An RLS-Based Natural Actor-Critic Algorithm for Locomotion of a Two-Linked Robot Arm

Jooyoung Park, Jongho Kim, and Daesung Kang

Department of Control and Instrumentation Engineering, Korea University, Jochiwon, Chungnam 339-700, Korea
{parkj, oyeasw, mpkds}@korea.ac.kr

Abstract. Recently, actor-critic methods have drawn much interests in the area of reinforcement learning, and several algorithms have been studied along the line of the actor-critic strategy. This paper studies an actor-critic type algorithm utilizing the RLS(recursive least-squares) method, which is one of the most efficient techniques for adaptive signal processing, together with natural policy gradient. In the actor part of the studied algorithm, we follow the strategy of performing parameter update via the natural gradient method, while in its update for the critic part, the recursive least-squares method is employed in order to make the parameter estimation for the value functions more efficient. The studied algorithm was applied to locomotion of a two-linked robot arm, and showed better performance compared to the conventional stochastic gradient ascent algorithm.

1 Introduction

Recently, actor-critic methods have drawn much interests in the areas of reinforcement learning, and several algorithms have been studied along the line of the actor-critic strategy. In the actor-critic methods, a separate memory structure is used to explicitly represent the policy independent of the value functions, and the policy structure is known as the actor, because it is used to select actions, while the part handling estimated value functions is called the critic, because it criticizes the actions made by the actor [1]. Among the various implementations of the actor-critic algorithms, particularly pertinent to this paper is the results on the natural actor-critic algorithm [2], which show that the actor-critic algorithm using the natural policy gradient is significantly better than other algorithms belonging to greedy methods or vanilla policy gradient methods, and can be a promising route to develop a reinforcement learning method for truly high-dimensional state-action systems. In this paper, we address the problem of modifying the natural actor-critic algorithm of [2] toward the use of the RLS(recursive least-squares) method, which is one of the most efficient techniques for adaptive signal processing, to estimate the critic parameters recursively, and also apply the resulting algorithm, which will be called the RLS-based natural actor-critic algorithm in this paper, to locomotion of a two-linked robot arm. A previous work on the use of the RLS method for reinforcement learning is

Y. Hao et al. (Eds.): CIS 2005, Part I, LNAI 3801, pp. 65–72, 2005.

the so-called RLS-TD(λ) [3], which shows how to convert existing reinforcement learning algorithms to recursively updated versions utilizing the RLS method.

The remaining parts of this paper are organized as follows: In Section 2, after briefly describing about the actor part of the natural actor-critic algorithm, we report on how the recursive least-squares method can be employed for the estimation of the critic parameters. Section 3 shows the applicability of the RLS-based natural actor-critic algorithm via an example dealing with locomotion of a two-linked robot arm. Finally, in Section 4, concluding remarks are given.

2 RLS-Based Natural Actor-Critic Method

In this paper, we consider a discounted reward reinforcement learning problem [1] with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, rewards $r \in \Re$, and time steps $t \in \{0, 1, 2, \cdots\}$, in which a learning agent interacts with an MDP(Markov decision process). As usual, the environment's dynamics are characterized by state transition probabilities $p(s'|s,a) \stackrel{\triangle}{=} Pr\{s_{t+1} = s'|s_t = s, a_t = a\}$, and expected rewards $r(s,a) \stackrel{\triangle}{=} E\{r_t|s_t = s, a_t = a\}$. The objective of the learning agent is to pursue a policy that can maximize the discounted sum of rewards $J(\pi) \stackrel{\triangle}{=} E\{\sum_{k=0}^{\infty} \gamma^k r_k | s_0, \pi\}$, where $\gamma \in (0,1)$ is the discount rate, r_k is the immediate reward observed after the state transition from state s_k to s_{k+1}, s_0 is a designated start state, and π denotes the policy from which actions are chosen. In general, the policy is described as a conditional probability $\pi(a|s) \stackrel{\triangle}{=} Pr\{a_t = a|s_t = s\}$. Note that by introducing the state value function $V^\pi(s) \stackrel{\triangle}{=} E\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, \pi\}$, the action value function $Q^\pi(s,a) \stackrel{\triangle}{=} E\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a, \pi\}$, together with the so-called discounted state distribution [4] $d^\pi(s) \stackrel{\triangle}{=} \sum_{k=0}^{\infty} \gamma^k Pr\{s_k = s|s_0, \pi\}$, we can rewrite the objective function in the following form[1]:

$$J(\pi) = V^\pi(s_0) = \sum_a \pi(a|s_0) Q^\pi(s_0, a) = \sum_s d^\pi(s) \sum_a \pi(a|s) r(s,a).$$

The real essence of the actor-critic methods is in using separate parametrized families for the policy distribution $\pi(a|s)$ and approximators for the value functions. In the following, we describe the actor part updated via the natural gradient method [2], and then address on how the recursive least-squares method can be employed for the estimation of its critic part parameters. The resultant algorithm will be applied to the locomotion of a two-linked robot arm later in this paper.

2.1 Actor

The main role of the actor is to generate actions via a parametrized family. At each state $s \in \mathcal{S}$, an action $a \in \mathcal{A}$ is drawn in accordance with the conditional distribution $\pi_\theta(a|s)$, where θ is the parameter vector characterizing the

[1] Dealing with continuous states and actions requires the corresponding summations changed into integral representation. Throughout this paper, the summation representation is used for notational simplicity.

distribution. Thus, the objective we seek to maximize can be written as follows: $J(\pi) = J(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) r(s,a)$. One of the convenient strategies for seeking the best distribution parameter θ is to utilize the direction of $\nabla_\theta J(\theta)$, which is often called the policy gradient. Utilizing the famous policy gradient theorem [4], [5] along with the equality $\sum_a \nabla_\theta \pi(s,a) = 0$ for $\forall s \in \mathcal{S}$, the policy gradient can be written as follows:

$$\begin{aligned} &\nabla_\theta J(\theta) = \textstyle\sum_s d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \\ &= \textstyle\sum_s d^{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s) (Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)) \\ &= \textstyle\sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a), \end{aligned} \tag{1}$$

where $A^{\pi_\theta}(s,a) \stackrel{\triangle}{=} Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$ is the advantage value function which gives the advantage of action a over the average value in a state s. According to a remarkable observation introduced in [4] and [5], the advantage value function $A^{\pi_\theta}(s,a)$ can be replaced by the so-called compatible approximator[2]

$$\tilde{A}_w(s,a) \stackrel{\triangle}{=} \nabla_\theta \log \pi_\theta(a|s)^T w \tag{2}$$

without affecting the unbiasedness of the gradient estimate. Note that the compatible approximator $\tilde{A}_w(s,a)$ is linear with respect to the parameter vector w. Based on equations (1) and (2), we see that a desirable function form for an estimate of the policy gradient can be given as follows:

$$\begin{aligned} &\nabla_\theta J(\theta) = \textstyle\sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a) \\ &\approx \textstyle\sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s) \tilde{A}_w(s,a) = F(\theta) w, \end{aligned}$$

where $F(\theta) \stackrel{\triangle}{=} \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^T$. Among a variety of powerful gradient based algorithms, one of the most efficient tools for updating the parameter vector θ of $\pi_\theta(a|s)$ would be the natural gradient method [6]. When an objective function is of the form $L(\theta) = \sum_s p(s) l(s,\theta)$, where $p(s)$ is a probability mass function, and its gradient $\nabla_\theta L(\theta)$ is given, the natural gradient of L, which is denoted by $\tilde{\nabla} L(\theta)$, equals the steepest ascent in a Riemannian space with respect to the Fisher information metric $G(\theta) \stackrel{\triangle}{=} \sum_s p(s) \nabla_\theta \log p(s) \nabla_\theta \log p(s)^T$, *i.e.*, $\tilde{\nabla}_\theta L(\theta) = G^{-1}(\theta) \nabla_\theta L(\theta)$. In a recent remarkable paper of Peters *et al.* [2], it was shown that for the reinforcement learning problem, matrix $F(\theta)$ is exactly the same with the Fisher information matrix, thus the natural gradient of $J(\theta)$ can be estimated using the following: $\tilde{\nabla}_\theta J(\theta) = G^{-1}(\theta) \nabla J_\theta(\theta) \approx G^{-1}(\theta) F(\theta) w = w$. This result enables us to update the actor parameter θ via the following simple rule:

$$\theta \leftarrow \theta + \alpha \tilde{\nabla}_\theta J(\theta) \approx \theta + \alpha w, \tag{3}$$

where $\alpha > 0$ is the learning rate. Note that the actor part described above is that of the natural actor-critic algorithm in [2]. Also, note that in practical applications of the update rule (3), the use of upper bound for the magnitude of each actor parameter may be often desirable.

[2] In this paper, we assume that the policy distribution π_θ is such that the gradient $\nabla_\theta \log \pi_\theta(a|s)^T$ is well-defined.

2.2 Critic

As mentioned before, the essence of the actor-critic methods is in using separate parametrized families for the actor part which is represented by the policy distribution $\pi_\theta(a|s)$, and the critic part which is represented by value functions. For the parametrized families for the critic part, this paper employs the linear functions $\tilde{A}_w(s,a) \triangleq \nabla_\theta \log \pi_\theta(a|s)^T w$ and $\tilde{V}_v(s) \triangleq \phi^T(s)v$, which approximate the advantage value function $A^{\pi_\theta}(s,a)$ and the state value function $V^{\pi_\theta}(s)$ for action-generating policy π_θ, respectively. From the Bellman equation [1] $Q^{\pi_\theta}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a)V^{\pi_\theta}(s')$, we see that through a sampled trajectory, $Q^{\pi_{\theta_k}}(s_k,a_k)$ can be approximated by $r_k + \gamma V^{\pi_{\theta_k}}(s_{k+1})$; thus $r_k + \gamma \tilde{V}_v(s_{k+1})$ is a valid estimate for the $Q^{\pi_{\theta_k}}(s_k,a_k)$. Also from $Q^{\pi_\theta}(s,a) = A^{\pi_\theta}(s,a) + V^{\pi_\theta}(s)$ and the usual strategy using the eligibility trace [1], we see that in order for the approximators $\tilde{A}_w(s,a)$ and $\tilde{V}_v(s)$ to be useful in the t-th time step, it is desirable to minimize the following:

$$\begin{aligned} &\Psi_t(v,w) \triangleq \| \textstyle\sum_{k=0}^{t} z_k[(\tilde{V}_v(s_k) + \tilde{A}_w(s_k,a_k)) - (r_k + \gamma \tilde{V}_v(s_{k+1})]\|^2 = \\ &\| \textstyle\sum_{k=0}^{t} z_k[\phi^T(s_k) - \gamma\phi^T(s_{k+1}), \nabla_\theta \log \pi(a_k|s_k)^T] \begin{bmatrix} v \\ w \end{bmatrix} - \sum_{k=0}^{t} z_k r_k\|^2, \end{aligned} \tag{4}$$

where z_k is the eligibility trace vector defined via

$$\begin{aligned} &z_k = \gamma\lambda z_{k-1} + [\phi^T(s_k), \nabla_\theta \log \pi(a_k|s_k)^T]^T \text{ for } k \geq 1, \\ &z_0 = [\phi^T(s_0), \nabla_\theta \log \pi(a_0|s_0)^T]^T, \end{aligned}$$

and $\lambda \in [0,1]$ is the trace-decay parameter. Note that minimizing (4) is simply a least-squares problem utilizing the entire history of agent-environment interactions up to the t-th time step. When there is a need to put more emphasis on recent observations, the use of the so-called forgetting factor $\beta \in (0,1)$ is desirable. In this case, the following needs to be used instead of (4): $\tilde{\Psi}_t(v,w) \triangleq \|M_t \begin{bmatrix} v \\ w \end{bmatrix} - b_t\|^2$, where $M_t \triangleq \sum_{k=0}^{t} \beta^{t-k} z_k[\phi^T(s_k) - \gamma\phi^T(s_{k+1}), \nabla_\theta \log \pi(a_k|s_k)^T]$, and $b_t \triangleq \sum_{k=0}^{t} \beta^{t-k} z_k r_k$. Note that for $t \geq 1$, the above M_t and b_t can be written in the following recursive form: $M_t = \beta M_{t-1} + z_t[\phi^T(s_t) - \gamma\phi^T(s_{t+1}), \nabla_\theta \log \pi(a_t|s_t)^T]$, $b_t = \beta b_{t-1} + z_t r_t$. Also, note that when M_t is invertible, the optimal solution to the problem of minimizing $\tilde{\Psi}_t(v,w)$ is obviously $\begin{bmatrix} v_t \\ w_t \end{bmatrix} = M_t^{-1} b_t$. However, M_t is usually not invertible until a sufficient number of samples have been included in its summation. A common strategy used in the recursive least-squares method for ensuring the invertibility of M_t is to use δI for its initialization [3]. Employing the strategy, we use $M_0 = \delta I + z_0[\phi^T(s_0) - \gamma\phi^T(s_1), \nabla_\theta \log \pi(a_0|s_0)^T]$, where δ is a positive number, instead of $M_0 = z_0[\phi^T(s_0) - \gamma\phi^T(s_1), \nabla_\theta \log \pi(a_0|s_0)^T]$. Now, by applying the matrix inversion formula [7] $(A + XY)^{-1} = A^{-1} - A^{-1}X(I + YA^{-1}X)^{-1}YA^{-1}$ to the equation for M_t, we can obtain a recursive update rule for M_t, and also the following procedure for an approximate solution to minimizing $\tilde{\Psi}_t(v,w)$: Let

$$
\begin{aligned}
&z_0 \stackrel{\triangle}{=} [\phi^T(s_0), \nabla_\theta \log \pi(a_0|s_0)^T]^T, \\
&M_0 \stackrel{\triangle}{=} \delta I + z_0[\phi^T(s_0) - \gamma\phi^T(s_1), \nabla_\theta \log \pi(a_0|s_0)^T], \\
&M_t \stackrel{\triangle}{=} \beta M_{t-1} + z_t[\phi^T(s_t) - \gamma\phi^T(s_{t+1}), \nabla_\theta \log \pi(a_t|s_t)^T] \text{ for } t \geq 1, \\
&P_t \stackrel{\triangle}{=} M_t^{-1} \text{ for } t \geq 0, \text{ and } K_t \stackrel{\triangle}{=} P_t z_t \text{ for } t \geq 0.
\end{aligned}
$$

Then with the update rules

$$
\begin{aligned}
z_t &= \gamma\lambda z_{t-1} + [\phi^T(s_t),\ \nabla_\theta \log \pi(a_t|s_t)^T]^T, \\
P_t &= \frac{1}{\beta}\left(P_{t-1} - \frac{P_{t-1}z_t\left[\phi^T(s_t) - \gamma\phi^T(s_{t+1}),\ \nabla_\theta \log \pi(a_t|s_t)^T\right]P_{t-1}}{\beta + [\phi^T(s_t) - \gamma\phi^T(s_{t+1}),\ \nabla_\theta \log \pi(a_t|s_t)^T]P_{t-1}z_t}\right), \\
K_t &= \frac{P_{t-1}z_t}{\beta + [\phi^T(s_t) - \gamma\phi^T(s_{t+1}),\ \nabla_\theta \log \pi(a_t|s_t)^T]P_{t-1}z_t},
\end{aligned}
\tag{5}
$$

the solution for the critic parameters at time t can be obtained by the following recursive equation:

$$
\begin{aligned}
&\begin{bmatrix} v_t \\ w_t \end{bmatrix} = \begin{bmatrix} v_{t-1} \\ w_{t-1} \end{bmatrix} \\
&+ K_t(r_t - \left[\phi^T(s_t) - \gamma\phi^T(s_{t+1}),\ \nabla_\theta \log \pi(a_t|s_t)^T\right]\begin{bmatrix} v_{t-1} \\ w_{t-1} \end{bmatrix}).
\end{aligned}
\tag{6}
$$

Note that the resultant recursive least-squares solution w_t will be used in the update process for the actor part via (3).

2.3 Algorithm

The algorithm considered in this paper repeats two tasks: An agent-environment interaction task in which the agent interacts with its environment with an action generated by the current policy and observes the consequence of the interaction, and a task for the estimation and improvement in which the agent optimizes its policy by updating the actor and critic parameters on the basis of the natural gradient and the recursive least-squares method. More precisely, the RLS-based natural actor-critic algorithm can be summarized as follows:

Given:

- Initial state s_0
- Parametrized policy $\pi_\theta(a|s)$ with its initial parameter vector $\theta = \theta_0$, and derivative $\nabla_\theta \log \pi_\theta(a|s)$
- Basis functions $\phi(s) \stackrel{\triangle}{=} [\phi_1(s) \cdots \phi_K(s)]^T$ in use for $\tilde{V}_v(s) \stackrel{\triangle}{=} \phi(s)^T v$, which approximates the state value function
- Learning rate $\alpha > 0$ for updating the actor parameter θ
- Forgetting factor $\beta \in (0, 1)$
- Discount rate $\gamma \in (0, 1)$
- Trace-decay parameter $\lambda \in [0, 1]$
- Constant $\delta > 0$
- Bound M for limiting the magnitude of each actor parameter

Goal:

- Parameter vectors θ_t which tend to that of a locally optimal policy distribution $\pi_\theta(a|s)$
- Parameter vectors v_t and w_t which yield good approximation for $\tilde{V}_{v_t}$ and $\tilde{A}_{w_t}$

Algorithm:
for $t := 0, 1, 2, \cdots$ **do**
Draw a control action a_t from the distribution $\pi_{\theta_t}(\cdot|s_t)$.
Perform a_t, and observe the reward r_t and the next state s_{t+1}.
Use the recursive least-squares rules (5) and (6) to find w_t and v_t.
Adjust the policy distribution parameters via $\theta_{t+1} = \theta_t + \alpha w_t$.
if any entry of θ_{t+1} is of magnitude bigger than M,
then reduce it to M.
end
end

3 Locomotion of a Two-Linked Robot Arm

In this section, we address the application of the RLS-based natural actor-critic algorithm to an example of [8], which dealt with locomotion of a robot arm. This example considers a planar two-linked manipulator in a gravitational environment. The mission assigned to the robot is to move forward as fast as possible, without knowing the environment in advance. So, the agent needs to find out

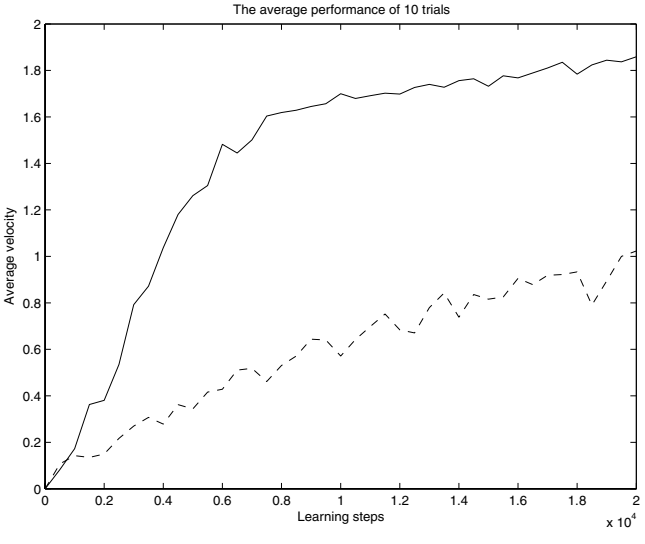

Fig. 1. The average performance of 10 trials. RLS-based natural actor-critic approach (solid), and the SGA approach of [8] (dashed).

an efficient policy based on the observed agent-environment interactions. The immediate reward for this problem is defined as the distance that the body of the robot moved forward in the current step. At the t-th time step, the agent reads the normalized joint angles, $x_1(t)$ and $x_2(t)$, and outputs a binary action value ($+1$ or -1) for each joint, which indicates turning direction of the joint motor, according to its stochastic policy. Here, the policy is represented by the logistic distribution function, i.e., $\pi_{\theta_t}(a_t = 1|s_t) = 1/(1 + \exp(-\theta_t^T s_t))$, where s_t is the state vector at the t-th time step defined by $s_t \stackrel{\triangle}{=} [x_1(t), x_2(t), 1]^T$. The considered robot has the same specifications with [8], thus it satisfies the following:

- The upper arm length is 34 [cm].
- The fore arm length is 20 [cm].
- The joint of the body and the arm is located on the height = 18 [cm], the horizontal distance = 32 [cm] from the body's bottom left corner.
- The angle from the horizontal of the first joint connected to the upper arm is constrained such that $-4^\circ \leq \angle(Joint1) \leq 35^\circ$.
- The angle of the second joint from the axis of the upper arm to the fore arm is constrained such that $-120^\circ \leq \angle(Joint2) \leq 10^\circ$.
- The motor of each joint moves the arm to $12^\circ + \epsilon$ in the commanded direction, where ϵ is a random variable uniformly distributed on the interval $[-4^\circ, +4^\circ]$.
- When the arm is touching the ground, the arm does not slip while the body slips.

The proposed method was applied to the robot for $20,000$ time steps with the following parameters:

- Initial state $s_0 = [0\ 0\ 1]^T$
- Initial policy distribution vector $\theta_0 = [0\ 0\ 0\ 0\ 0\ 0]^T$
- Basis functions $\phi(s) \stackrel{\triangle}{=} [x_1\ x_2\ 1]^T$
- Learning rate $\alpha = 0.0003$
- Forgetting factor $\beta = 0.99$
- Discount rate $\gamma = 0.99$
- Trace-decay parameter $\lambda = 0.5$
- Constant $\delta = 10$
- Bound for each actor parameter $M = 100$

The solid curve of Fig. 1 shows the average velocity of 10 trials resulting from the RLS-based natural actor-critic algorithm applied to the robot example. For comparison, we also performed simulations for the SGA(stochastic gradient ascent) method of [8]. Shown in the dashed curve of Fig. 1 is the average performance of 10 trials for the SGA method. Comparing these curves in the figure, we see that the RLS-based natural actor-critic algorithm gave significantly better results.

4 Concluding Remarks

In this paper, we studied on the RLS-based natural actor-critic algorithm, and applied it to locomotion of a two-linked robot arm. The natural policy gradient

and the recursive least-squares method are two key ingredients of the considered algorithm, and they are used in the process of updating the actor and critic parameters, respectively. Simulation results for the example dealing with locomotion of a two-linked robot arm showed that the algorithm considered in this paper yields better performance compared to the conventional SGA method. Further investigations yet to be done include extensive comparative studies which can reveal the strength and weakness of the considered method.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
2. Peters, J., Vijayakumar, S., Schaal, S.: Reinforcement learning for humanoid robotics. In: Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots (2003)
3. Xu, X., He, H., Hu, D.: Efficient reinforcement learning using recursive least-squaes methods. Journal of Artificial Intelligent Research **16** (2002) 259–292
4. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. Advances in Neural Information Processing Systems **12** (2000) 1057–1063
5. Konda, V., Tsitsiklis, J.N.: Actor-Critic Algorithms. SIAM Journal on Control and Optimization **42(4)** (2003) 1143–1166
6. Amari, S.: Natural gradient works efficiently in learning. Neural Computation **10(2)** (1998) 251–276
7. Moon, T.K., Stirling, W.C.: Mathematical Methods and Algorithm for Signal Processing. Prentice Hall, Upper Saddle River, NJ (2000)
8. Kimura, H., Miyazaki, K., Kobayashi, S.: Reinforcement learning in POMDPs with function approximation. In: Fisher, D.H. (ed.): Proceedings of the Fourteenth International Conference on Machine Learning (1997) 152–160