NOTE Communicated by Rajesh P. Rao

Kalman Filter Control Embedded into the Reinforcement Learning Framework

István Szita
szityu@eotvoscollegium.hu
András Lőrincz
alorincz@axelero.hu
Department of Information Systems, Eötvös Loránd University,
Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary

There is a growing interest in using Kalman filter models in brain modeling. The question arises whether Kalman filter models can be used on-line not only for estimation but for control. The usual method of optimal control of Kalman filter makes use of off-line backward recursion, which is not satisfactory for this purpose. Here, it is shown that a slight modification of the linear-quadratic-gaussian Kalman filter model allows the on-line estimation of optimal control by using reinforcement learning and overcomes this difficulty. Moreover, the emerging learning rule for value estimation exhibits a Hebbian form, which is weighted by the error of the value estimation.

1 Motivation

Kalman filters and their various extensions are well studied and widely applied tools in both state estimation and control. Recently, there has been increasing interest in Kalman filters (KF) or Kalman filter–like structures as models for neurobiological substrates. It has been suggested that Kalman filtering may occur at sensory processing (Rao & Ballard, 1997, 1999) or at acquisition of behavioral responses (Kakade & Dayan, 2000), may be the underlying computation of the hippocampus (Bousquet, Balakrishnan, & Honavar, 1998), and may be the underlying principle in control architectures (Todorov & Jordan, 2002a, 2002b). Detailed architectural similarities between Kalman filter and the entorhinal-hippocampal loop, as well as between Kalman filters and the neocortical hierarchy, have been described recently (Lőrincz & Buzsáki, 2000; Lőrincz, Szatmáry, & Szirtes, 2002). The interplay between the dynamics of Kalman filter–like architectures and learning of parameters of neuronal networks is promising for explaining known and puzzling phenomena such as priming, repetition suppression, and categorization (Lőrincz, Szirtes, Takács, Biederman, & Vogels, 2002; Kéri et al., 2002).

It is well known that Kalman filters provide an on-line estimation of the state of the system. On the other hand, optimal control typically cannot be computed on-line, because it is given by a backward recursion, the Ricatti-equations. (For on-line parameter estimations of Kalman filters but without control aspects, see Rao, 1999.)

The aim of this article is to demonstrate that Kalman filters can be embedded into a goal-oriented framework. We show that slight modification of the linear-quadratic-gaussian (LQG) Kalman filter model is satisfactory to integrate the LQG model into the reinforcement learning (RL) framework.

2 The Kalman Filter and the LQG Model

Consider a linear dynamical system with state $\mathbf{x_t} \in \mathbb{R}^{\mathbf{n}}$, control $\mathbf{u_t} \in \mathbb{R}^{\mathbf{m}}$, observation $\mathbf{y_t} \in \mathbb{R}^{\mathbf{k}}$, and noises $\mathbf{w_t} \in \mathbb{R}^{\mathbf{n}}$ and $\mathbf{e_t} \in \mathbb{R}^{\mathbf{k}}$ (which are assumed to be uncorrelated Gaussians with covariance matrix Ω^w and Ω^e, respectively) in discrete time t:

$$\mathbf{x_{t+1}} = F\mathbf{x_t} + G\mathbf{u_t} + \mathbf{w_t} \tag{2.1}$$

$$\mathbf{y_t} = H\mathbf{x_t} + \mathbf{e_t}. \tag{2.2}$$

Assume that the initial state has mean $\hat{\mathbf{x}}_1$, covariance Σ_1 and that executing control step $\mathbf{u_t}$ in $\mathbf{x_t}$ costs

$$c(\mathbf{x}_t, \mathbf{u}_t) := \mathbf{x}_t^T Q \mathbf{x}_t + \mathbf{u}_t^T R \mathbf{u}_t. \tag{2.3}$$

Further, assume that after the Nth step, the controller halts and receives a final cost of $\mathbf{x}_N^T Q_N \mathbf{x}_N$. The task is to find a control sequence with minimum total cost.[1] This problem has the well-known solution

$$\hat{\mathbf{x}}_{t+1} = F\hat{\mathbf{x}}_t + G\mathbf{u}_t + K_t(\mathbf{y}_t - H\hat{\mathbf{x}}_t) \tag{2.4}$$

$$K_t = F\Sigma_t H^T (H\Sigma_t H^T + \Omega^e)^{-1} \tag{2.5}$$

$$\Sigma_{t+1} = \Omega^w + F\Sigma_t F^T - K_t H \Sigma_t F^T \qquad \text{(state estimation)} \tag{2.6}$$

and

$$\mathbf{u}_t = -L_t \hat{\mathbf{x}}_t \tag{2.7}$$

$$L_t = (G^T S_{t+1} G + R)^{-1} G^T S_{t+1} F \tag{2.8}$$

$$S_t = Q_t + F^T S_{t+1} F - F^T S_{t+1} G L_t. \qquad \text{(optimal control)} \tag{2.9}$$

Unfortunately, the optimal control equations are not on-line, because they can be solved only by stepping backward from the final (i.e., the Nth) step.

[1] In a more general setting, c is a general quadratic function of $\mathbf{x}_t$, $\mathbf{u}_t$, and an optional $\mathbf{x}_t^{\text{track}}$, where $\mathbf{x}_t^{\text{track}}$ is a trajectory of a linear system to be tracked.

3 Integrating Kalman Filtering into the Reinforcement Learning Framework

First, we slightly modify the problem: the run time of the controller will not be a fixed number N. Instead, after each time step, the process will be stopped with some fixed probability p (and then the controller incurs the final cost $c_f(\mathbf{x}_f) := \mathbf{x}_f^t Q_f \mathbf{x}_f$). This modification is commonly used in the RL literature; it makes the problem more amenable to mathematical treatment.

3.1 The Cost-to-Go Function. Let $V_t^*(\mathbf{x})$ be the optimal cost-to-go function at time step t,

$$V_t^*(\mathbf{x}) := \inf_{\mathbf{u}_t, \mathbf{u}_{t+1}, \ldots} E[c(\mathbf{x}_t, \mathbf{u}_t) + c(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) + \cdots + c_f(\mathbf{x}_f) \mid \mathbf{x}_t = \mathbf{x}]. \quad (3.1)$$

Considering that the controller is stopped with probability p, equation 3.1 assumes the following form,

$$V_t^*(\mathbf{x}) = p \cdot c_f(\mathbf{x}) + (1-p) \inf_{\mathbf{u}} (c(\mathbf{x}, \mathbf{u}) + E_{\mathbf{w}}[V_{t+1}^*(F\mathbf{x} + G\mathbf{u} + \mathbf{w})]) \quad (3.2)$$

for any state $\mathbf{x}$. It is easy to show that the optimal cost-to-go function is time independent and it is a quadratic function of $\mathbf{x}$. That is, the optimal cost-to-go action-value function assumes the form

$$V^*(\mathbf{x}) = \mathbf{x}^T \Pi^* \mathbf{x}. \quad (3.3)$$

Our task is to estimate the optimal value functions (parameter matrix Π^*) on-line. This can be done by the method of temporal differences.

We start with an arbitrary initial cost-to-go function $V_0(\mathbf{x}) = \mathbf{x}^T \Pi_0 \mathbf{x}$. After this, (1) control actions are selected according to the current value function estimate, (2) the value function is updated according to the experience, and (3) these two steps are iterated.

The tth estimate of V^* is $V_t(\mathbf{x}) = \mathbf{x}^T \Pi_t \mathbf{x}$. The greedy control action according to this is given by

$$\begin{aligned} \mathbf{u}_t &= \arg\min_{\mathbf{u}} (E[c(\mathbf{x}_t, \mathbf{u}) + V_t(F\mathbf{x}_t + G\mathbf{u} + w)]) \\ &= \arg\min_{\mathbf{u}} (\mathbf{u}^T R \mathbf{u} + (F\hat{\mathbf{x}}_t + G\mathbf{u})^T \Pi_t (F\hat{\mathbf{x}}_t + G\mathbf{u})) \\ &= -(R + G^T \Pi_t G)^{-1} (G^T \Pi_t F) \hat{\mathbf{x}}_t. \end{aligned} \quad (3.4)$$

For simplicity, the cost-to-go function will be updated by using the one-step temporal differencing (TD) method, although it could be substituted with more sophisticated methods like multistep TD or eligibility traces. The TD error is

$$\delta_t = \begin{cases} c_f(\hat{\mathbf{x}}) - V_t(\hat{\mathbf{x}}_{t-1}) & \text{if } t = t_{STOP}, \\ (c(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) + V_t(\hat{\mathbf{x}}_t)) - V_t(\hat{\mathbf{x}}_{t-1}), & \text{otherwise}, \end{cases} \quad (3.5)$$

and the update rule for the parameter matrix Π_t is

$$\begin{aligned}\Pi_{t+1} &= \Pi_t + \alpha_t \cdot \delta_t \cdot \nabla_{\Pi_t} V_t(\hat{\mathbf{x}}_t) \\ &= \Pi_t + \alpha_t \cdot \delta_t \cdot \hat{\mathbf{x}}_t \hat{\mathbf{x}}_t^T,\end{aligned} \tag{3.6}$$

where α_t is the learning rate. Note that value-estimation error weighted Hebbian learning rule has emerged.

3.2 Sarsa. The cost-to-go function is used to select control actions, so the estimation of the action-value function $Q_t^*(\mathbf{x}, \mathbf{u})$ is more appropriate here. The action-value function is defined as

$$\begin{aligned}Q_t^*(\mathbf{x}, \mathbf{u}) := &\inf_{\mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \ldots} \\ &E[c(\mathbf{x}_t, \mathbf{u}_t) + c(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) + \cdots + c_f(\mathbf{x}_f) \mid \mathbf{x}_t = \mathbf{x}, \mathbf{u}_t = \mathbf{u}],\end{aligned} \tag{3.7}$$

and analogously to V_t^*, it can be shown that it is time independent and assumes the form

$$Q^*(\mathbf{x}, \mathbf{u}) = \begin{pmatrix}\mathbf{x}^T & \mathbf{u}^T\end{pmatrix} \begin{pmatrix}\Theta_{11}^* & \Theta_{12}^* \\ \Theta_{21}^* & \Theta_{22}^*\end{pmatrix} \begin{pmatrix}\mathbf{x} \\ \mathbf{u}\end{pmatrix} = \begin{pmatrix}\mathbf{x}^T & \mathbf{u}^T\end{pmatrix} \Theta^* \begin{pmatrix}\mathbf{x} \\ \mathbf{u}\end{pmatrix}. \tag{3.8}$$

If the tth estimate of Q^* is $Q_t(\mathbf{x}, \mathbf{u}) = [\mathbf{x}^T, \mathbf{u}^T]\Theta_t[\mathbf{x}^T, \mathbf{u}^T]^T$, then the greedy control action is given by

$$\mathbf{u}_t = \arg\min_{\mathbf{u}} E(Q_t(\mathbf{x}_t, \mathbf{u})) = -\Theta_{22}^{-1}\Theta_{21}\hat{\mathbf{x}}_t, \tag{3.9}$$

where subscript t of Θ has been omitted to improve readability. Note that the value function estimate (but not model) is needed to compute $\mathbf{u}_t$.

The estimation error and the weight update are quite similar to the state-value case:

$$\delta_t = \begin{cases} c_f(\hat{\mathbf{x}}_t) - Q_t(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) & \text{if } t = t_{STOP}, \\ (c(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) + Q_t(\hat{\mathbf{x}}_t, \mathbf{u}_t)) - Q_t(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}), & \text{otherwise,} \end{cases} \tag{3.10}$$

$$\begin{aligned}\Theta_{t+1} &= \Theta_t + \alpha_t \cdot \delta_t \cdot \nabla_{\Theta_t} Q_t(\hat{\mathbf{x}}_t, \mathbf{u}_t) \\ &= \Theta_t + \alpha_t \cdot \delta_t \cdot \begin{pmatrix}\hat{\mathbf{x}}_t \\ \mathbf{u}_t\end{pmatrix} \begin{pmatrix}\hat{\mathbf{x}}_t \\ \mathbf{u}_t\end{pmatrix}^T.\end{aligned} \tag{3.11}$$

3.3 Convergence. There are numerous results showing that simple RL algorithms with linear function approximation can diverge (see, e.g., Baird, 1995). There are also positive results dealing with the constant policy case (i.e., with policy evaluation) and iterative policy improvements (Tsitsiklis & Van Roy, 1996).

For our case, convergence proof can be provided (the complete proof can be found in Szita & Lőrincz, 2003).

1. One can show that an appropriate form of the separation principle holds for value estimation. That is, using the state estimates $\hat{\mathbf{x}}_t$ for computing the control is equivalent to using the exact states.
2. One can apply Gordon's (2001) results, which states that under appropriate conditions, TD and Sarsa control with linear function approximation cannot diverge.
3. In our problem, where the system is linear and the costs are quadratic, using Gordon's (2001) technique, one can prove the stronger result:

   Theorem. *If the system* (F, H) *is observable,* $\Pi_0 \geq \Pi^*$ *(or* $\Theta_0 \geq \Theta^*$ *in the case of Sarsa), there exists an* L *such that* $\|F + GL\| \leq 1/\sqrt{1-p}$*, there exists an* M *such that* $\|\mathbf{x}_t\| \leq M$ *for all* t*, and the constants* α_t *satisfy* $0 < \alpha_t < 1/M^4$, $\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$*. Then combined TD-KF methods converge to the optimal policy with probability 1 for both state-value and for action-value estimations.*

4 Discussion

4.1 Possible Extensions. We have demonstrated that Kalman filtering can be integrated into the framework of reinforcement learning. There are numerous possibilities for extending this scheme on both sides; both KF and RL can be extended. We list some of these possibilities:

- *Advanced RL algorithms.* The one-step TD method can be replaced by more efficient algorithms like TD(λ) (eligibility traces), Sarsa(λ), and Q-learning.
- *Parameter estimation.* In its present form, the algorithm needs a model of the system (i.e., the parameters F, G, and H). However, for Q-learning, reinforcement learning may not require any model. In turn, the algorithm can be augmented by standard KF parameter estimation techniques like expectation maximization (see, e.g., Rao, 1999) and by information maximization methods (Lőrincz, Szatmáry, & Szirtes, 2002). As a result, an on-line, model-free control algorithm for the linear quadratic regulation problem can be obtained.
- *Extended Kalman filtering.* Kalman filtering makes the assumption that both the system and the observations are linear. This restriction can be overcome by using extended Kalman filters (EKF), adding further potential to our approach.
- *Kalman smoother.* To obtain more accurate and less noisy state estimations, we could use Kalman smoothing instead of the filtering equation. One-step smoothing does not impose much additional computational difficulty, because one-step lookahead is needed anyway for the temporal differencing method.

- *More general quadratic and nonquadratic cost functions.* One is not restricted to the simplest quadratic loss functions. The KF equations are independent of the costs, and RL can handle arbitrary costs. Moreover, in many cases, costs can be rewritten into the form of a linear function approximation (see, e.g., Bradtke, 1993), and convergence is then warranted.

4.2 Related Works in Reinforcement Learning. Reinforcement learning has been applied in the linear quadratic regulation problem (Landelius & Knutsson, 1996; Bradtke, 1993; ten Hagen & Kröse, 1998). The main difference is that these works assume fully observed systems and that either Q-learning or the recursive least-squares methods were used as the RL component.

It is important that our approach can be interpreted as a partially observed Markov decision process (POMDP), with continuous state and action spaces. In general, learning in POMDPs is a very difficult task (see Murphy, 2000, for a review). We consider the LQG approach attractive, because it is a POMDP and yet can be handled efficiently because of its particular properties: both the transitions and the observations are linear, and uncertainties assume gaussian form.

4.3 Neurobiological Connections. The modeling of neural conditioning by TD learning is not new: a series of papers have been published on this topic (e.g., Montague, Dayan, & Sejnowski, 1996; Schultz, Dayan, & Montague, 1997). In these works, it is typically assumed that states can be fully observed. Recently, Daw, Courville, and Touretzky (in press) proposed partially observable semi-Markov processes as an underlying model to deal with nonobservability.

The motivation for our work comes from neurobiology. Some novel theoretical works in neurobiology (Rao & Ballard, 1997; Lőrincz & Buzsáki, 2000; Lőrincz, Szatmáry, & Szirtes, 2002; Todorov & Jordan, 2002a) claim that both sensory processing and control may be based on Kalman filter–like structures. Notably, some odd predictions (Lőrincz & Buzsáki, 2000; Lőrincz, Szatmáry, & Szirtes, 2002) have been reinforced recently:

- Properties of neurons corresponding to the internal representation of the Kalman filter (the Vth and the VIth layers of the entorhinal cortex, EC) versus properties of neurons corresponding the reconstruction error and the filtered input (EC layers II and III), respectively
- Adaptable long-delay properties of neurons with the putative role of eliminating temporal convolutions arising in Kalman filter–like structures with not-yet-tuned parameters

Both have been confirmed by experiments reported in Egorov, Hamam, Fransén, Hasselmo, and Alonso (2002) and in Henze, Wittner, and Buzsáki (2002), respectively.

Clearly, the brain is a goal-oriented system, and Kalman filters need to be embedded into a goal-oriented framework. Moreover, this framework should exhibit Hebbian learning properties. Further, although batch learning does have some plausibility in neurobiological models—consider, for example, hippocampal replays of time sequences (Skaggs & McNaughton, 1996; Nadasdy, Hirase, Czurko, Csicsvari, & Buzsáki, 1999)—the extent of such batch learning should be limited given that the environment is subject to changes. In turn, our work reinforces the efforts dealing with Kalman filter description of neocortical processing.

From the point of view of parameter estimation, the Kalman filter seems to be an ideal neurobiological candidate (Lőrincz & Buzsáki, 2000; Lőrincz, Szatmáry, & Szirtes, 2002). On-line estimation of the Kalman gain, however, remains a problem. (But see Póczos & Lőrincz, 2003.)

It has been noted that smoothing should improve the efficiency of the algorithm. Albeit the importance of smoothing and switching between smooth solutions is striking in the neurobiological context, it remains an open and intriguing issue if and how the neurobiological framework may allow the integration of smoothing.

5 Conclusions

Recent research in theoretical neurobiology indicates that Kalman filters have intriguing potentials in brain modeling. However, the classical method for Kalman filter control requires off-line computations, which are unlikely to take place in the brain. Our aim here was to embed Kalman filters into the reinforcement learning framework. This was achieved by applying the method of temporal differences. Theoretical achievements of reinforcement learning were used to describe the asymptotic optimality of the algorithm.

Although our algorithm is only asymptotically optimal and may be slower than the classical approaches like solving the Ricatti recursions (used commonly in engineering applications), it has several advantages: it works on-line, no model is needed for computing the control law, and the learning is Hebbian, which is weighted by the error of value estimation. These properties make it an attractive approach for brain modeling and neurobiological applications.

Furthermore, the algorithm described here admits several straightforward generalizations, including the extended Kalman filters, parameter estimations, nonquadratic cost functions, and eligibility traces, which can extend its applicability and improve its performance.

Acknowledgments

We are grateful to the anonymous referees who called our attention to recent work on the topic. Their suggestions helped us to improve this note. This work was supported by the Hungarian National Science Foundation (Grant No. T-32487).

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning* (pp. 30–37). San Francisco: Morgan Kaufmann.

Bousquet, O., Balakrishnan, K., & Honavar, V. (1998). Is the hippocampus a Kalman filter? In *Proceedings of the Pacific Symposium on Biocomputing* (pp. 655–666). Singapore: World Scientific.

Bradtke, S. J. (1993). Reinforcement learning applied to linear quadratic regulation. In C. L. Giles, S. J. Hanson, & J. D. Cowan (Eds.), *Advances in neural information processing systems, 5*, (pp. 295–302). San Mateo, CA: Morgan Kaufmann.

Daw, N., Courville, A., & Touretzky, D. S. (in press). Timing and partial observability in the dopamine system. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15*. Cambridge, MA: MIT Press.

Egorov, A., Hamam, B., Fransén, E., Hasselmo, M., & Alonso, A. (2002). Graded persistent activity in entorhinal cortex neurons. *Nature, 420*, 173–178.

Gordon, G. J. (2001). Reinforcement learning with function approximation converges to a region. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13* (pp. 1040–1046). Cambridge, MA: MIT Press.

Henze, D., Wittner, L., & Buzsáki, G. (2002). Single granule cells reliably discharge targets in the hippocampal CA3 network in vivo. *Nature Neuroscience, 5*, 790–795.

Kakade, S., & Dayan, P. (2000). Acquisition in autoshaping. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing, 12* (pp. 24–30). Cambridge, MA: MIT Press.

Kéri, S., Benedek, G., Janka, Z., Aszalós, P., Szatmáry, B., Szirtes, G., & Lőrincz, A. (2002). Categories, prototypes and memory systems in Alzheimer's disease. *Trends in Cognitive Science, 6*, 132–136.

Landelius, T., & Knutsson, H. (1996). Greedy adaptive critics for LQR problems: Convergence proofs (Tech. Rep. No. LiTH-ISY-R-1896). Linköping, Sweden: Computer Vision Laboratory.

Lőrincz, A., & Buzsáki, G. (2000). The parahippocampal region: Implications for neurological and psychiatric diseases. In H. Scharfman, M. Witter, & R. Schwarz (Eds.), *Annals of the New York Academy of Sciences* (Vol. 911, pp. 83–111). New York: New York Academy of Sciences.

Lőrincz, A., Szatmáry, B., & Szirtes, G. (2002). Mystery of structure and function of sensory processing areas of the neocortex: A resolution. *J. Comp. Neurosci., 13*, 187–205.

Lőrincz, A., Szirtes, G., Takács, B., Biederman, I., & Vogels, R. (2002). Relating priming and repetition suppression. *Int. J. Neural Systems, 12*, 187–202.

Montague, P., Dayan, P., & Sejnowski, T. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience, 16*(5), 1936–1947.

Murphy, K. P. (2000). *A survey of POMDP solution techniques.* Available on-line at: http://www.ai.mit.edu/m̃urphyk/Papers/pomdp.ps.gz.

Nadasdy, Z., Hirase, H., Czurko, A., Csicsvari, J., & Buzsáki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *Journal of Neuroscience, 19*, 9497–9507.

Póczos, B., & Lőrincz, A. (2003). *Kalman-filtering using local interactions* (Tech. Rep. No. NIPG-ELU-28-02-2003). Budapest: Faculty of Informatics, Eötvös Loránd University. Available on-line at: http://arxiv.org/abs/cs.AI/0302039.

Rao, R. (1999). An optimal estimation approach to visual perception and learning. *Vision Research, 39*, 1963–1989.

Rao, R., & Ballard, D. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation, 9*, 721–763.

Rao, R., & Ballard, D. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience, 2*, 79–87.

Schultz, W., Dayan, P., & Montague, P. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593–1599.

Skaggs, W., & McNaughton, B. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science, 271*, 1870–1873.

Szita, I., & Lőrincz, A. (2003). *Reinforcement learning with linear function approximation and LQ control coverves.* (Tech. Rep. No. NIPG-ELU-22-06-2003). Budapest: Faculty of Informatics, Eötvös Loránd University. Available on-line at: http://arvis.org/abs/cs.LG/0306120.

ten Hagen, S., & Kröse, B. (1998). Linear quadratic regulation using reinforcement learning. In F. Verdenius & W. van den Broek (Eds.), *Proceedings of the 8th Belgian-Dutch Conf. on Machine Learning* (pp. 39–46). Wageningen: BENELEARN-98.

Todorov, E., & Jordan, M. (2002a). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience, 5*, 1226–1235.

Todorov, E., & Jordan, M. (2002b). *Supplementary notes for optimal feedback control as a theory of motor coordination.* Available on-line at: http://www.nature.com/neuro/supplements/.

Tsitsiklis, J. N., & Van Roy, B. (1996). *An analysis of temporal-difference learning with function approximation* (Tech. Rep. No. LIDS-P-2322). Cambridge, MA: MIT.

Received December 16, 2002; accepted July 15, 2003.