters**	
Inhibitory weight (w)	2.0	Inhibitory weight (w)	4.012
Time constant (T_r)	0.25	Time constant (T_r)	1.601
(T_a)	0.5	(T_a)	3.210
Sensory gain (k)	1	Sensory gain (k)	10.010
Tonic input (s)	60	Tonic input (s)	57.358
Robot Arm Model			
Mass 1 (m_1),	Mass 2 (m_2)	2.347kg,	0.834kg
Inertia 1 (I_1),	Inertia 2 (I_2)	0.0098kgm^2,	0.0035kgm^2
Length 1 (l_1),	Length 2 (I_2)	0.224m,	0.225m

Table 1. Initial and tuned parameters of the neural oscillator with robot arm model

5. Experiments with a Real Robot Arm

To validate the proposed control scheme described in Section 4, we employed a real robot arm with 6 degrees of freedom (see Fig. 5 (b)) and constructed a real time control system. This arm controller runs at 200 Hz and is connected via IEEE 1394 for data transmission at 4 kHz. ATI industrial automation's Mini40 sensor was fitted to the wrist joint of the arm to detect external disturbances. The optimized parameters in Table 1 were used for the neural oscillator.

Figure 7 shows the arm kinematics. Since the crank motion is generated in the horizontal plane, $q_1$and q_3 are set to 90°. The initial values of q_5 and q_6 are set to 0°, respectively. q_2 and q_4, corresponding to θ_1 and θ_2 in Fig. 5 (a), respectively, are controlled by the neural oscillators and the constraint force given in Eq. (10). The constraint force enables the end-effector to trace the outline of the (virtual) crank. Hence, the end-effector can draw the circles as shown in Fig. 8 (see the overlapping circles in the center part of the figure).

Now, we will examine what happens in the arm motion if additive external disturbances exist. Arbitrary forces are applied to the end-effector at 15s, 28s, 44s, 57s, 73s and 89s sequentially as shown in Fig. 9. We first pushed the end-effector along the minus x direction. The force sensor value in the x and y direction are added to Eq. (10). Then, the joint angles change according to the direction of the applied force, which makes the neural oscillators entrain the joint angles as shown in Fig. 10. The solid line is the output of the neural oscillator connected to the first joint (q_2) and the dashed line indicates that of the neural oscillator connected to the second one (q_4). Hence a change in the output of the neural oscillator causes a change in the joint torque. Finally the joint angles are modified as shown in Fig. 11, where the bottom plot is the output of q_2 and the top one is the output of q_4. Fig. 12 shows the snap shots of the simulated crank motion by the robot arm, where we can observe that the end-effector traces the circle well, and adapts its motion when an external force is applied to it.

Table 2 compares the power consumption of the robot arm performing the above task with different parameters of the neural oscillator. The parameters were drawn arbitrary among the ones that guarantee a successful completion of the task. If the optimized parameters (set D) were employed, the most energy-efficient motion was realized.

	Parameter set A	Parameter set B	Parameter set C	Parameter set D (optimized)
Inhibitory weight (w)	2.0	2.503	4.012	**4.012**
Time constant	0.25	0.896	1.601	**1.601**
(T_r)	0.5	5.0	3.210	**3.210**
(T_a)	1.0	1.241	15.010	**10.010**
Sensory gain (k)	60.0	60.660	57.358	**57.358**
Tonic input (s)				
Measured current [A]	1.871	0.794	0.591	**0.572**
Power [W] consumption	89.808	38.112	28.368	**27.456**

Table 2. Power Consumption according to the selected parameter set of the neural oscillator

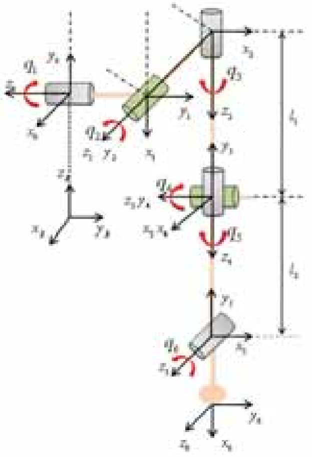


Fig. 7. Kinematic parameters of the robot arm

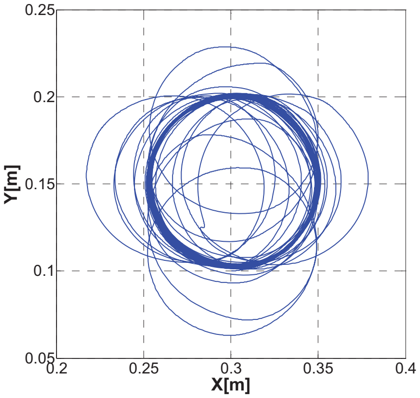


Fig. 8. The trajectory drawn by the end-effector of the arm

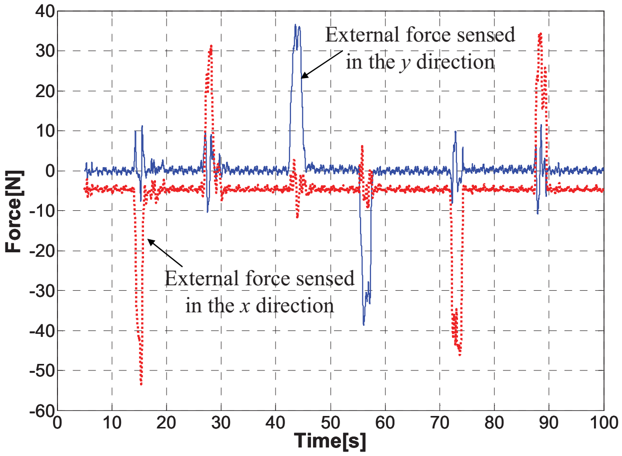


Fig. 9. The external forces measured by the force sensor in the *x* and *y* direction

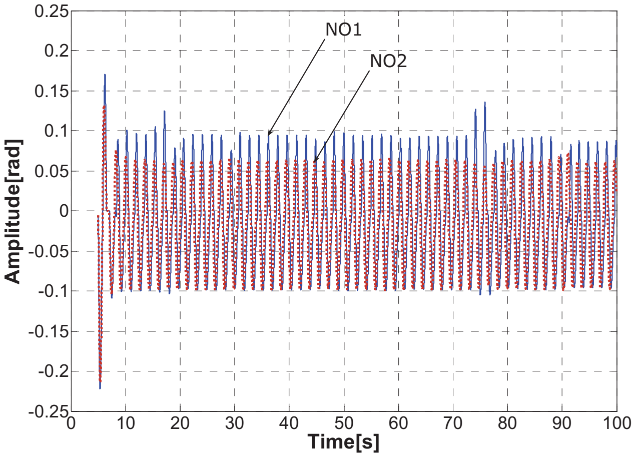


Fig. 10. The output of the neural oscillator coupled to the joints of the arm

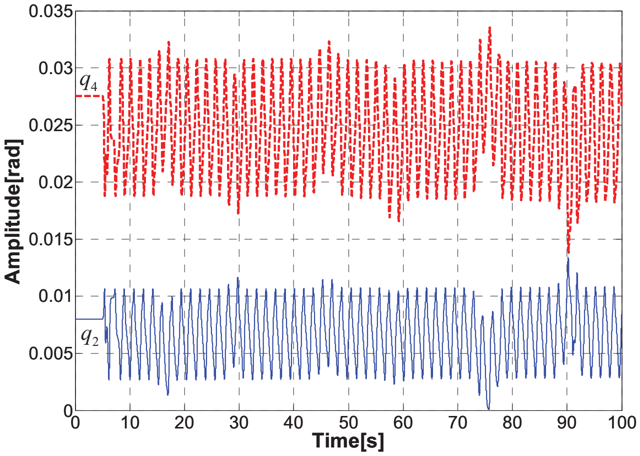


Fig. 11. The output of the first joint (q_2) and the second joint (q_4)

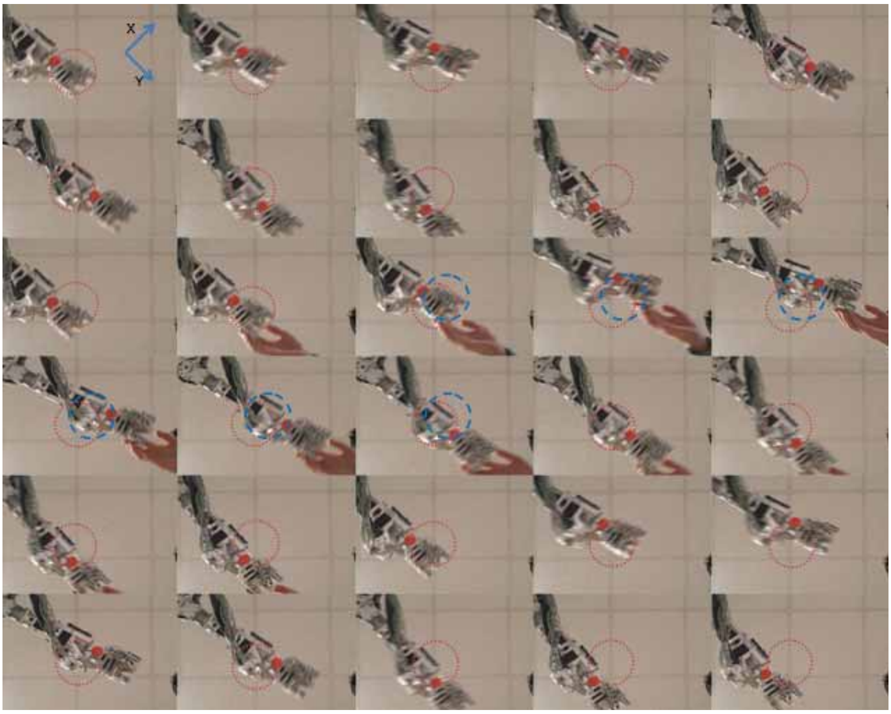


Fig. 12. Snap shots of the arm motion

6. Conclusion

This chapter presents an example of human-like behavior of a planar robot arm whose joints were coupled to neural oscillators. In contrast to existing works that were only capable of rhythmic pattern generation, the proposed approach allowed the robot arm to trace a trajectory correctly through entrainment. For successfully achieving this, we proposed an optimization approach for obtaining the parameters of the neural oscillator modifying the simulated annealing method. Simulation and experimental results showed the effectiveness of the proposed approach. Moreover, it was demonstrated that the robot arm could adaptively behave responding to external disturbances keeping the shape of the trajectory unchanged. This approach will be extended to a more complex behavior toward the realization of biologically inspired robot control architectures.

7. Acknowledgement

This work was supported in part by Korea MIC & IITA through IT Leading R&D Support Project. This research was also conducted as a program for the “Fostering Talent in Emergent Research Fields” in Special Coordination Funds for Promoting Science and Technology by Japan Ministry of Education, Culture, Sports, Science and Technology.

8. References

Matsuoka, K. (1985). Sustained Oscillations Generated by Mutually Inhibiting Neurons with Adaptation, *Biological Cybernetics*, Vol. 52, No. 6, pp. 367-376, ISSN 0340-1200, October 1985

Matsuoka, K. (1987). Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators, *Biological Cybernetics*, Vol. 56, No. 5-6, pp. 345-353, ISSN 0340-1200, July 1987

Taga, G.; Yamagushi, Y. & Shimizu, H. (1991). Self-organized Control of Bipedal Locomotion by Neural Oscillators in Unpredictable Environment, *Biological Cybernetics*, Vol. 65, No. 3, pp. 147-159, ISSN 0340-1200, July 1991

Taga, G. (1995). Model of the Neuro-musculo-skeletal System for Human Locomotion, *Biological Cybernetics*, Vol. 73, No. 2, pp. 91-111, ISSN 0340-1200, August 1995

Miyakoshi, S.; Taga, G.; Kuniyoshi, Y. & Nagakubo, A. (1998). Three-dimensional Bipedal Stepping Motion Using Neural Oscillators-Towards Humanoid Motion in the Real World, *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 84-89, Canada, October 1998, Victoria B.C.

Fukuoka, Y.; Kimura, H. & Cohen, A. H. (2003). Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts, *Int. Journal of Robotics Research*, Vol. 22, No. 3-4, pp. 84-89, ISSN, 0278-3649, March-April 2003

Endo, G.; Nakanishi, J.; Morimoto, J. & Cheng, G. (2005). Experimental Studies of a Neural Oscillator for Biped Locomotion with QRIO, *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pp. 596-602, Spain, April 2005, Barcelona

Yang, W.; Chong, N. Y.; Ra, S.; Kim, C. & You, B. J. (2008). Self-stabilizing Bipedal Locomotion Employing Neural Oscillators, *Proceedings of IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 8-5, Korea, December 2008, Daejeon

Williamson, M. M. (1996). Postural Primitives: Interactive Behavior for a Humanoid Robot Arm, *Proceedings of 4th Int. Conf. on Simulation of Adaptive Behavior*, pp. 124-131, ISBN-10:0-262-63178-4, USA, September 1996, MIT Press, Cape Cod

Williamson, M. M. (1998). Rhythmic Robot Arm Control Using Oscillators, *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 77-83, Canada, October 1998, Victoria

A. M. Arsenio, A. M. (2000). Tuning of neural oscillators for the design of rhythmic motions, *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pp. 1888-1893, USA, April 2000, California

Yang, W.; Chong, N. Y.; Kim, C. & You, B. J. (2007). Optimizing Neural Oscillator for Rhythmic Movement Control, *Proceedings of IEEE Int. Symp. on Robot and Human Interactive Communication*, pp. 807-814, Korea, September 2007, Jeju

Yang, W.; Chong, N. Y.; Kim, C. & You, B. J. (2008). Entrainment-enhanced Neural Oscillator for Rhythmic Motion Control, *Journal of Intelligent Service Robotics*, Vol. 1, No. 4, pp. 303-311, ISSN 1861-2776, October 2008

Slotine, J.-J. E. & Li, W. (1991). *Applied Nonlinear Control*, Prentice Hall., ISBN 0-13-040890, New Jersey

Kirkpatrick, S.; Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, ISSN 0036-8075, May 1983

Geman, S. & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721-741, ISSN 0018-9340, November 1984

Gomi, H., & Osu, R. (1998). Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environment," *Journal of Neuroscience*, pp. 8965-8978, ISSN 0270-6474, November 1998, Washington DC