14

Novel Framework of Robot Force Control Using Reinforcement Learning

Byungchan Kim[1] and Shinsuk Park[2]
[1]*Center for Cognitive Robotics Research, Korea Institute of Science and Technology*
[2]*Department of Mechanical Engineering, Korea University*
Korea

1. Introduction

Over the past decades, robotic technologies have advanced remarkably and have been proven to be successful, especially in the field of manufacturing. In manufacturing, conventional position-controlled robots perform simple repeated tasks in static environments. In recent years, there are increasing needs for robot systems in many areas that involve physical contacts with human-populated environments. Conventional robotic systems, however, have been ineffective in contact tasks. Contrary to robots, humans cope with the problems with dynamic environments by the aid of excellent adaptation and learning ability. In this sense, robot force control strategy inspired by human motor control would be a promising approach.

There have been several studies on biologically-inspired motor learning. Cohen et al. suggested impedance learning strategy in a contact task by using associative search network (Cohen et al., 1991). They applied this approach to wall-following task. Another study on motor learning investigated a motor learning method for a musculoskeletal arm model in free space motion using reinforcement learning (Izawa et al., 2002). These studies, however, are limited to rather simple problems. In other studies, artificial neural network models were used for impedance learning in contact tasks (Jung et al., 2001; Tsuji et al., 2004). One of the noticeable works by Tsuji et al. suggested on-line virtual impedance learning method by exploiting visual information. Despite of its usefulness, however, neural network-based learning involves heavy computational load and may lead to local optimum solutions easily.

The purpose of this study is to present a novel framework of force control for robotic contact tasks. To develop appropriate motor skills for various contact tasks, this study employs the following methodologies. First, our robot control strategy employs impedance control based on a human motor control theory - the equilibrium point control model. The equilibrium point control model suggests that the central nervous system utilizes the *spring*-like property of the neuromuscular system in coordinating multi-DOF human limb movements (Flash, 1987). Under the equilibrium point control scheme, force can be controlled separately by a series of equilibrium points and modulated stiffness (or more generally impedance) at the joints, so the control scheme can become simplified considerably. Second, as the learning framework, reinforcement learning (RL) is employed to optimize the performance of contact task. RL can handle an optimization problem in an

unknown environment by making sequential decision policies that maximize external reward (Sutton et al., 1998). While RL is widely used in machine learning, it is not computationally efficient since it is basically a Monte-Carlo-based estimation method with heavy calculation burden and large variance of samples. For enhancing the learning performance, two approaches are usually employed to determine policy gradient. One approach provides the baseline for gradient estimator for reducing variance (Peters et al., 2006), and the other suggests Bayesian update rule for estimating gradient (Engel et al., 2003). This study employs the former approach for constructing the RL algorithm.

In this work, episodic natural actor-critic algorithm based on the RLS filter was implemented for RL 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 (Peters et al., 2005). A RLS filter is used with the Natural Actor-Critic algorithm to further reduce computational burden as in the work of Park *et al.* (Park et al., 2005). Finally, different task goals or performance indices are selected depending on the characteristics of each task. In this work, the performance indices for two contact tasks were chosen to be optimized: point-to-point movement in an unknown force field, and catching a flying ball. The performance of the tasks was tested through dynamic simulations.

This paper is organized as follows. Section 2 introduces the equilibrium point control model based impedance control methods. In section 3, we describe the details of motor skill learning based on reinforcement learning. Finally, simulation results and discussion of the results are presented.

2. Impedance control based on equilibrium point control model

Mechanical impedance of a robot arm plays an important role in the dynamic interaction between the robot arm and its environment in contact. Impedance control is a widely-adopted control method to execute robotic contact tasks by regulating its mechanical impedance which characterizes the dynamic behavior of the robot at the port of interaction with its environment. The impedance control law may be described as follows (Asada et al., 1986):

$$\boldsymbol{\tau}_{actuator} = -\mathbf{J}^T(\mathbf{q})\left[\mathbf{K}_C(\mathbf{x}-\mathbf{x}_d)+\mathbf{B}_C\dot{\mathbf{x}}\right] \tag{1}$$

Where $\boldsymbol{\tau}_{actuator}$ represents the joint torque exerted by the actuators, and the current and desired positions of the end-effector are denoted by vectors $\mathbf{x}$ and $\mathbf{x}_d$, respectively. Matrices $\mathbf{K}_C$ and $\mathbf{B}_C$ are stiffness and damping matrices in Cartesian space. This form of impedance control is analogous to the equilibrium point control, which suggests that the resulting torque by the muscles is given by the deviations of the instantaneous hand position from its corresponding equilibrium position. The equilibrium point control model proposes that the muscles and neural control circuits have "spring-like" properties, and the central nervous system 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 (Park et al., 2004; Hogan, 1985). Fig. 1 illustrates the concept of the equilibrium point control. Impedance control is an extension of the equilibrium point control in the context of robotics, where robotic control is achieved by imposing the end-effector dynamic behavior described by mechanical impedance.

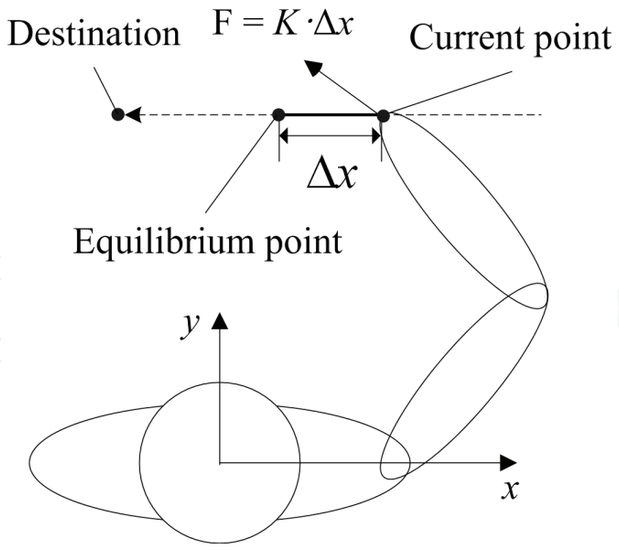

Figure 1. Conceptual model of equilibrium point control hypothesis

For impedance control of a two-link manipulator, stiffness matrix $\mathbf{K}_C$ is formed as follows:

$$\mathbf{K}_C = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \tag{2}$$

Stiffness matrix $\mathbf{K}_C$ can be decomposed using singular value decomposition:

$$\mathbf{K}_C = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T \tag{3}$$

, where $\boldsymbol{\Sigma} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

In equation (3), orthogonal matrix $\mathbf{V}$ is composed of the eigenvectors of stiffness matrix $\mathbf{K}_C$, and the diagonal elements of diagonal matrix $\boldsymbol{\Sigma}$ consists of the eigenvalues of stiffness matrix $\mathbf{K}_C$. Stiffness matrix $\mathbf{K}_C$ can be graphically represented by the *stiffness ellipse* (Lipkin et al., 1992). As shown in Fig. 2, the eigenvectors and eigenvalues of stiffness matrix correspond to the directions and lengths of principal axes of the stiffness ellipse, respectively. The characteristics of stiffness matrix $\mathbf{K}_C$ are determined by three parameters of its corresponding stiffness ellipse: the magnitude (the area of ellipse: $2\lambda_1\lambda_2$), shape (the length ratio of major and minor axes: λ_1/λ_2), and orientation (the directions of principal axes: θ). By regulating the three parameters, all the elements of stiffness matrix $\mathbf{K}_C$ can be determined.

In this study, the stiffness matrix in Cartesian space is assumed to be symmetric and positive definite. This provides a sufficient condition for static stability of the manipulator when it interacts with a passive environment (Kazerooni et al., 1986). It is also assumed that damping matrix $\mathbf{B}_C$ is approximately proportional to stiffness matrix $\mathbf{K}_C$. The ratio $\mathbf{B}_C/\mathbf{K}_C$ is chosen to be a constant of 0.05 as in the work of Won for human arm movement (Won, 1993).

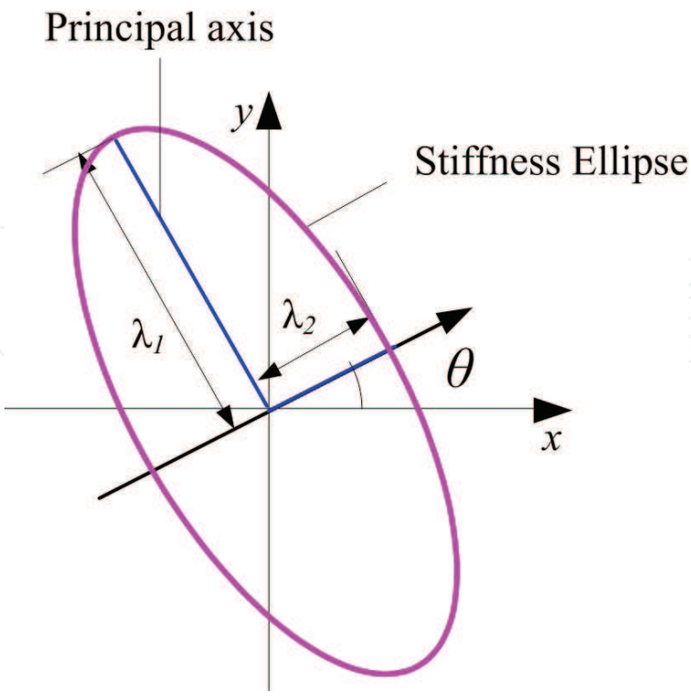

Figure 2. A graphical representation of the end-effector's stiffness in Cartesian space. The lengths λ_1 and λ_2 of principal axes and relative angle θ represent the magnitude and the orientation of the end-effectors stiffness, respectively

For trajectory planning, it is assumed that the trajectory of equilibrium point for the end-effector, which is also called the *virtual trajectory*, has a minimum- jerk velocity profile for smooth movement of the robot arm (Flash et al., 1985). The virtual trajectory is calculated from the start point x_i to the final point x_f as follows:

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

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

3. Motor Skill Learning Strategy

A two-link robotic manipulator for two-dimensional contact tasks was modeled as shown in Fig. 3. The robotic manipulator is controlled using the impedance control method based on the equilibrium point control hypothesis as described in Section 2. The stiffness and damping of the manipulator are modulated during a contact task, while the trajectory of equilibrium point is given for the task. The manipulator learns the impedance modulation strategy for a specific task through reinforcement learning. The state vector is composed of the joint angles and velocities at the shoulder and elbow joints. The action vector changes the three parameters of stiffness ellipse: the magnitude (the area of ellipse), shape (the length ratio of major and minor axes), and orientation (the direction of principal axes). This section describes the learning method based on reinforcement learning for controlling task impedance of the two-link manipulator in performing two-dimensional contact tasks.

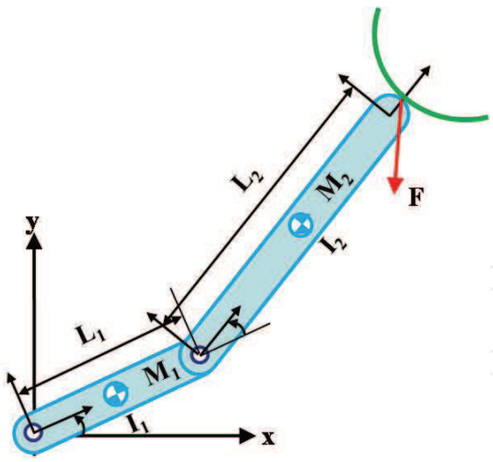

Figure 3. Two-link robotic manipulator (L_1 and L_2: Length of the link, M_1 and M_2: Mass of the link, I_1 and I_2: Inertia of the link)

3.1 Reinforcement Learning

The main components of RL are the decision maker, the agent, and the interaction with the external environment. In the interaction process, the agent selects action $\mathbf{a}_t$ and receives environmental state $\mathbf{s}_t$ and scalar-valued reward r_t as a result of the action at discrete time t. The reward r_t is a function that indicates the action performance. The agent strives to maximize reward r_t by modulating policy $\pi(\mathbf{s}_t,\mathbf{a}_t)$ that chooses the action for a given state $\mathbf{s}_t$. The RL algorithm aims to maximize the total sum of future rewards or the expected return, rather than the present reward. A discounted sum of rewards during one episode (the sequence of steps to achieve the goal from the start state) is widely used as the expected return:

$$R_t = \sum_{i=0}^{T} \gamma^k \cdot r_{t+i+1}$$
$$V^{\pi}(\mathbf{s}) = E_{\pi}\left\{ \sum_{i=0}^{T} \gamma^k \cdot r_{t+i+1} \middle| \mathbf{s}_t = \mathbf{s} \right\} \tag{5}$$

Here, γ is the discounting factor ($0 \le \gamma \le 1$), and $V^{\pi}(\mathbf{s})$ is the value function that represents an expected sum of rewards. The update rule of value function $V^{\pi}(\mathbf{s})$ is given as follows:

$$V^{\pi}(\mathbf{s}_t) \leftarrow V^{\pi}(\mathbf{s}_t) + \alpha\left(r_t + \gamma \cdot V^{\pi}(\mathbf{s}_{t+1}) - V^{\pi}(\mathbf{s}_t)\right) \tag{6}$$

In equation (6), the term $r_t + \gamma \cdot V^{\pi}(\mathbf{s}_{t+1}) - V^{\pi}(\mathbf{s}_t)$ is called the Temporal Difference (TD) error. The TD error indicates whether action $\mathbf{a}_t$ at state $\mathbf{s}_t$ is good or not. This updated rule is repeated to decrease the TD error so that value function $V^{\pi}(\mathbf{s}_t)$ converges to the maximum point.

3.2 RLS-based episodic Natural Actor-Critic algorithm

For many robotic problems, RL schemes are required to deal with continuous state/action space since the learning methods based on discrete space are not applicable to high dimensional systems. High-dimensional continuous state/action system problems, however, are more complicated to solve than discrete state/action space problems. While Natural Actor-Critic (NAC) algorithm is known to be an effective approach for solving continuous state/action system problems, this algorithm requires high computational burden in calculating inverse matrices. To relieve the computational burden, Park et al. suggested modified NAC algorithm combined with RLS filter. The RLS filter is used for adaptive signal filtering due to its fast convergence speed and low computational burden while the possibility of divergence is known to be rather low (Xu et al., 2002). Since this filter is designed for infinitely repeated task with no final state (*non-episodic* task), this approach is unable to deal with *episodic* tasks.

This work develops a novel NAC algorithm combined with RLS filter for *episodic* tasks. We named this algorithm the "RLS-based eNAC (*episodic* Natural Actor-Critic) algorithm." The RLS-based eNAC algorithm has two separate memory structures: the actor structure and the critic structures. The actor structure determines policies that select actions at each state, and the critic structure criticizes the selected action of the actor structure whether the action is good or not.

In the actor structure, the policy at state $\boldsymbol{s}_t$ in episode e is parameterized as $\pi(\mathbf{a}_t \mid \mathbf{s}_t)=p(\mathbf{a}_t \mid \mathbf{s}_t, \boldsymbol{\psi}_e)$, and policy parameter vector $\boldsymbol{\psi}_e$ is iteratively updated after finishing one episode by the following update rule:

$$\boldsymbol{\psi}_{e+1} \leftarrow \boldsymbol{\psi}_e + \alpha \nabla_{\boldsymbol{\psi}_e} J(\boldsymbol{\psi}_e) \tag{7}$$

, where $J(\boldsymbol{\psi}_e)$ is the objective function to be optimized (*value function* $V^{\pi}(s)$), and $\nabla_{\boldsymbol{\psi}_e} J(\boldsymbol{\psi}_e)$ represents the gradient of objective function $J(\boldsymbol{\psi}_e)$. Peters et al. derived the gradient of the objective function based on the natural gradient method originally proposed by Amari (Amari, 1998). They suggested a simpler update rule by introducing the natural gradient vector w_e as follows:

$$\boldsymbol{\psi}_{e+1} \leftarrow \boldsymbol{\psi}_e + \alpha \nabla_{\boldsymbol{\psi}_e} J(\boldsymbol{\psi}_e) \approx \boldsymbol{\psi}_e + \alpha \mathbf{w}_e \tag{8}$$

, where a denotes the learning rate ($0 \le a \le 1$).

In the critic structure, the least-squares (LS) TD-Q(1) algorithm is used for minimizing the TD error which represents the deviation between the expected return and the current prediction value (Boyan, 1999). By considering the Bellman equation in deriving LSTD-Q(1) algorithm, the action-value function $Q^{\pi}(\mathbf{s}, \mathbf{a})$ can be formulated as follows (Sutton et al., 1998):

$$\begin{aligned} Q^{\pi}(\mathbf{s},\mathbf{a}) &= \sum_{\mathbf{s}'} P_{\mathbf{ss}'}^{\mathbf{a}} \left[R_{\mathbf{ss}'}^{\mathbf{a}} + \gamma V^{\pi}(\mathbf{s}') \right] \\ &= E_{\pi} \left\{ r(\mathbf{s}_t) + \gamma V^{\pi}(\mathbf{s}_{t+1}) \middle| \mathbf{s}_t = \mathbf{s}, \mathbf{a}_t = \mathbf{a} \right\} \end{aligned} \tag{9}$$

Equation (9) can be approximated as

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) \approx r(\mathbf{s}_t) + \gamma V(\mathbf{s}_{t+1}) \tag{10}$$

where $V(\mathbf{s}_{t+1})$ approximates value function $V^{\pi}(\mathbf{s}_{t+1})$ as iterative learning is repeated. Peters *et al*. introduced advantage value function $A^{\pi}(\mathbf{s}, \mathbf{a})=Q^{\pi}(\mathbf{s}, \mathbf{a})-V^{\pi}(\mathbf{s})$ and assumed that the function can be approximated using the compatible function approximation $A^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = \nabla_{\psi_e} \log \pi(\mathbf{a}_t | \mathbf{s}_t)^T \mathbf{w}_e$ (Peters et al., 2003). With this assumption, we can rearrange the discounted summation of (10) for one episode trajectory with N states like below:

$$\begin{aligned} \sum_{t=0}^{N} \gamma^t A^{\pi}(\mathbf{s}_t, \mathbf{a}_t) &= \sum_{t=0}^{N} \gamma^t (Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) - V^{\pi}(\mathbf{s}_t)) \\ \sum_{t=0}^{N} \gamma^t \nabla_{\psi_e} \log \pi(\mathbf{a}_t | \mathbf{s}_t)^T \mathbf{w}_e &= \sum_{t=0}^{N} \gamma^t (r(\mathbf{s}_t, \mathbf{a}_t) + \gamma V(\mathbf{s}_{t+1}) - V(\mathbf{s}_t)) \\ &= \sum_{t=0}^{N} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) + \cancel{\gamma^{N+1} V(\mathbf{s}_{N+1})} - V(\mathbf{s}_0) \\ \sum_{t=0}^{N} \gamma^t \nabla_{\psi_e} \log \pi(\mathbf{a}_t | \mathbf{s}_t)^T \mathbf{w}_e + V(\mathbf{s}_0) &= \sum_{t=0}^{N} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \end{aligned} \tag{11}$$

where the term $\gamma^{N+1}V(\mathbf{s}_{N+1})$ is negligible for its small effect. By letting $V(s_0)$ the product of 1-by-1 critic parameter vector $\mathbf{v}$ and 1-by-1 feature vector [1], we can formulate the following regression equation:

$$\sum_{t=0}^{N} \gamma^t [\nabla_{\psi_e} \log \pi(\mathbf{a}_t | \mathbf{s}_t)^T, 1] \begin{bmatrix} \mathbf{w}_e \\ v_e \end{bmatrix} = \sum_{t=0}^{N} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \tag{12}$$

Here, natural gradient vector $\mathbf{w}_e$ is used for updating policy parameter vector $\boldsymbol{\theta}_e$ (equation (8)), and value function parameter v_e is used for approximating value function $V^{\pi}(\mathbf{s}_t)$. By letting $\tilde{\boldsymbol{\varphi}}_e = \sum_{t=0}^{N} \gamma^t [\nabla_{\psi_e} \log \pi(\mathbf{a}_t | \mathbf{s}_t)^T, 1]^T$, $\boldsymbol{\chi}_e = [\mathbf{w}_e^T, v_e]^T$, and $\tilde{r} = \sum_{t=0}^{N} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)$, we can rearrange equation (12) as a least-square problem as follow:

$$\mathbf{G}_e \boldsymbol{\chi}_e = \mathbf{b}_e \tag{13}$$

where $\mathbf{G}_e = \tilde{\boldsymbol{\varphi}}_e \tilde{\boldsymbol{\varphi}}_e^T$ and $\mathbf{b}_e = \tilde{\boldsymbol{\varphi}}_e \tilde{r}$. Matrix $\mathbf{G}_e$ for episode e is updated to yield the solution vector $\boldsymbol{\chi}_e$ using the following update rule:

$$\begin{aligned} \mathbf{G}_e &\leftarrow \mathbf{G}_e + \delta \mathbf{I} \\ \mathbf{P}_e &= \mathbf{G}_e^{-1} \\ \boldsymbol{\chi}_e &= \mathbf{P}_e \mathbf{b}_e \end{aligned} \tag{14}$$

, where δ is a positive scalar constant, and $\mathbf{I}$ is an identity matrix. By adding the term $\delta\mathbf{I}$ in (14), matrix $\mathbf{G}_e$ becomes diagonally-dominant and non-singular, and thus invertible (Strang, 1988). As the update progresses, the effect of perturbation is diminished. Equation (14) is the conventional form of critic update rule of eNAC algorithm.

The main difference between the conventional eNAC algorithm and the RLS-based eNAC algorithm is the way matrix $\mathbf{G}_e$ and solution vector χ_e are updated. The RLS-based eNAC algorithm employs the update rule of RLS filter. The key feature of RLS algorithm is to exploit $\mathbf{G}_{e\text{-}1}^{-1}$, which already exists, for the estimation of $\mathbf{G}_e^{-1}$. Rather than calculating $\mathbf{P}_e$

(inverse matrix of $\mathbf{G}_e$) using the conventional method for matrix inversion, we used the RLS filter-based update rule as follows: (Moon et al., 2000):

$$\begin{aligned}\mathbf{P}_e &= \frac{1}{\beta}\left(\mathbf{P}_{e-1} - \frac{\mathbf{P}_{e-1}\tilde{\boldsymbol{\varphi}}_e\tilde{\boldsymbol{\varphi}}_e^T\mathbf{P}_{e-1}}{\beta+\tilde{\boldsymbol{\varphi}}_e^T\mathbf{P}_{e-1}\tilde{\boldsymbol{\varphi}}_e}\right)\\ \mathbf{k}_e &= \frac{\mathbf{P}_{e-1}\tilde{\boldsymbol{\varphi}}_e}{\beta+\tilde{\boldsymbol{\varphi}}_e^T\mathbf{P}_{e-1}\tilde{\boldsymbol{\varphi}}_e}\\ \boldsymbol{\chi}_e &= \boldsymbol{\chi}_{e-1} + \mathbf{k}_e(\tilde{r}_e - \boldsymbol{\varphi}_e^T\boldsymbol{\chi}_{e-1})\end{aligned} \tag{15}$$

Where forgetting factor β ($0 \le \beta \le 1$) is used to accumulate the past information in a discount manner. By using the RLS-filter based update rule, the inverse matrix can be calculated without too much computational burden. This is repeated until the solution vector $\boldsymbol{\chi}_e$ converges. It should be noted, however, that matrix $\mathbf{G}_e$ should be obtained using (14) for the first episode (episode 1).

The entire procedure of the RLS-based episodic Natural Actor-Critic algorithm is summarized in Table 1.

Initialize each parameter vector:
$\boldsymbol{\psi} = \boldsymbol{\psi}_0, \mathbf{G} = \mathbf{0}, \mathbf{P} = \mathbf{0}, \mathbf{b} = \mathbf{0}, \mathbf{s}_t = \mathbf{0}$
for each episode,
Run simulator:
for each step,
Take action a_t, from stochastic policy π,
then, observe next state $\mathbf{s}_{t+1}$, reward r_t.
end
Update Critic structure:
if first update,
Update critic information matrices,
following the initial update rule in (13) (14).
else
Update critic information matrices,
following the recursive least-squares update rule in (15).
end
Update Actor structure:
Update policy parameter vector following the rule in (8).
repeat until converge

Table 1. RLS-based episodic Natural Actor-Critic algorithm

3.3 Stochastic Action Selection

As discussed in section 2, the characteristics of stiffness ellipse can be changed by modulating three parameters: the magnitude, shape, and orientation. For performing contact tasks, policy π is designed to plan the change rate of the magnitude, shape, and orientation of stiffness ellipse at each state by taking actions ($\mathbf{a}_t = [a_{mag},\ a_{shape},\ a_{orient}]^T$). Through the sequence of an episode, policy π determines those actions. The goal of the

learning algorithm is to find the trajectory of the optimal stiffness ellipse during the episode. Fig. 4 illustrates the change of stiffness ellipse corresponding to each action.
Policy π is in the form of Gaussian density function. In the critic structure, compatible function approximation $\nabla_{\psi} \log \pi(\mathbf{a}_t|\mathbf{s}_t)^T \mathbf{w}$ is derived from the stochastic policy π based on the algorithm suggested by Williams (Williams, 1992). Policy π for each component of action vector $\mathbf{a}_t$ can be described as follows:

$$\pi(a_t|\mathbf{s}_t) = N(a_t|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(a_t-\mu)^2}{2\sigma^2}) \tag{16}$$

Since the stochastic action selection is dependent on the conditions of Gaussian density function, the action policy can be designed by controlling the mean μ and the variance σ of equation (16). These variables are defined as:

$$\mu = \xi \boldsymbol{\omega}^T \mathbf{s}_t$$
$$\sigma = \tilde{\xi}\left(0.001 + \frac{1}{1+\exp(-\eta)}\right)$$

Since the stochastic action selection is dependent on the conditions of Gaussian density function, the action policy can be designed by controlling the mean μ and the variance σ of equation (16). These variables are defined as:

$$\mu = \xi \boldsymbol{\omega}^T \mathbf{s}_t$$
$$\sigma = \tilde{\xi}\left(0.001 + \frac{1}{1+\exp(-\eta)}\right) \tag{17}$$

As can be seen in the equation, mean μ is a linear combination of the vector $\boldsymbol{\omega}$ and state vector $\boldsymbol{s}_t$ with mean scaling factor ξ. Variance σ is in the form of the sigmoid function with the positive scalar parameter η and variance scaling factor $\tilde{\xi}$. As a two-link manipulator model is used in this study, the components of state vector $\boldsymbol{s}_t$ include the joint angles and velocities of shoulder and elbow joints ($\mathbf{s}_t = [x_1, x_2, \dot{x}_1, \dot{x}_2, 1]^T$, where the fifth component of 1 is a bias factor). Therefore, natural gradient vector $\mathbf{w}$ (and thus policy parameter vector $\boldsymbol{\psi}$) is composed of 18 components as follows:

$$\mathbf{w} = [\boldsymbol{\omega}_{mag}^T, \boldsymbol{\omega}_{shape}^T, \boldsymbol{\omega}_{orient}^T, \eta_{mag}, \eta_{shape}, \eta_{orient}]^T$$

, where, $\boldsymbol{\omega}_{mag}$, $\boldsymbol{\omega}_{shape}$, and $\boldsymbol{\omega}_{orient}$ are 5-by-1 vectors and η_{mag}, η_{shape}, and η_{orient} are parameters corresponding to the three components of action vector ($\mathbf{a}_t = [a_{mag}, a_{shape}, a_{orient}]^T$).

4. Contact Task Applications

In this section, the method developed in the previous section is applied to two contact tasks: point-to-point movement in an unknown force field, and catching a flying ball. The two-link manipulator developed in Section 3 was used to perform the tasks in two-dimensional

space. The dynamic simulator is constructed by MSC.ADAMS2005, and the control algorithm is implemented using Matlab/Simulink (Mathworks, Inc.).

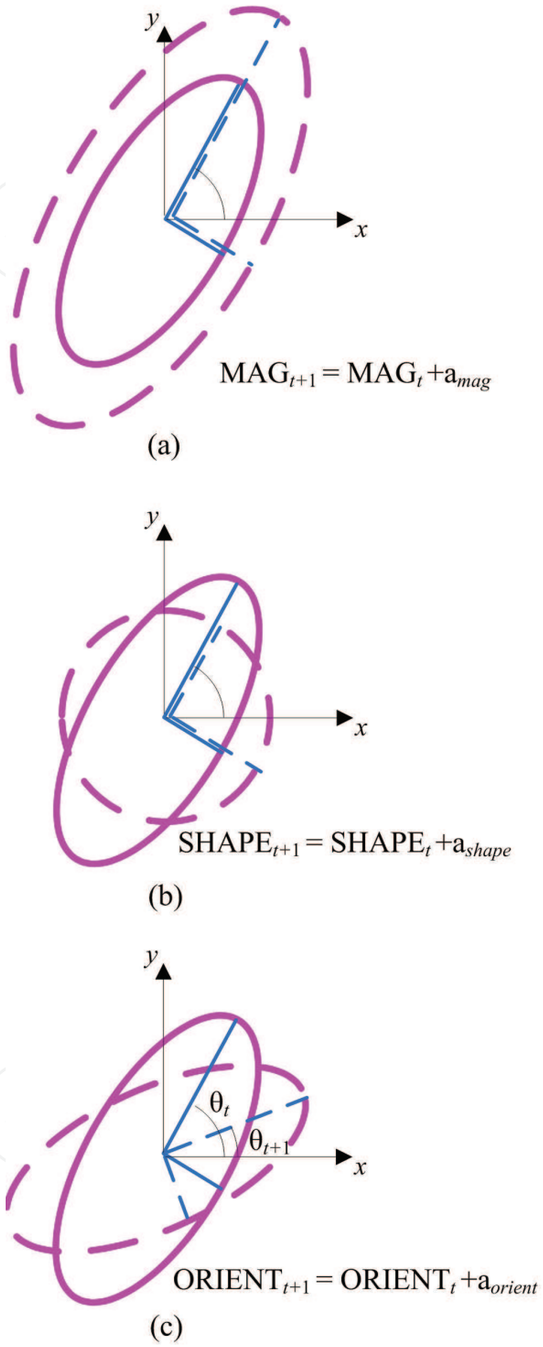

Figure 4. Some variation examples of stiffness ellipse. (a) The magnitude (area of ellipse) is changed (b) The shape (length ratio of major and minor axes) is changed (c) The orientation (direction of major axis) is changed (solid line: stiffness ellipse at time *t*, dashed line: stiffness ellipse at time *t*+1)

4.1 Point-to-Point Movement in an Unknown Force Field

In this task, the two-link robotic manipulator makes a point-to-point movement in an unknown force field. The velocity-dependent force is given as:

$$\mathbf{F}_{viscous} = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix} \begin{Bmatrix} \dot{x} \\ \dot{y} \end{Bmatrix} \tag{18}$$

The movement of the endpoint of the manipulator is impeded by the force proportional to its velocity, as it moves to the goal position in the force field. This condition is identical to that of biomechanics study of Kang et al. (Kang et al., 2007). In their study, the subject moves ones hand from a point to another point in a velocity-dependent force field, which is same as equation (18), where the force field was applied by an special apparatus (KINARM, BKIN Technology).

For this task, the performance indices are chosen as follows:

1. The root mean square of the difference between the desired position (virtual trajectory) and the actual position of the endpoint (Δ_{rms})
2. The magnitude of the time rate of torque vector of two arm joints ($\|\dot{\boldsymbol{\tau}}\| = \sqrt{\dot{\tau}_1^{\,2} + \dot{\tau}_2^{\,2}}$)

The torque rate represents power consumption, which can also be interpreted as metabolic costs for human arm movement (Franklin et al., 2004; Uno et al., 1989). By combining the two performance indices, two different rewards are formulated for one episode as follows:

$$reward1 = \kappa_1 - \sum_{t=1}^{N} (\Delta_{rms})_t$$

$$reward2 = w_1 \left(\kappa_1 - \sum_{t=1}^{N} (\Delta_{rms})_t \right) + w_2 \left(\kappa_2 - \sum_{t=1}^{N} \|\dot{\boldsymbol{\tau}}\|_t \right)$$

, where w_1 and w_2 are the weighting factors, and κ_1 and κ_2 are constants. The reward is a weighted linear combination of time integrals of two performance indices.

The learning parameters were chosen as follows: α = 0.05, β = 0.99, γ = 0.99. The change limits for action are set as [-10, 10] degrees for the orientation, [-2, 2] for the major/minor ratio, and [-200π, 200π] for the area of stiffness ellipse. The initial ellipse before learning was set to be circular with the area of 2500π.

The same physical properties as in (Kang et al., 2007) were chosen for dynamic simulations (Table 2).

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

Table 2. Physical properties of two link arm model

Fig. 5 shows the change of stiffness ellipse trajectory before and after learning. Before learning, the endpoint of the manipulator was not even able to reach the goal position (Fig. 5 (a)). Figs. 5 (b) and (c) compare the stiffness ellipse trajectories after learning using two different rewards (*reward1* and *reward2*). As can be seen in the figures, for both rewards the major axis of stiffness ellipse was directed to the goal position to overcome resistance of viscous force field.

Fig. 6 compares the effects of two rewards on the changes of two performance indices (Δ_{rms} and $\|\dot{\boldsymbol{\tau}}\|$) as learning iterates. While the choice of reward does not affect the time integral of Δ_{rms}, the time integral of $\|\dot{\boldsymbol{\tau}}\|$ was suppressed considerably by using *reward2* in learning.

The results of dynamic simulations are comparable with the biomechanics study of Kang et al.. The results of their study suggest that the human actively modulates the major axis toward the direction of the external force against the motion, which is in accordance with our results.

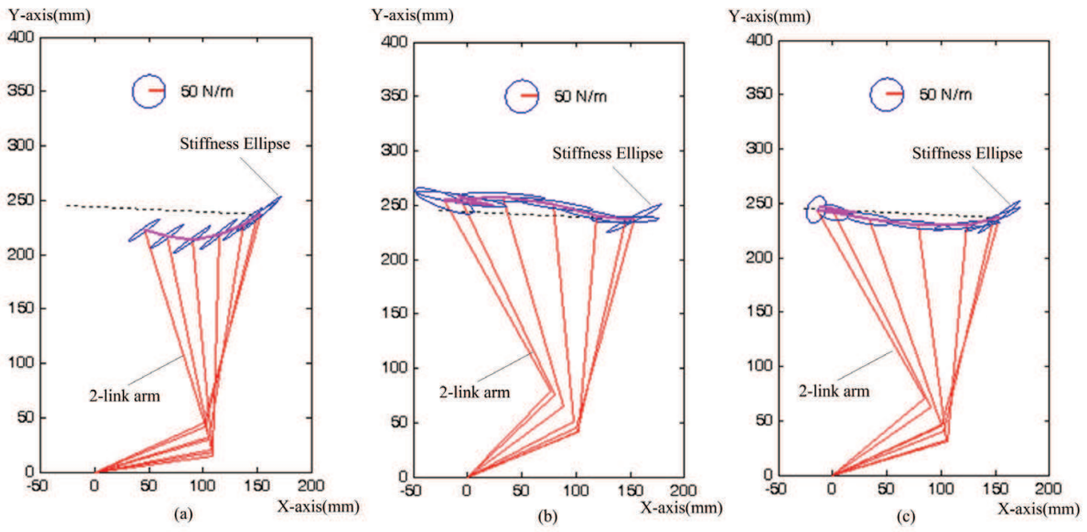

Figure 5. Stiffness ellipse trajectories. (dotted line: virtual trajectory, solid line: actual trajectory). (a) Before learning. (b) After learning (*reward1*). (c) After learning (*reward2*)

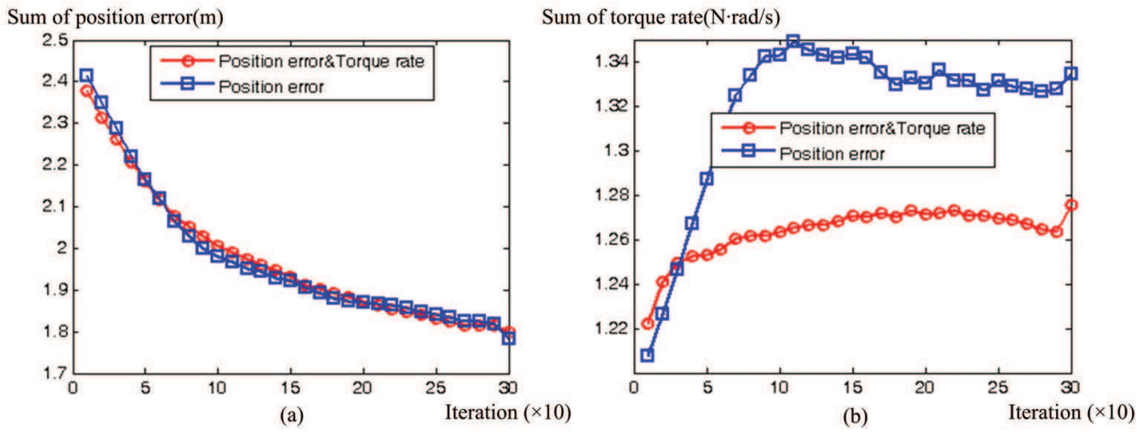

Figure 6. Learning effects of performance indices (average of 10 learning trial). (a) position error. (b) torque rate

4.2 Catching a Flying Ball

In this task, the two-link robotic arm catches a flying ball illustrated in Fig. 7. The simulation was performed using the physical properties of the arm as listed in Table 2. The main issues

in ball-catching task would be how to detect the ball trajectory and how to reduce the impulsive force between the ball and the end-effector. This work focuses on the latter and assumes that the ball trajectory is known in advance.

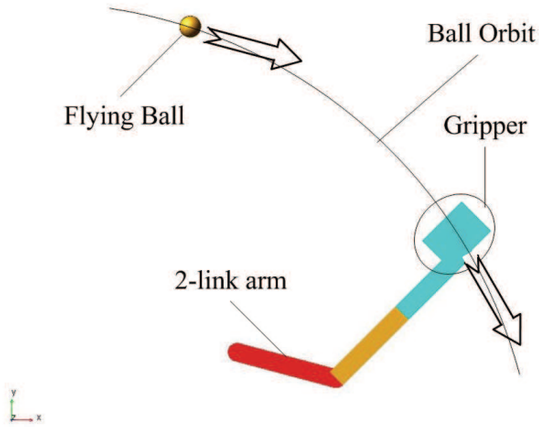

Figure 7. Catching a flying ball

When a human catches a ball, one moves ones arm backward to reduce the impulsive contact force. By considering the human ball-catching, the task is modeled as follow: A ball is thrown to the end-effector of the robot arm. The time for the ball to reach the end-effector is approximately 0.8sec. After the ball is thrown, the arm starts to move following the parabolic orbit of the flying ball. While the end-effector is moving, the ball is caught and then moves to the goal position together. The robot is set to catch the ball when the end-effector's moving at its highest speed to reduce the impulsive contact force between the ball and the end-effector. The impulsive force can also be reduced by modulating the stiffness ellipse during the contact.

The learning parameters were chosen as follows: a = 0.05, β = 0.99, γ = 0.99. The change limits for action are set as [-10, 10] degrees for the orientation, [-2, 2] for the major/minor ratio, and [-200π, 200π] for the area of stiffness ellipse. The initial ellipse before learning was set to be circular with the area of 10000 π.

For this task, the contact force is chosen as the performance index:

$$\left\|\mathbf{F}_{contact}\right\| = \sqrt{F_x^{\,2} + F_y^{\,2}}$$

The reward to be maximized is the impulse (time integral of contact force) during contact:

$$reward = \kappa - \sum_{t=1}^{N} \left\|\mathbf{F}_{contact}\right\|_t \Delta t_t$$

where κ is a constant. Fig. 8 illustrates the change of stiffness during contact after learning. As can be seen in the figure, the stiffness is tuned soft in the direction of ball trajectory, while the stiffness normal to the trajectory is much higher. Fig. 9 shows the change of the impulse as learning continues. As can be seen in the figure, the impulse was reduced considerably after learning.

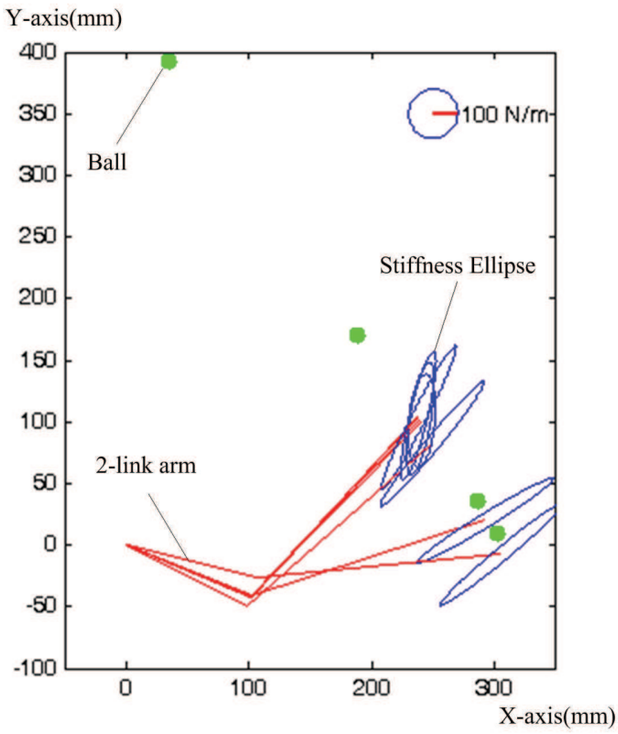

Figure 8. Catching a flying ball

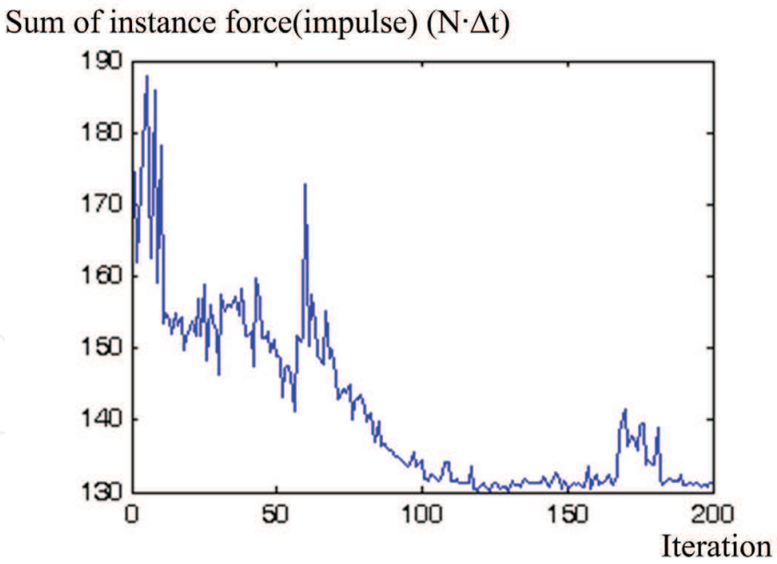

Figure 9. Catching a flying ball

5. Conclusions

Safety in robotic contact tasks has become one of the important issues as robots spread their applications to dynamic, human-populated environments. The determination of impedance

control parameters for a specific contact task would be the key feature in enhancing the robot performance. This study proposes a novel motor learning framework for determining impedance parameters required for various contact tasks. As a learning framework, we employed reinforcement learning to optimize the performance of contact task. We have demonstrated that the proposed framework enhances contact tasks, such as door-opening, point-to-point movement, and ball-catching.

In our future works we will extend our method to apply it to teach a service robot that is required to perform more realistic tasks in three-dimensional space. Also, we are currently investigating a learning method to develop motor schemata that combine the internal models of contact tasks with the actor-critic algorithm developed in this study.

6. References

Amari, S. (1998). Natural gradient works efficiently in learning, *Neural Computation*, Vol. 10, No. 2, pp. 251-276, ISSN 0899-7667

Asada, H. & Slotine, J-J. E. (1986). *Robot Analysis and Control*, John Wiley & Sons, Inc., ISBN 978-0471830290

Boyan. J. (1999). Least-squares temporal difference learning, *Proceeding of the 16th International Conference on Machine Learning*, pp. 49-56

Cohen, M. & Flash, T. (1991). Learning impedance parameters for robot control using associative search network, *IEEE Transactions on Robotics and Automation*, Vol. 7, Issue. 3, pp. 382-390, ISSN 1042-296X

Engel, Y.; Mannor, S. & Meir, R. (2003). Bayes meets bellman: the gaussian process approach to temporal difference learning, *Proceeding of the 20th International Conference on Machine Learning*, pp. 154-161

Flash, T. & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model, *Journal of Neuroscience*, Vol. 5, No. 7, pp. 1688-1703, ISSN 1529-2401

Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movement, *Biological Cybernetics*, Vol. 57, No. 4-5, pp. 257-274, ISSN 1432-0770

Franklin, D. W.; So, U.; Kawato, M. & Milner, T. E. (2004). Impedance control balances stability with metabolically costly muscle activation, *Journal of Neurophysiology*, Vol. 92, pp. 3097-3105, ISSN 0022-3077

Hogan, N. (1985). Impedance control: An approach to manipulation: part I. theory, part II. implementation, part III. application, *ASME Journal of Dynamic System, Measurement, and Control*, Vol. 107, No. 1, pp. 1-24, ISSN 0022-0434

Izawa, J.; Kondo, T. & Ito, K. (2002). Biological robot arm motion through reinforcement learning, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3398-3403, ISBN 0-7803-7272-7

Jung, S.; Yim, S. B. & Hsia, T. C. (2001). Experimental studies of neural network impedance force control of robot manipulator, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3453-3458, ISBN 0-7803-6576-3

Kang, B.; Kim, B.; Park, S. & Kim, H. (2007). Modeling of artificial neural network for the prediction of the multi-joint stiffness in dynamic condition, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1840-1845, ISBN 978-1-4244-0912-9, San Diego, CA, USA

Kazerooni, H.; Houpt, P. K. & Sheridan, T. B. (1986). The fundamental concepts of robust compliant motion for robot manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 418-427, MN, USA

Lipkin, H. & Patterson, T. (1992). Generalized center of compliance and stiffness, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1251-1256, ISBN 0-8186-2720-4, Nice, France

Moon, T. K. & Stirling, W. C. (2000). *Mathematical Methods and Algorithm for Signal Processing*, Prentice Hall, Upper Saddle River, NJ, ISBN 0-201-36186-8

Park, J.; Kim, J. & Kang, D. (2005). An rls-based natural actor-critic algorithm for locomotion of a two-linked robot arm, *Proceeding of International Conference on Computational Intelligence and Security, Part I, LNAI,* Vol. 3801, pp. 65-72, ISSN 1611-3349

Park, S. & Sheridan, T. B. (2004). Enhanced human-machine interface in braking, *IEEE Transactions on Systems, Man, and Cybernetics, - Part A: Systems and Humans*, Vol. 34, No. 5, pp. 615-629, ISSN 1083-4427

Peters, J.; Vijayakumar, S. & Schaal, S. (2005). Natural actor-critic, *Proceeding of the 16th European Conference on Machine Learning*, *LNCS*, Vol. 3720, pp.280-291, ISSN 1611-3349

Peters J. & Schaal, S. (2006). Policy gradient methods for robotics, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2219-2225, ISBN 1-4244-0259-X, Beijing, China

Strang, G. (1988). *Linear Algebra and Its Applications*, Harcourt Brace & Company

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, MIT Press, ISBN 0-262-19398-1

Tsuji, T.; Terauchi, M. & Tanaka, Y. (2004). Online learning of virtual impedance parameters in non-contact impedance control using neural networks, *IEEE Transactions on Systems, Man, and Cybernetics, - Part B: Cybernetics*, Vol. 34, Issue 5, pp. 2112-2118, ISSN 1083-4419

Uno, Y.; Suzuki, R. & Kawato, M. (1989). Formation and control of optimal trajectory in human multi-joint arm movement: minimum torque change model, *Biological Cybernetics*, Vol. 61, No. 2, pp. 89-101, ISSN 1432-0770

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learning*, Vol. 8, No. 3-4, pp. 229-256. ISSN 1573-0565

Won, J. (1993). The Control of Constrained and Partially Constrained Arm Movement, *S. M. Thesis*, Department of Mechanical Engineering, MIT, Cambridge, MA, 1993.

Xu, X.; He, H. & Hu, D. (2002). Efficient reinforcement learning using recursive least-squares methods, *Journal of Artificial Intelligent Research*, Vol. 16, pp. 259-292

Robot Manipulators

Edited by Marco Ceccarelli

ISBN 978-953-7619-06-0

Hard cover, 546 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

In this book we have grouped contributions in 28 chapters from several authors all around the world on the several aspects and challenges of research and applications of robots with the aim to show the recent advances and problems that still need to be considered for future improvements of robot success in worldwide frames. Each chapter addresses a specific area of modeling, design, and application of robots but with an eye to give an integrated view of what make a robot a unique modern system for many different uses and future potential applications. Main attention has been focused on design issues as thought challenging for improving capabilities and further possibilities of robots for new and old applications, as seen from today technologies and research programs. Thus, great attention has been addressed to control aspects that are strongly evolving also as function of the improvements in robot modeling, sensors, servo-power systems, and informatics. But even other aspects are considered as of fundamental challenge both in design and use of robots with improved performance and capabilities, like for example kinematic design, dynamics, vision integration.

IntechOpen