2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Learning Robot Stiffness for Contact Tasks Using the Natural Actor-Critic

Byungchan Kim, Byungduk Kang, Shinsuk Park, and Sungchul Kang

***Abstract*— This paper introduces a novel motor learning strategy for robotic contact task based on a human motor control theory and machine learning schemes. Humans modulate their arm joint impedance parameters during contact tasks, and such aspect suggests a key feature how human successfully executes various contact tasks in variable environments. Our strategy for successful contact tasks is to find appropriate impedance parameters for optimal task execution by Reinforcement Learning (RL). In this study Recursive Least-Square (RLS) filter based episodic Natural Actor-Critic is employed to determine the optimal impedance parameters. Through dynamic simulations of contact tasks, this paper demonstrates the effectiveness of the proposed strategy. The simulation results show that the proposed method successfully optimizes the performance of the contact task and adapts to uncertain conditions of the environment.**

I. Introduction

FOR the past decades, robot technologies have advanced remarkably, especially in the field of factory automation. In such fields, the capability of robot was evaluated by standards, such as accurate movement, repeatability, and durability. It is well known that robots perform tasks well in static environments. On the contrary, increasing demands for robot technologies request for robots new assignments which contain various task abilities including contact tasks in human-living environments. However, robots exhibit poor performances in such contact tasks in dynamic environment. Contrary to robots, humans cope with the problems with dynamic environments by the aid of excellent adaptation and learning ability. In this sense, robot control strategy inspired by human motor control would be a promising approach.

As one of the human motor control method, the equilibrium point control hypothesis offers fundamental framework for motor control [1]. Fig. 1 illustrates the concept of the equilibrium point control hypothesis. The equilibrium point control hypothesis states that the muscles and neural control circuits have "spring-like" properties, and the central nervous system (CNS) may generate a series of equilibrium points for a limb, and the "spring-like" properties of the neuromuscular system will tend to drive the motion along a trajectory that follows these intermediate equilibrium postures [2]. The "spring-like" properties could be interpreted as a muscle's "stiffness" or more generally "impedance," which plays a principal role in handling dynamic characteristics [3]. These features enable effective control of interaction between the manipulator and the environment by simply modulating these impedance parameters.

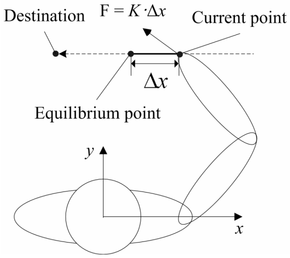

Fig. 1 A conceptual model of equilibrium point control hypothesis.

There have been several studies on biologically-inspired motor learning. Cohen *et al.* suggested impedance learning strategy in a wall-following contact task by using Associative Search Network [4]. Another study on motor learning was performed by Izawa *et al.* [5]. They have investigated a motor learning method developed by RL for the musculoskeletal arm model in free movement. Though these studies are fairly interesting approach, they have the limitation of considering simple contact/non-contact problems only.

On the other hand, there have been several works on impedance learning using Artificial Neural Network (ANN) in contact task [6], [7]. One of the noticeable works was done by Tsuji *et al.* [7]. This work suggests on-line virtual impedance learning method by exploiting visual information. Despite of its usefulness, ANN-based learning involves heavy computational load and easily lead to local minima problem.

In this study, we chose RL as our learning framework. RL can handle an optimization problem in an unknown environment by making sequential decision policies that maximize external reward [8]. Though RL is used in wide areas of machine learning, there are several disadvantages, e.g., much computation time and large variance of samples, since it is the Monte-Carlo-based estimation method. Currently, two policy gradient approaches are being investigated to resolve these problems. One approach provides baseline for gradient estimator for reducing variance [9], [10], and the other suggests Bayesian update rule for

This research was performed for the Intelligent Robotics Development Program which is one of the 21st Century Frontier R&D Program funded by the Korean Ministry of Commerce, Industry and Energy.

Byungchan Kim and Byungduk Kang are with the Department of Mechanical Engineering, Korea University, Seoul (phone: +82-2-3290-3868; e-mail: biomimetic@korea.ac.kr, haga99@korea.ac.kr).

Shinsuk Park is with the Department of Mechanical Engineering, Korea University, Seoul (phone: +82-2-3290-3373; fax: +82-2-926-9230; e-mail: drsspark@korea.ac.kr).

Sungchul Kang is with the Center for Cognitive Robotics Research, KIST, Seoul (phone: +82-2-958-5589; fax: +82-2-958-5629; e-mail: kasch804@gmail.com).

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

estimating gradient [11]. In this work, we select the former approach for constructing our RL algorithm.

In this paper we propose a robot motor learning strategy based on the equilibrium point control hypothesis, in managing a contact task problem. In this work, episodic Natural Actor-Critic algorithm based on RLS filter is implemented for learning algorithm. Episodic Natural Actor-Critic method proposed by Peters *et al.* is known effective in high-dimensional continuous state/action system problems and can provide optimum closest solution [12]. To further reduce computational burden, we combined the method with RLS filter that was firstly suggested in the study of Park *et al.* [13]. This paper shows the effectiveness of the proposed strategy from the simulation results of multi-dimensional contact task, which is trajectory following in unknown environment and catching a flying ball using a two-link manipulator.

This paper is organized as follows. The following section introduces the impedance-based controller. Next, we describe the details of learning strategy. Finally, simulation results and discussion of the results are presented.

II. Design Motor Controller

A. Impedance Control and Stiffness ellipse

Impedance as a physical characteristic of robot arm is one important factor for the dynamic interaction between a robot and its environment in task-space. Like a human executes the given task by changing human muscle's property, a robot modulates impedance parameters to adapt to an environment.

The control law for contact task is given as follows: [14]

$$\mathbf{T} = -J^T(\mathbf{q})\left[\mathbf{K}_C\tilde{\mathbf{x}} + \mathbf{B}_C\dot{\tilde{\mathbf{x}}}\right], \tag{1}$$

$$\tilde{\mathbf{x}} := \mathbf{x} - \mathbf{x}_d, \tag{2}$$

In equation (1), J ($\mathbf{q}$) is the manipulator Jacobian. In equation (2), the difference between the current position and the desired position of end-effector is defined as the displacement vector $\tilde{\mathbf{x}}$. Matrices $\mathbf{K}_C$ and $\mathbf{B}_C$ in (1) are cartesian-stiffness matrix and cartesian-damping matrix, respectively. In this work, we assumes that the damping matrix is roughly proportional to the stiffness in joint-space, and the ratio of joint-damping matrix and the joint-stiffness matrix implies a time constant $\tau = \mathbf{B}_J/\mathbf{K}_J$. The time constant is chosen to be 0.05 sec as in works of Flash and Won for arm movement [15].

For impedance control of a two-link manipulator, 2×2 cartesian-stiffness matrix is formed as follows:

$$K_C = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}. \tag{3}$$

Using Singular Value Decomposition, the stiffness matrix can be decomposed as follows:

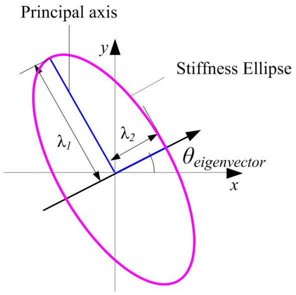

Fig. 2 A graphical representation of the end-effector's stiffness in Cartesian space. The principal axes length λ_i and relative angle θ represents the magnitude and orientation of the end-effectors stiffness, respectively.

$$K_C = V\Sigma V^T, \tag{4}$$

where, orthogonal matrix V consists of the eigenvectors of stiffness matrix, and diagonal matrix Σ consists of the eigenvalues of the stiffness matrix. These matrices could be represented geometrically as in Fig. 2. Now that the features of ellipse reflect the properties of end-effector's stiffness, elements of stiffness matrix should be determined considering the ellipse's characteristics – the magnitude (area of ellipse), shape (the ratio of major and minor axes), and orientation (direction of the major axis). Thus, the magnitude, shape, and orientation of the stiffness ellipse can be chosen to be control variables of the stiffness. By regulating the three parameters, desired impedance control can be achieved. The task-space stiffness matrix is assumed to be symmetrical and positively definite, and this provides a sufficient condition for static stability of the manipulator when it interacts with a passive environment [16].

Another issue for impedance control is determination of virtual trajectory. This work sets the virtual end-effector trajectory which has a minimum- jerk velocity profile for the manipulator's smooth movement [17]. The end-effector trajectory in task-space is planned from the start point x_i to the final point x_f as follows:

$$x(t) = x_i + (x_f - x_i)(10(\frac{t}{t_f})^3 - 15(\frac{t}{t_f})^4 + 6(\frac{t}{t_f})^5) \tag{5}$$

Here, t is a current time and t_f is the duration of movement.

III. Motor Skill Learning Strategy

A. Reinforcement Learning

In RL, the agent selects action a_t in discrete time t, then, receives next environmental state s_t and scalar valued reward r_t as a result of the action. The reward r_t is a criterion which indicates the action performance. Thus, the agent strives to maximize reward r_t by modulating policy $\pi(s_t,a_t)$ which determines action at each state.

The RL aims to maximize the total sum of afterward rewards or the expected return. A discounted sum of rewards during one episode is widely used for the expected return as follows:

$$R_t = \sum_{k=0}^{T} \gamma^k \cdot r_{t+k+1}, \; V^\pi(s) = E_\pi\left\{ \sum_{k=0}^{T} \gamma^k \cdot r_{t+k+1} \middle| s_t = s \right\}. \tag{6}$$

In equation (6), γ is a discounting factor ($0 \le \gamma \le 1$), and $V^\pi(s)$ is the value function which is an expected sum of rewards.

In RL, the update rule of value function givens as,

$$V(s_t) \leftarrow V(s_t) + \alpha\big(r_t - \gamma \cdot V(s_{t+1}) - V(s_t)\big), \tag{7}$$

here, $r_t - \gamma \cdot V(s_{t+1}) - V(s_t)$ is called a Temporal Difference (TD) error which plays an important role indicating whether the action at state s_t is good or not. Therefore, this updated rule is repeated for maximizing value function $V^\pi(s)$ until converge.

B. *RLS-based episodic Natural Actor-Critic*

In determining the learning algorithm, the task to be learned should be modeled. There are two conditions of task model. First, the task is modeled as a series of discrete moments (state). In this model, one procedure (one episode) finishes when state transition ends, and then it goes back to the initial state. Second, the task is modeled as a continuous state/action problem. Since such a high-dimensional continuous state/action system problems are more complicated to solve than discrete state/action ones, adopting Natural Actor-Critic (NAC) algorithm is an effective approach [12]. For reducing computational burden, Park *et al.* suggested modified NAC algorithm combined with RLS filter which is based on the Peters' work [13]. However, that algorithm developed for infinitely repeated task which has no final state (non-episodic task). Hence, that is not compatible with our approach that treats episodic task. Therefore, this paper suggests a new algorithm based on the episodic NAC algorithm. We call this algorithm the "RLS-based eNAC algorithm."

The RLS-based eNAC algorithm has two separated memory structure which are called as actor and critic structure, respectively. Actor structure determines policy which selects action at each state, and critic structure criticizes a selected action of actor structure whether the action is good or not.

In the actor structure, the policy is defined as $\pi(a|s) = p(a|s,\theta)$, and policy parameter θ is iteratively updated by following update rule,

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta), \tag{8}$$

where, $\nabla_\theta J(\theta)$ denoted a gradient of objective function. In the work of [9], the gradient of the objective function is derived from the natural gradient method studied by Amari [18]. It suggested a simpler update rule as follows:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \approx \theta + \alpha w. \tag{9}$$

where, α is a learning rate ($0 \le \alpha \le 1$).

In the critic structure, least-squares TD-Q(λ) algorithm is adopted for minimizing TD error which is a deviation between expected return and current prediction value. From the value function update rule in (7), value function can be approximated by summing of two critic information vector, one is the compatible function approximator $A^\pi(s_t, a_t) = \nabla_\theta \log \pi(a_t|s_t)^T w_t$ and the other is the value function $V^\pi(\mathbf{s})$[1]. Rolling out all steps information of one episode, equation is summarized as:

$$\sum\nolimits_{t=0}^{N} \gamma^t [\nabla_\theta \log \pi(a_t|s_t)^T, 1][\mathbf{w}^T, \mathbf{v}^T]^T = \sum_{i=0}^{N} \gamma^i r(\mathbf{s}_i, \mathbf{a}_i). \tag{10}$$

In equation (10), two vector w and v are used for updating actor parameter in (9), and for approximating a value function, respectively. Now, we can handle (10) as a least-square problem. Denoting each component of (10) to $\tilde{\phi} = \sum\nolimits_{t=0}^{N} \gamma^t [\nabla_\theta \log \pi(a_t|s_t)^T, 1]^T$, $\chi = [\mathbf{w}^T, \mathbf{v}^T]^T$ and $\tilde{r} = \sum_{i=0}^{N} \gamma^i r(\mathbf{s}_i, \mathbf{a}_i)$, a least-square form of (10) could be denoted by,

$$\chi = G^{-1} u, \tag{11}$$

where $u = \tilde{\phi}\tilde{r}$, $G = \tilde{\phi}\tilde{\phi}^T$. Using this method, solution vector χ can be solved immediately from the accumulated information obtained through all the episodes. After the first episode is ended, the critic information matrices G, u, χ are updated like this:

$$G_0 \triangleq \delta \mathbf{I} + \tilde{\phi}_0 \tilde{\phi}_0^T, \; P_0 \triangleq G_0^{-1}. \tag{12}$$

In equation (12), additional identity matrix $\delta\mathbf{I}$, where δ is a positive scalar constant, is added to ensure the invertibility. However, from the second episode of critic update, the inverse matrix of P is obtained by RLS update rule which efficiently computes matrix inverse. Then, we could get the solution vector χ [19]. RLS update rule is given by,

$$\begin{aligned} P_{e+1} &= \frac{1}{\beta}\left(P_e - \frac{P_e \tilde{\phi}_e \tilde{\phi}_e^T P_e}{\beta + \tilde{\phi}_e^T P_e \tilde{\phi}_e} \right), \\ k_{e+1} &= \frac{P_e \tilde{\phi}_e}{\beta + \tilde{\phi}_e^T P_e \tilde{\phi}_e}, \\ \chi_{e+1} &= \chi_e + k_{e+1}(\tilde{r}_e - \phi_e^T \chi_e). \end{aligned} \tag{13}$$

[1] More details about these derivation procedures were presented in [12].

TABLE I
RLS-BASED EPISODIC NATURAL ACTOR CRITIC ALGORITHM

```
Initialize each parameter vector:
  $\theta = \theta_0, A = \mathbf{0}, b = \mathbf{0}, s_t = \mathbf{0}$
for each episode,
  Run simulator:
    for each step,
      Take action $a_t$, from stochastic policy $\pi$,
      then, observe next state $s_{t+1}$, reward $r_t$.
    end
  Update Critic structure:
    if first update,
      Update critic information matrices,
      following the initial update rule in (12).
    else
      Update critic information matrices,
      following the recursive least-squares update rule in (13).
    end
  Update Actor structure:
    Update policy parameter vector following the rule in (9).
repeat until converge
```

In equation (13), β is a forgetting factor ($0 \leq \beta \leq 1$) for accumulating past information in a discount manner. TABLE I shows the entire process of RLS-based episodic Natural Actor-Critic algorithm.

C. Stochastic Action Selection

The policy π plans a change rate of the magnitude, shape and orientation of stiffness ellipse at each state. Through the sequence of one episode, the parameters of stiffness ellipse are changed by policy. The final goal of the learning algorithm is to find optimized stiffness ellipse trajectory in episode.

Policy π is in the form of Gaussian density function. In critic structure, compatible function approximator $\nabla_\theta \log \pi(a_t|s_t)^T$ is derived from the stochastic policy π, and derivation procedure follows Williams's study [9]. Policy is given by,

$$\pi(a|s) = N(a|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(a-\mu)^2}{2\sigma^2}\right)\cdot \tag{14}$$

Since stochastic action selection is dependent on the condition of Gaussian density function, determining action policy is accomplished by tailoring the means μ and variances σ in (14). These variables are defined by,

$$\mu_{act,t} = \xi_{act}\boldsymbol{\omega}^T_{act}\mathbf{s}_t\,,\quad \sigma_{act} = \tilde{\xi}_{act}\left(0.001 + \frac{1}{1+\exp(-\eta_{act})}\right), \tag{15}$$

where mean μ has a linear form of vector product with actor parameter vector $\boldsymbol{\omega}$ and state vector $\boldsymbol{s}_t$ of each state. Constant coefficient ξ is a mean scaling factor, and variable σ with sigmoid function form has another actor parameter η and variance scaling factor $\tilde{\xi}$. $\boldsymbol{s}_t$ is 5×1 vector is composed of joint angle displacement and velocity, $\mathbf{s}_t = [x_1\ x_2\ \dot{x}_1\ \dot{x}_2\ 1]^T$. There are three actions as shown in II. Therefore, actor parameter vector $\mathbf{w}=[\boldsymbol{\omega}^T, \boldsymbol{\eta}^T]^T$ is composed of 15 mean elements ω and 3 variance elements η, totally 18×1 vector form.

IV. CONTACT TASK APPLICATIONS

This section shows two contact task problems and the results of task simulations.

A. Trajectory Following in Unknown Environment

First task is a trajectory following using a simple two-link revolute joint manipulator in an unknown force field environment. This task is motivated by the biomechanical study on arm movement [20]. In this work, the subject moves her hand to the goal position in a given time. However, unlike the case of free arm movement, the environment the subject interacts with is set to be a velocity-dependent force field. Therefore, the hand is hampered by receiving a force resistance as it move to the goal direction. The force field defined as follows,

$$F_{damping} = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}\begin{Bmatrix} \dot{x} \\ \dot{y} \end{Bmatrix}. \tag{16}$$

The general dynamic equation for the simulation model is,

$$\mathbf{M(q)\ddot{q}} + \mathbf{C(q,\dot{q})} + \mathbf{G(q)} = \tau + \mathbf{J}^T\mathbf{F}_{damping}\,, \tag{17}$$

where, $\mathbf{M(q)}$ are moment of inertia terms, $\mathbf{C(q,\dot{q})}$ are coriolis and centrifugal terms, and $\mathbf{G(q)}$ are gravitational terms. In this work, the gravitational terms $\mathbf{G(q)}$ are neglected since the arm motion plane is perpendicular to the gravity direction. TABLE II presents the physical properties of simulation model.

TABLE II
PHYSICAL PROPERTIES OF SIMULATION MODEL

	Length(m)	Mass(Kg)	Inertia(Kg·m^2)
Link 1	0.11	0.20	0.0002297
Link 2	0.20	0.18	0.0006887

The learning parameters are determined at α = 0.05, β = 0.99, γ = 0.99. The limit bound of each action for changing stiffness ellipse configuration is set to [-10°, 10°](direction angle) for orientation, [-2, 2](ratio) for shape and [-200π, 200π](area) for magnitude. Initial stiffness ellipse follows the configuration in [18]. From the initial state, stiffness ellipse is changed using the action policy at each state.

The performance indices could be denoted by,

$$reward = \rho_1(\kappa_1 - PI_1) + \rho_2(\kappa_2 - PI_2)\,,$$
$$PI_1 = \sum_{t=1}^{N}\sqrt{(x_{d,t}-x_t)^2 + (y_{d,t}-y_t)^2}\,,\ PI_2 = \sum_{t=1}^{N}\sqrt{\dot{\tau}_{1,t}{}^2 + \dot{\tau}_{2,t}{}^2}\,. \tag{18}$$

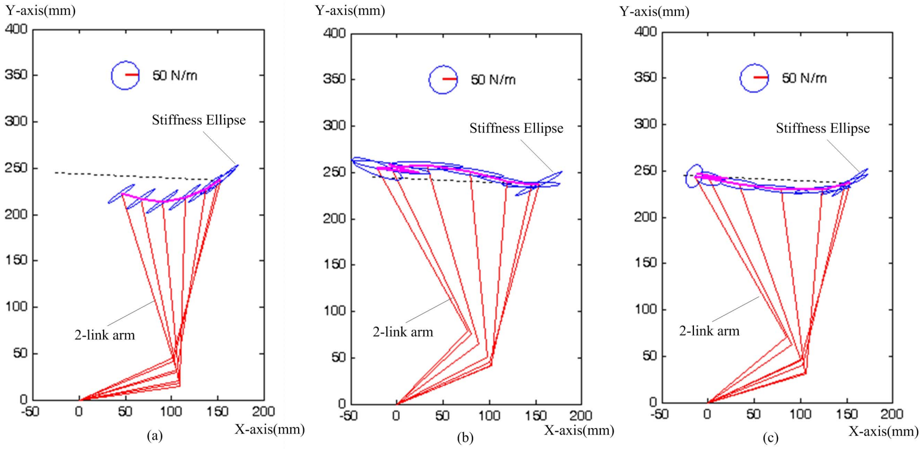

Fig. 3 A stiffness ellipse trajectory. From these figures, dotted line is a virtual trajectory and solid line is an actual trajectory. (a) Before learning. (b) After learning (PI: Position error). (c) After learning (PI: position error and torque rate)

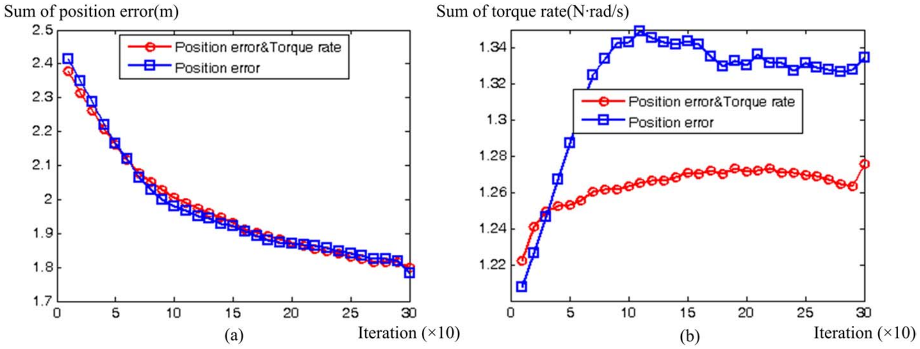

Fig. 4 The learning result of performance indices (average of 10 learning trial). (a) Sum of position error. (b) Sum of torque rate.

In equation (18), PI_1 is the sum of the position error between desired position and actual position of end-position of arm, and PI_2 is the sum of the torque rate of each arm joint. The latter one represents the minimum-energy comsumption, which can be interpreted by metabolic costs for arm [21]. *reward* is set to be a weighted linear combination of these two PIs using parameter ρ. Parameter κ is an first-trial value of each PIs, therefore, the more PI is decrease, the more reward is increase.

Fig. 3 shows the transition of stiffness ellipse trajectory before and after learning. As shown is Fig. 3, different combination of PIs yield different action phases. By and large, the major axis of the ellipse was directed to the goal position to overcome resistance of force field. However, examining the learning results in Fig. 4, the case of performance index combined with position error and torque rate supresses the increase of sum of torque rate than the case of only position error, with maintaining the similar performace enhancement in reducing position error.

B. Catching a Flying Ball

This task is a ball catching problem. The main issues in ball catching problem are how to detect ball trajectory and how to reduce impulse when a gripper catch the flying ball. This work focuses on the latter and assumes that the robot knows the ball trajectory in advance.

Fig. 5 shows a simulation model of ball catching task. The physical properties of the arm are identical to those of the task model in the previous section. In this experiment, the dynamic simulation was performed using MSC.ADAMS2005, and control algorithm is implemented using Matlab/Simulink (Mathworks, Inc.).

A task process is as follows: A ball is thrown to the arm gripper. The time for the ball to reach the gripper is approximately 0.8sec. After the ball is thrown to the gripper, the arm starts to move following the parabolic orbit of flying ball. While the gripper is moving, the gripper catches the ball, and then the arm moves to the goal point. The robot is set to catch the ball when the end-effector's moving speed is the highest. This is assumed by considering human catching action: when a human catches a ball, she moves her arm backward to reduce impulse between the ball and her hand.

The learning parameters and the limit bound of stiffness ellipse are identical to those of the task of the previous section except that we set the initial condition of stiffness ellipse to have a circular form with the area of 10000π. The reward is set to be an accumulated impulse at the moment of impact.

$$reward = \sum_{i=1}^{N} \sqrt{F_{x,i}^{\ 2} + F_{y,i}^{\ 2}}\,\Delta t_i \,. \qquad (19)$$

Fig. 6 illustrates the stiffness ellipse trajectory after

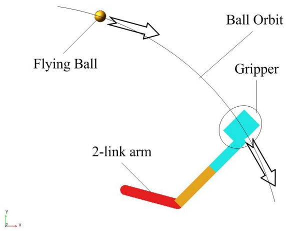

Fig. 5 A simulation model of catching a flying ball.

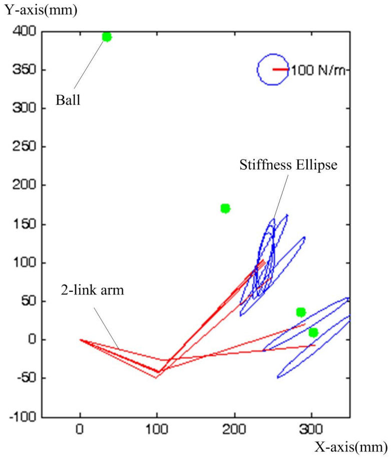

Fig. 6 A stiffness ellipse trajectory.

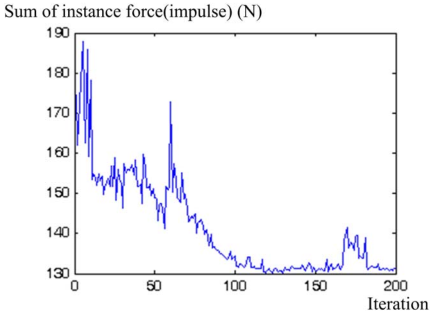

Fig. 7 A learning result of accumulated impulse.

learning. As can be seen in the figure, the stiffness is soft in the direction of ball trajectory, while the stiffness normal to the ball orbit is much higher. Fig. 7 shows the gradual reduction of the accumulated impulse after iterations of learning.

V. Conclusions

Safety in contact tasks in unknown environments has become one of the important issues as robots spread their application to human-living space. The determination of impedance control parameters for a specific contact task would be the key feature in enhancing the robot performance. This work proposes a motor learning strategy for acquiring manipulation skills for contact tasks. We have demonstrated that the proposed learning method enhances the performance of some contact tasks.

Future works we are preparing is to extend task field which treats more complex tasks such as complex assembly and 3 dimensional problems from the 2 dimensional tasks as shown in this paper. We expect that this strategy can be used to teach motor skills to a service robot that needs to carry out various contact tasks.

References

[1] T. Flash, “The Control of Hand Equilibrium Trajectories in Multi-Joint Arm Movement,” *Biol. Cybern.* vol. 57, pp. 257-274, 1987.

[2] S. Park and T. B. Sheridan, “Enhanced Human-Machine Interface in Braking,” *IEEE Trans. Sys. Man, and Cyber., - Part A: Sys., and Humans*, vol. 34, no. 5, pp. 615-629, 2004.

[3] N. Hogan, “Impedance control: An approach to manipulation: Part I. Theory, Part II. Implementaion, Part III. Application,” *ASME J. Dynamic Syst. Measurement Control*, vol. 107, pp. 12-24, 1985.

[4] M. Cohen and T. Flash, “Learning Impedance Parameters for Robot control using Associative Search Network,” *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 382-390, 1991.

[5] J. Izawa, T. Kondo, and K. Ito, “Biological Robot Arm Motion through Reinforcement Learning,” *In Proc. of IEEE Int . Conf. on Robotics and Automat.*, vol. 4, pp. 3398-3403, 2002.

[6] S. Jung, S. B. Yim and T. C. Hsia, “Experimental Studies of Neural Network Impedance Force Control of Robot Manipulator,” *In Proc. of IEEE Int . Conf. on Robotics and Automat.*, pp. 3453-3458, 2001.

[7] T. Tsuji, M. Terauchi, and Y. Tanaka, “Online Learning of Virtual Impedance Parameters in Non-Contact Impedance Control Using Neural Networks,” *IEEE Trans. Sys. Man, and Cyber., - Part B: Cyber.*, vol. 34, no. 5, pp. 2112-2118, 2004.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*, MIT Press, 1998.

[9] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning*, vol. 8, pp. 229-256. 1992.

[10] J. Peters and S. Schaal, “Policy Gradient Methods for Robotics,” *In Proc. of IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp. , 2006.

[11] Y. Engel, S. Mannor, and Y. Mansour, “Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning,” *In Proc. of the 20th Int. Conf. on Machine Learning*, pp. 154-161, 2003.

[12] J. Peters, S. Vijayakumar, and Schaal, "Natural Actor-Critic," *In Proc. of the 16th Euro. Conf. on Machine Learning,* vol. 3720, pp.280-291, 2005.

[13] J. Park, J. Kim, and D. Kang, “An RLS-Based Natural Actor-Critic Algorithm for Locomotion of a Two-Linked Robot Arm,” *In Proc. of Int. Conf. CIS*, Part I, LNAI, vol. 3801, pp. 65-72, 2005.

[14] H. Asada and J-J. E. Slotine, *Robot Analysis and Control*, John Wiley & Sons, Inc., 1986.

[15] J. Won, “The Control of Constrained and Partially Constrained Arm Movement,” S. M. Thesis, Dep. Of Mechanical Engineering, MIT, Cambridge, MA, 1993.

[16] H. Kazeroni, P. K. Houpt, and T. B. Sheridan, “The Fundamental Concepts of Robust Compliant Motion for Robot Manipulators,” *In Proc. of IEEE Int. Conf. on Intel. Robotics and Automat.*, vol. 1, pp. 418-427, 1986.

[17] T. Flash and N. Hogan, “The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model,” *J.of Neurosci.*, vol. 5, n. 7, pp. 1688-1703, 1985.

[18] S. Amari, “Natural Gradient Works Efficiently in Learning,” *Neural Computation*, vol. 10, 1998.

[19] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithm for Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 2000.

[20] B. Kang, B. Kim, S. Park, and H. Kim, “Modeling of Artificial Neural Network for the Prediction of the Multi-Joint Stiffness in Dynamic Condition,” *In Proc. of IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pp. 1840-1845, 2007.

[21] D. W. Franklin, U. So, M. Kawato, and T. E. Milner, “Impedance Control Balances Stability With Metabolically Costly Muscle Activation,” *J. of Neurophysiol.*, vol 92, pp. 3097-3105, 2004.