A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal–Difference Learning[1]

David Choi[2] and Benjamin Van Roy[3]

[1] This research was supported in part by NSF CAREER Grant ECS–9985229, and by the ONR under Grant MURI N00014-00-1-0637.

[2] Department of Electrical Engineering, Stanford University.

[3] Departments of Management Science and Engineering and Electrical Engineering, Stanford University.

Abstract

The traditional Kalman filter can be viewed as a recursive stochastic algorithm that approximates an unknown function via a linear combination of prespecified basis functions given a sequence of noisy samples. In this paper, we generalize the algorithm to one that approximates the fixed point of an operator that is known to be a Euclidean norm contraction. Instead of noisy samples of the desired fixed point, the algorithm updates parameters based on noisy samples of functions generated by application of the operator, in the spirit of Robbins–Monro stochastic approximation. The algorithm is motivated by temporal–difference learning, and our developments lead to a possibly more efficient variant of temporal–difference learning. We establish convergence of the algorithm and explore efficiency gains through computational experiments involving optimal stopping and queueing problems.

Key Words: Dynamic programming, Kalman filter, optimal stopping, queuing, recursive least–squares, reinforcement learning, temporal–difference learning.

1 Introduction

We consider the problem of fixed point computation:

> Given an operator F acting on functions $J : S \mapsto \Re$, find a function J^* such that $J^* = FJ^*$.

Our interest is in cases where S is very large, as we are motivated by problems in which elements of S are identified with vectors of d variable components, and the cardinality of S therefore grows exponentially in d. In such situations, a fixed point is often too large to store, let alone compute. Known as "the curse of dimensionality," this phenomenon gives rise to prohibitive computational requirements for many problems of practical interest.

1.1 Contractions and Successive Approximations

In this paper, we develop approximation methods for fixed point problems where the cardinality n of S is finite, though possibly enormous, so that mappings of the form $S \mapsto \Re$ can be represented by vectors in $\Re^{|S|}$. An approximation is sought for the fixed point of an operator $F : \Re^{|S|} \mapsto \Re^{|S|}$, assumed to be a contraction with respect to a weighted Euclidean–norm $\| \cdot \|_D$, defined by

$$\|J\|_D = (J'DJ)^{1/2},$$

where D is a positive definite diagonal matrix. Without loss of generality, we let the diagonal elements of D sum to 1, so that they can be viewed as probabilities over S.

By virtue of being a contraction, the operator F is guaranteed to possess a unique fixed point J^*, and given sufficient compute time, one can generate successive approximations according to

$$J_{t+1} = FJ_t,$$

that converge to J^*. In particular, for any $J_0 \in \Re^n$, we have $J_t \rightarrow J^*$. Unfortunately, the curse of dimensionality often renders the successive approximations method intractable.

1.2 Fitting Basis Functions

One simple approach to alleviating computational requirements involves approximating each iterate J_t by a linear combination of prespecified basis functions $\phi_1, \ldots, \phi_K \in \Re^{|S|}$, in a spirit reminiscent of statistical regression. In particular, for each t, a weight vector $r_t \in \Re^K$ is generated with the intent of offering an approximation

$$\sum_{k=1}^{K} r_t(k)\phi_k \approx J_t.$$

Or defining a matrix

$$\Phi = \begin{bmatrix} | & & | \\ \phi_1 & \cdots & \phi_K \\ | & & | \end{bmatrix},$$

the approximation can be written as Φr_t. One method for generating the sequence of weight vectors iteratively solves for r_{t+1} satisfying the relation

$$\Phi r_{t+1} = \Pi F \Phi r_t,$$

where Π is a projection operator that projects onto the span of $\phi_1, \ldots, \phi_K$ with respect to $\| \cdot \|_D$. (In our discussion, $|S|$ is finite, so Π is easily verified to be $\Pi = \Phi(\Phi' D \Phi)^{-1} \Phi' D$) As established by [23], this sequence of approximations converges to a fixed point Φr^* equaling $\Pi F \Phi r^*$, which satisfies an error bound

$$\|J^* - \Phi r^*\|_D \leq \frac{1}{\sqrt{1-\alpha^2}} \|J^* - \Pi J^*\|_D,$$

where α is the contraction factor of F. In other words, the resulting approximation error is within a constant factor of the best possible, which is the error associated with the projection ΠJ^*.

Our developments in this paper are motivated by ongoing work on approximation methods for large–scale dynamic programming. To put our formulation and algorithms in perspective, let us present a relevant example, involving optimal stopping problems as treated in [24, 26].

Example 1 Approximate Value Iteration for Optimal Stopping Problems *Consider a discrete–time Markov process* $x_0, x_1, x_2, \ldots$ *evolving on a finite state space* S*, with dynamics characterized by an* $n \times n$ *irreducible aperiodic transition matrix* P*. Let the* σ*–field generated by* $x_0, \ldots, x_t$ *be denoted by* $\mathcal{F}_t$*. A stopping time is a random variable* τ *taking on values in* $0, 1, 2, \ldots$ *such that for any* t*, the event* $\{\tau \leq t\}$ *is* $\mathcal{F}_t$*–measurable. We consider an optimal stopping problem of the form*

$$\max_{\tau} E[\alpha^{\tau} G(x_{\tau})],$$

where G *is a reward received upon stopping,* α *is a discount factor in* $(0,1)$*, and the maximization is over stopping times.*

This problem can be solved by dynamic programming. In particular, letting an operator $T : \Re^n \mapsto \Re^n$ *be defined by*

$$TJ = \max(G, \alpha P J),$$

with the maximization taken pointwise, the problem amounts to computing the fixed point J^* *of* T*. It is well known that* T *is a* $\| \cdot \|_D$*-norm contraction with respect to the maximum norm, and therefore, it possesses a unique fixed point. Given* J^**, an optimal stopping time* τ^* *can be generated according to*

$$\tau^* = \min\{t | G(x_t) \geq J^*(x_t)\}.$$

It has been shown in [24] that T *is contraction with respect to* $\| \cdot \|_D$*, where the diagonal entries of* D *correspond to stationary probabilities* $\pi(x)$ *of the Markov process. This motivates use of the approximation method discussed in the previous section. In particular, when the state space underlying an optimal stopping problem is too large and* *exact solution is infeasible, one can select basis functions $\phi_1, \ldots, \phi_K$ and then hope to generate an approximation within their span using the iteration*

$$\Phi r_{m+1} = \Pi T \Phi r_m,$$

where the projection matrix Π (given by $\Pi = \Phi(\Phi' D \Phi)^{-1}\Phi' D$) projects onto the span of $\phi_1, \ldots, \phi_K$ with respect to $\|\cdot\|_D$.

As in [24], given the limit Φr^* of this sequence, one might generate an approximately optimal stopping time $\tilde{\tau}$ according to

$$\tilde{\tau} = \min\{t | G(x_t) \geq (\Phi r^*)(x_t)\}.$$

1.3 Stochastic Samples

The advantage of the iteration $\Phi r_{m+1} = \Pi F \Phi r_m$ over the standard successive approximations iteration $J_{m+1} = F J_m$ is that it updates a weight vector r_m of dimension K rather than a vector J_m of dimension $n \gg K$. However, the iteration is still impractical in many contexts for the following reasons:

1. The iteration $\Phi r_{m+1} = \Pi F \Phi r_m$ entails computation of $(F\Phi r_m)(x)$ for all $x \in S$, and this its projection to the span of Φ. Due to the curse of dimensionality, $|S|$ can be enormous, making it impossible to compute or store this many values.
2. It is often difficult to compute $(F\Phi r_m)(x)$ exactly, even for a particular x. This is because the value of $(F\Phi r_m)(x)$ can depend on every component of Φr_m, and the number $|S|$ of components can be enormous,

It turns out that both of these obstacles can often be tackled using stochastic algorithms. In particular, it is often the case that, given some $x \in S$ and $r \in \Re^k$, one can efficiently generate a noisy stochastic sample $\eta(x)$ such that $E[\eta(x)] = (F\Phi r)(x)$, and that there are efficient algorithms that converge on Φr^* using a reasonable number of such samples. We will discuss such algorithms later. For now, let us motivate the obstacles and how the samples of interest might be generated in the optimal stopping context introduced in the previous section.

Example 2 Stochastic Sampling for Optimal Stopping Problems

To approximate the solution of an optimal stopping problem, we proposed an iteration $\Phi r_{m+1} = \Pi T \Phi r_m$. It is clear that, if the underlying state space is intractable, we should not aim at computing $T\Phi r_m$. However, we might hope to compute a sample value

$$(T\Phi r_m)(x) = \max\left(G(x), \alpha \sum_{y \in S} P_{xy}(\Phi r_m)(y)\right),$$

for a representative state x. If there are only a few possible states that can follow x, there will be only a few nonzero P_{xy}'s, and the summation can be computed efficiently. In this case, we can obtain $(T\Phi r_m)(x)$. However, when there are many possible states that could follow x, the summation becomes intractable. In the spirit of Monte–Carlo methods, however, we might try to generate an unbiased estimate.

In particular, instead of summing over all possible states that could follow x, sample a single state y according to transition probabilities P_{xy}. Since y is a stochastic sample *for the next state, we might hope to use y to produce an unbiased estimate of $(T\Phi r_m)(x)$. One might imagine using an estimate of the form $\max(G(x), \alpha(\Phi r_m)(y))$; however, this estimate is not unbiased, since $\mathrm{E}_y[\max(G(x), \alpha(\Phi r_m)(y))|x]$ is generally greater than $(T\Phi r_m)(x)$.*

Instead of trying to find an unbiased estimate for $(T\Phi r_m)(x)$, we can circumvent the problem by estimating an entirely different (though related) value $(H\Phi r_m)(x)$, where the operator H is defined as

$$HQ = \alpha P \max(G, Q).$$

It is well known (see Section 6.8.4, p.358 of [7]) that H is a contraction with fixed point Q^* defined by

$$Q^* = \alpha P J^*,$$

and that Q^* also satisfies

$$J^*(x) = \max(G(x), Q^*(x)).$$

As a result, a sample $\eta(x) = \alpha \max(G(y), (\Phi r_m)(y))$ can be generated efficiently and serves as an unbiased estimate of $(H\Phi r_m)(x)$. However, the fixed point of H – and therefore the function we are going to end up approximating – is not J^*, but rather Q^*. We will discuss later algorithms that produce weights r^* such that Φr^* approximates Q^* and $\max(G, \Phi r^*)$ approximates J^*.

1.4 Stochastic Steepest Descent and Temporal–Difference Learning

Our work is motivated largely by the temporal-difference learning algorithm (TD) [20]. As presented in [23, 26], TD operates along the lines of our discussion, using stochastic samples $\eta(x)$ to estimate $(F\Phi r)(x)$. The stochastic estimation error can be represented as a zero mean noise term $w = \eta(x) - (F\Phi r)(x)$. The iterations for TD also depend on a tuning parameter $\lambda \in [0, 1]$ and a stepsize schedule γ_t; for the case where $\lambda = 0$, the iterations take the form:

$$r_{t+1} = r_t + \gamma_t \phi(x_t)((F\Phi r_t)(x_t) + w_{t+1} - (\Phi r_t)(x_t)),$$

where $x_t \in S$ is generated by some sampling mechanism, $\phi(x) \in \Re^k = (\phi_1(x), \ldots, \phi_K(x))'$, and $(F\Phi r_t)(x_t) + w_{t+1}$ represents a stochastic sample generated and used by the algorithm. Roughly speaking, a random sample x_t is drawn, and then r_t is adapted to reduce the difference between $(F\Phi r_t)(x_t)$ and $(\Phi r_t)(x_t)$. There is a rich literature on convergence analysis of this algorithm [20, 13, 15, 23, 19, 28, 26, 21]. Though more complicated sampling mechanisms will be considered later, we will limit attention for now to a situation where each sample x_t is independently drawn from S with probabilities given by the diagonal elements of D.

A more sophisticated interpretation of Temporal Difference learning might view the algorithm as a generalization of stochastic steepest descent. A formula for stochastic steepest descent would be written as:

$$r_{t+1} = r_t + \gamma_t \phi(x_t)(J^*(x_t) + w_{t+1} - (\Phi r_t)(x_t)).$$

At each iteration, stochastic steepest descent estimates the gradient of the error function $\|J^* - \Phi r_t\|_D$, and then adapts the weights r_t in the direction of the estimated gradient. Note, in particular, that for stochastic steepest descent,

$$E[r_{t+1}|r_t] = r_t + \gamma_t \Phi' D(J^* - \Phi r_t) = r_t - \frac{1}{2}\nabla_r \|J^* - \Phi r_t\|_D^2.$$

TD differers from stochastic steepest descent in that it applies to problems for which J^* is unavailable, as is the gradient of the error function $\|J^* - \Phi r\|_D$. TD instead estimates the gradient of $\|F\Phi r_t - \Phi r\|_D^2$ with respect to r, and then adapts the weights r_t in the direction of the estimated gradient.

1.5 The Fixed Point Kalman Filter

A well known alternative to stochastic steepest descent is the Kalman filter, also known as recursive least-squares. While stochastic steepest descent chooses r_t to adapt $(\Phi r_t)(x_t)$ towards $J^*(x_t)$, recursive least-squares chooses r_t to give the best fit to J^* at all points $x_1, \ldots, x_t$ drawn up to that point. The Kalman filter is best known as a methods for recursively solving the least-squares problem to find r_{t+1} without having to store $x_1, \ldots, x_t$. Many features of the Kalman filter are specific to its more common applications in control theory and communications, so we do not consider them here. For the problem of estimating J^* from samples $J^*(x_1), \ldots, J^*(x_t)$, the iterates r_t generated by the Kalman filter satisfy

$$r_t = \operatorname*{argmin}_r \sum_{s=1}^{t-1} (J^*(x_s) + w_{s+1} - (\Phi r)(x_s))^2,$$

taking r_t to be the vector of minimum norm when there are multiple candidates r that minimize the right hand side. The iterates are computed recursively according to

$$r_{t+1} = r_t + \frac{1}{t} H_t \phi(x_t)(J^*(x_t) + w_{t+1} - (\Phi r_t)(x_t)), \tag{1}$$

where

$$H_t = \left(\frac{1}{t}\sum_{s=1}^{t} \phi(x_s)\phi'(x_s)\right)^{-1},$$

assuming that the summation $\sum_{s=1}^{t} \phi(x_s)\phi'(x_s)$ is nonsingular. As this summation may be singular for small values of t, any of three different methods are commonly used.

1. The pseudo-inverse is used instead of matrix inversion; i.e., $x = A^\dagger y$ minimizes $\|Ax - y\|$, and if there are multiple vectors attaining this minimum, it is the one of minimal norm.

2. The matrix is regularized, meaning that

$$H_t = \left(\frac{\epsilon}{t} I + \frac{1}{t}\sum_{s=1}^{t} \phi(x_s)\phi'(x_s)\right)^{-1},$$

3. Additional states $x_{-\iota}, \ldots, x_{-1}$ are drawn, until $\sum_{s=-\iota}^{0} \phi(x_s)\phi'(x_s)$ is nonsingular. At each iteration, the matrix H_t equals

$$H_t = \left(\frac{1}{t+\iota} \sum_{s=-\iota}^{t} \phi(x_s)\phi'(x_s) \right)^{-1}.$$

In this paper, we assume the first method is implemented.

The Kalman Filter generally converges in far fewer iterations than stochastic steepest descent. One interpretation for the faster convergence of the Kalman filter is that the iterates are identical if the matrix H_t happens to equal the identity I. This occurs when the basis functions are chosen to be orthonormal. The Kalman filter can be seen as a steepest descent algorithm that adaptively rescales the basis functions to compensate for functions that are chosen to be poorly scaled.

We now motivate the methods presented in this paper by analogy with the Kalman filter. Similarly with the case of stochastic steepest descent, when samples of the function J^* are unavailable, the Kalman filter is inappropriate. TD can be viewed as a generalization of stochastic steepest descent that is applicable when J^* is unavailable. In this paper, we study an analogous generalization of the Kalman filter, which we call the fixed point Kalman filter. The fixed point Kalman filter updates iterates according to

$$r_{t+1} = r_t + \gamma_t H_t \phi(x_t)((F\Phi r_t)(x_t) + w_{t+1} - (\Phi r_t)(x_t)), \qquad (2)$$

where γ_t is a sequence of scalar step sizes and H_t is a sequence of matrices converging to $(\Phi' D \Phi)^{-1}$.

There are many versions of the fixed point Kalman filter, each associated with a step size sequence γ_t, a sequence of matrices H_t. We discuss two versions:

1. The one most closely resembling the Kalman filter generates weights satisfying

$$r_t \;=\; \underset{r}{\operatorname{argmin}} \sum_{s=1}^{t-1} ((F\Phi r_s)(x_s) + w_{s+1} - (\Phi r)(x_s))^2 \qquad (3)$$

taking r_t to be the vector of minimum norm when there are multiple candidates r that minimize the right hand side. This is accomplished by setting the matrix H_t to

$$H_t = \left(\frac{1}{t} \sum_{s=1}^{t} \phi(x_s)\phi'(x_s) \right)^{\dagger},$$

and the step sizes to $\gamma_t = 1/t$.

2. In the previous version, the weights r_t are chosen so that Φr_t fits prior samples $(F\Phi r_{t-1})(x_t) + w_t, (F\Phi r_{t-2})(x_{t-2}) + w_{t-1}, \ldots$. Note that, since $\gamma_t = 1/t$, each prior sample is weighed to an equal extent. However, it seems that more recent samples should be more relevant than those collected in the distant past, because the weight vector r_t has evolved over time. In order to place more emphasis on recent samples, we can reduce the rate of decay of this step size sequence. For example, one could employ a sequence $\gamma_t = a/(a+t)$ for some large a. Slowing down the decay of the step size in this way can often lead to faster convergence. In particular, though $1/t$ is the optimal step size sequence for the traditional

Kalman filter which has a fixed “target” of J^*, the fact that the “target” Φr_m of the fixed point Kalman filter is evolving motivates maintenance of larger step sizes for a greater number of iterations.

1.6 Least–Squares TD

Least-squares temporal-difference learning (LSTD) is another generalization of the Kalman filter amenable to situations where the desired function J^* is not directly available. This algorithm was introduced by Bradtke and Barto [11], who also provide some convergence theory. An excellent discussion of the algorithm and extensions can also be found in [10]. Relevant convergence theory and further extensions are developed in [11] and [18]. In this section, we explain similarities and differences between the fixed point Kalman filter and LSTD, for the special case of $\lambda = 0$.

At each iteration, LSTD minimizes the empirical squared error between Φr and $\Pi F \Phi r$. Its iterates satisfy

$$r_{t+1} = \operatorname*{argmin}_r \sum_{s=1}^{t} ((F\Phi r)(x_s) + w_{s+1} - (\Phi r)(x_s))^2.$$

For problems involving autonomous systems or fixed policies, F is linear, and r_t is the solution to a linear least-squares problem, which can be efficiently solved. Empirical studies show that LSTD converges faster than TD, when F is linear [11, 10]. However, for the more general case where F is nonlinear, the iterates are difficult to compute and LSTD cannot be applied in a straightforward manner. In such cases, the fixed point Kalman filter can be viewed as a relaxation of LSTD whose iterates satisfy a slightly different equation:

$$r_{t+1} = \operatorname*{argmin}_r \sum_{s=1}^{t} ((F\Phi r_s)(x_s) + w_{s+1} - (\Phi r)(x_s))^2,$$

which can be efficiently computed.

1.7 Contributions and Organization of the Paper

The main contributions of this paper consist of:

1. A proof that the fixed point Kalman filter converges.
2. Computational experiments demonstrating potential advantages over TD.

The main benefit offered by the fixed point Kalman filter is that it converges in fewer iterations than TD. The compute time required per iteration grows, but even factoring in this increase, the rate of convergence in terms of compute time should compare favorably against that offered by TD. Furthermore, in the “learning” context, where samples used to update iterates are associated with empirical observations, the faster convergence delivered by the fixed point Kalman filter translates to more effective use of observations.

This paper represents an extended version of an earlier one [12]. This earlier paper did not present the convergence proofs included in the current paper. This extended paper also reports empirical results from new case studies.

The remainder of the paper is organized as follows. In the next section, we present and prove a convergence theorem. This theorem is proved using a supermartigale convergence theorem as a starting point because to the best of our knowledge, there are no general stochastic approximation theorems available in the literature that apply directly to our particular problem. In Section 3, we present computational results. The results are generated through experiments involving an optimal stopping problem and a queueing problem. In the optimal stopping context, the operator F of interest is a contraction with respect to a weighted Euclidean norm $\|\cdot\|_D$, along the lines we have discussed, and both TD and the fixed point Kalman filter are guaranteed to converge. This is not true, however, in the queueing context. Here, we apply TD and the fixed point Kalman filter even though the operator F is not a contraction with respect to a weighted Euclidean norm. Nevertheless, our computational results are promising, as both algorithms appear to generate effective control policies. Closing remarks are made in a concluding section.

2 A Convergence Theorem

In this section, we establish convergence of the fixed point Kalman filter. Our development closely follows Tsitsiklis and Van Roy's [23, 26] proof of convergence for TD, though significant additional work is required. The convergence proof in [23, 26] makes use of a general stochastic approximation result from the text of Benveniste, Métivier, and Priouret [3]. To the best of our knowledge, the fixed point Kalman filter cannot be reduced to a form that satisfies the assumptions imposed by results in this text.

In the next section, we provide an extension to a convergence theorem presented in [3]. This extension can be used to establish convergence of the fixed point Kalman filter, which we will do subsequently. It is worth noting that, though the result of the next section extends that of [3] in one dimension, to keep the exposition brief, assumptions are made that in many ways limit the scope of the result relative to that in [3].

2.1 A General Convergence Theorem

We consider a stochastic approximation algorithm that generates a sequence of vectors $\theta_t \in \Re^n$ according to

$$\theta_{t+1} = \theta_t + \gamma_t \Gamma_t h(\theta_t, x_t, x_{t+1}), \tag{4}$$

where γ_t is a sequence of scalar step sizes, Γ_t is a sequence of $n \times n$ matrices, h is a function from $\Re^n$ to $\Re^n$, and $\{x_t\}$ is a sequence of samples drawn from a finite set S. We take all random variables to be defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with respect to an increasing sequence of σ–fields $\mathcal{F}_t \subseteq \mathcal{F}$. We will impose several assumptions on the variables in the above iteration. Our first assumption concerns the step sizes and is standard in flavor.

Assumption 1 *The step size sequence γ_t is deterministic (predetermined), nonincreasing, and satisfies*

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \quad \text{and} \quad \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

A second assumption relates to the samples x_t.

Assumption 2 *Each x_t is measurable with respect to $\mathcal{F}_t$, and the sequence $x_0, x_1, x_2, \ldots$ is generated by an irreducible aperiodic Markov chain. Furthermore, x_0 is drawn from the invariant distribution of this Markov chain.*

Since the process is generated by an irreducible aperiodic Markov chain, there is a unique invariant distribution. We will denote this distribution by π. Also, let D be the diagonal matrix whose diagonal elements are $\pi(1), \ldots, \pi(n)$, and let $\|\cdot\|_D$ be the weighted Euclidean norm defined by $\|J\|_D^2 = J'DJ$, for any J.

Our next assumption defines requirements on Γ_t, which involve its convergence to the identity matrix I. The assumption may appear somewhat nonstandard; it is motivated by the desire to bound the speed of convergence of the pseudo-inverse matrix $H_t = (\frac{1}{t}\sum_{s=1}^{t} \phi(x_s)\phi'(x_s))^\dagger$ used in the Kalman filter iterates. If the pseudo-inverse in H_t is taken on a poorly conditioned matrix, the norm $\|H_t\|_D$ may take arbitrarily large values, complicating the analysis of its convergence. As a result, we found it most convenient to bound the rate of convergence of H_t as a function of $\sup_\tau \|H_t\|_D$.

Assumption 3 *Each Γ_t is $\mathcal{F}_t$–measurable, and the sequence Γ_t converges to I with probability one. Furthermore, there exists a nonincreasing deterministic scalar sequence β_t and a function $\psi : \Re \mapsto \Re$ such that $\|\Gamma_{t+1} - \Gamma_t\|_D \leq \beta_t \psi(\sup_\tau \|\Gamma_\tau\|_D)$, for all t, and $\sum_t \beta_t \gamma_t < \infty$, with probability one.*

The discussion of Assumption 9 will more explicitly establish the connection between H_t used in the Kalman filter, and the Γ_t used in the general convergence theorem. The next assumption calls for Lipschitz continuity of h with respect to θ.

Assumption 4 *There exists a scalar C such that for all $x, y, \theta, \overline{\theta}$,*

$$\|h(\theta, x, y) - h(\overline{\theta}, x, y)\|_D \leq C\|\theta - \overline{\theta}\|_D.$$

Note that this assumption implies that there exists a scalar C such that

$$\|h(\theta, x, y)\|_D \leq C(1 + \|\theta\|_D).$$

Furthermore, this assumption implies that the steady-state expectation $\overline{h}(\theta) = E[h(\theta, x_t, x_{t+1})]$ of h is also Lipschitz continuous.

Our final assumption ensures that $\overline{h}$ appropriately orients parameter adjustments.

Assumption 5 *There exists a scalar $c > 0$ such that*

$$\theta' D\overline{h}(\theta) \leq -c\|\theta\|_D^2,$$

for all $\theta \in \Re^n$, and $\overline{h}(0) = 0$.

Note that this assumption implies that $\overline{h}$ possesses a unique root at $\theta = 0$.

For now, we state the general convergence theorem.

Theorem 1 *Let Assumptions 1–5 hold. Then, for any $\theta_0 \in \Re^n$, the sequence θ_t generated according to Equation (4) converges to 0 with probability one.*

We defer the proof of this theorem to the appendix. This theorem will be used in the next section to prove the convergence of the fixed point Kalman filter, where we further discuss the above assumptions and how they can be verified for the algorithm.

2.2 Convergence of the Fixed Point Kalman Filter

Recall that the fixed point Kalman filter relies on a sequence of matrices H_t and generates iterates according to

$$r_{t+1} = r_t + \gamma_t H_t \phi(x_t)((F\Phi r_t)(x_t) + w_{t+1} - (\Phi r_t)(x_t)).$$

We once again take all random variables to be defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with respect to an increasing sequence of σ–fields $\mathcal{F}_t \in \mathcal{F}$. As in the previous section, we will assume that $x_t \in S$ for a finite set S, and that $x_0, x_1, x_2, \ldots$ is generated by an irreducible aperiodic Markov chain with invariant distribution π, and we take D to be the diagonal matrix with diagonal entries given by π. We introduce four assumptions. The first is that F is a contraction.

Assumption 6 *There exists $\alpha \in [0, 1)$ such that for all $J, \overline{J} \in \Re^n$,*

$$\|FJ - F\overline{J}\|_D \leq \alpha \|J - \overline{J}\|_D.$$

We let J^* denote the fixed point of F.

Our second assumption ensures that the basis functions are linearly independent. It is not absolutely necessary, but is introduced to simplify the exposition.

Assumption 7 *The columns of Φ are linearly independent.*

Let Π denote the projection matrix $\Phi(\Phi' D\Phi)^{-1}\Phi' D$, and recall that ΠF is a contraction with contraction factor α. We denote the fixed point of ΠF by Φr^*. By Assumption 7, this identifies a unique r^*.

The next assumption concerns the noise w_t, formalizing the notion of a stochastic sample $\eta(x)$ that was introduced in section 1.3.

Assumption 8 *For each t, w_t is generated according to $w_t = w(r_t, x_t, x_{t+1})$, for some function w. Furthermore, $E[w(\theta, x_t, x_{t+1})|x_t] = 0$, for all r, x_t, and there exists a scalar C such that*

$$\|w(r, x, y) - w(\overline{r}, x, y)\|_D \leq C\|r - \overline{r}\|_D,$$

for all r, x, y.

Roughly speaking, the stochastic sample approximating $(F\Phi r_t)(x_t)$ must be unbiased and have bounded error for Assumption 8 to hold. For the experiments in this paper, the fact that the state space is finite is sufficient to establish the bound given in this assumption.

Our final assumption applies to the convergence of H_t. As stated before, H_t may take large values before converging, so its rate of convergence can be difficult to bound in the usual ways.

Assumption 9 *Each H_t is $\mathcal{F}_t$-measurable, and the sequence H_t converges to $(\Phi' D\Phi)^{-1}$, with probability one. Furthermore, there exists a nonincreasing deterministic scalar sequence β_t and a function $\psi : \Re \mapsto \Re$ such that $\|H_{t+1} - H_t\| \leq \beta_t \psi(\sup_\tau \|H_\tau\|_D) < \infty$, for all t, and $\sum \beta_t \gamma_t < \infty$.*

To motivate why this assumption should hold for practical algorithms, consider a sequence H_t defined by

$$H_t = \left(\frac{1}{t} \sum_{s=1}^{t} \phi(x_s)\phi'(x_s) \right)^{\dagger}.$$

Clearly, H_t converges to $(\Phi' D\Phi)^{-1}$ (as $\sum_{s=1}^{t} \phi(x_s)\phi'(x_s)$ converges to $\Phi' D\Phi$, H_t must converge to its inverse, since matrix inversion is locally continuous for nonsingular matrices). Let us discuss why the remaining conditions are satisfied. The sequence can alternatively be written as $H_t = L_t^{\dagger}$, where L_t evolves according to

$$L_{t+1} = L_t + \frac{1}{t+1}(\phi(x_{t+1})\phi'(x_{t+1}) - L_t).$$

Let $\overline{H} = \sup_t \|H_t\|$. It holds that $\overline{H} < \infty$, since H_t is non-singular for all t. Since matrix inversion is locally Lipschitz over the space of positive definite matrices, there is a scalar $c(\overline{H})$ such that

$$\|L^{\dagger} - M^{\dagger}\| \leq c(\overline{H})\|L - M\|_D,$$

for $\|L^{\dagger}\|_D \leq \overline{H}$, $\|M^{\dagger}\|_D \leq \overline{H}$. It follows that

$$\begin{aligned}
\|H_{t+1} - H_t\|_D &= \left\| \frac{t}{t+1} H_{t+1} - H_t \right\|_D + \left\| \frac{1}{t+1} H_{t+1} \right\|_D \\
&\leq c(\overline{H}) \left\| \frac{t+1}{t} L_{t+1} - L_t \right\|_D + \frac{\overline{H}}{t} \\
&= \frac{c(\overline{H})}{t} \|\phi(x_{t+1})\phi'(x_{t+1})\|_D + \frac{\overline{H}}{t}.
\end{aligned}$$

Therefore,

$$\|H_{t+1} - H_t\|_D \leq \frac{\psi(\overline{H})}{t},$$

for some scalar $\psi(\overline{H})$. Hence, the assumption is satisfied by setting $\beta_t = 1/t$, in which case $\sum \beta_t \gamma_t < \infty$ holds since we assume that $\sum \gamma_t^2 < \infty$.

We now state the convergence theorem for the fixed point Kalman filter.

Theorem 2 *Let Assumptions 1, 2, 6, 7, 8, and 9 hold. Then, for any $r_0 \in \Re^K$, the sequence r_t generated by the fixed point Kalman filter converges to r^* with probability one.*

Proof: We will apply Theorem 1 to prove the present theorem. In doing so, we will associate θ_t with $\Phi r_t - \Phi r^*$. We set

$$\begin{aligned}
h(\theta, x, y) &= \Phi(\Phi' D\Phi)^{-1}\phi(x)((F\Phi r)(x) + w(\Phi r, x, y) - (\Phi r)(x)) \\
\Gamma_t &= \Phi H_t \Phi' D\Phi(\Phi'\Phi)^{-1}\Phi' + I - \Phi(\Phi'\Phi)^{-1}\Phi'.
\end{aligned}$$

It can be seen by inspection that the fixed point Kalman filter satisfies $\theta_{t+1} = \theta_t + \gamma_t \Gamma_t h(\theta_t, x_t, x_{t+1})$. It also follows that

$$\begin{aligned}
\overline{h}(\theta) &= \Phi(\Phi' D\Phi)^{-1}\Phi' D(F\theta - \theta) \\
&= \Pi(F\Phi r - \Phi r) \\
&= \Pi F\Phi r - \Phi r.
\end{aligned}$$

Assumptions 1 and 2 are common to Theorems 1 and 2. To establish convergence of $\theta_t = \Phi r_t - \Phi r^*$ to 0, we must show that Assumptions 3, 4, and 5 hold.

We begin with Assumption 3. Since H_t converges to $(\Phi' D \Phi)^{-1}$, Γ_t converges to I. Let β_t be a sequence and ψ a function satisfying the conditions of Assumption 9. Since the mapping from H_t to Γ_t is affine and Φ has full rank, it is easy to see that there exist scalars c_1 and c_2 such that

$$\|\Gamma_{t+1} - \Gamma_t\|_D \leq c_1 \|H_{t+1} - H_t\|_D \quad \text{and} \quad \|H_t\|_D \leq c_2 \|\Gamma_t\|_D.$$

It follows from Assumption 9 that

$$\|\Gamma_{t+1} - \Gamma_t\|_D \leq c_1 \beta_t \sup_\tau \psi(c_2 \|\Gamma_\tau\|_D).$$

Hence, Assumption 3 holds, being satisfied by a sequence $\overline{\beta}_t = c_1 \beta_t$ and a function $\overline{\psi}(\overline{\Gamma}) = \psi(c_2 \overline{\Gamma})$.

Assumption 4 follows immediately from the fact that F is a contraction mapping (Assumption 6) and w is Lipschitz continuous in θ (Assumption 8). We are left with the task of establishing validity of Assumption 5. For any $\theta = \Phi r - \Phi r^*$, we have

$$\begin{aligned}
\theta' D \overline{h}(\theta) &= (\Phi r - \Phi r^*)' D (\Pi F \Phi r - \Phi r) \\
&= (\Phi r - \Phi r^*)' D (\Pi F \Phi r - \Phi r^*) + (\Phi r - \Phi r^*)' D (\Phi r^* - \Phi r) \\
&\leq \|\Phi r - \Phi r^*\|_D \|\Pi F \Phi r - \Phi r^*\|_D - \|\Phi r - \Phi r^*\|_D^2 \\
&\leq (\alpha - 1) \|\Phi r - \Phi r^*\|_D^2,
\end{aligned}$$

where the first inequality follows from Cauchy–Schwartz, and the second follows from the fact that F is a contraction (Assumption 6). Note that $1 - \alpha > 0$. Furthermore,

$$\overline{h}(0) = \Pi F \Phi r^* - \Phi r^* = 0.$$

We have verified that all assumptions of Theorem 1 are satisfied. Theorem 2 follows.
qed

3 Computational Results

In this section, we discuss experimental results comparing TD(0) and the fixed-point Kalman filter. Our first case study involves an optimal stopping problem introduced in [23]. In this case, the fixed-point Kalman filter converges faster than TD(0), as anticipated. In a second case study, we consider a queueing problem for which an optimal policy is known. In this context, neither TD(0) nor the fixed point Kalman filter are guaranteed to converge. Nevertheless, both appear to deliver good control policies, with the fixed point Kalman filter converging faster and exhibiting more robustness than TD(0).

3.1 An Optimal Stopping Problem

We consider a factitious financial derivative whose payoff at the time of exercise is equal to the current price of a particular stock divided by its price one hundred days ago.

The security, once bought, may be held indefinitely by its holder while the price of the stock fluctuates autonomously, until its exercise is desired. As a result, the payoff of the derivative also fluctuates as an autonomous process until the decision is made to stop the process (by exercising the option), at which time a reward is collected.

It has been shown [16] that the value of this security is equal to the optimal reward for a particular optimal stopping problem, whose particulars we introduce here. Let ρ be the constant continuously compounded short-term interest rate of a money market available for investment. Let the actual price of the stock p_t be modeled by a geometric Brownian motion

$$p_t = p_{-100} + \int_{s=-100}^{t} \mu\rho p_s ds + \int_{s=-100}^{t} \rho p_s dw_s,$$

for some positive scalars p_0, μ, and σ, and a standard Brownian motion w_t. Then the value of the security is equal to the discounted (with discount factor $e^{-\rho}$) payoff of the security under the optimal exercise policy, but with the stock price modeled as $\tilde{p}_t$ rather rather than p_t , where p_t evolves according to

$$d\tilde{p}_t = \rho\tilde{p}_t det + \sigma\tilde{p}_t dw_t.$$

We now cast the task of finding an optimal exercise policy as an optimal stopping problem. Let $\{x_t | t = 0, 1, 2, \ldots\}$ be a Markov process where $x_t \in \Re^{100}$ and $x_t(i) = \tilde{p}_{t-i} - \tilde{p}_{t-100}$, for all $i = 0, 1, \ldots, 100$. Let $G(x) = x(100)$ and $\alpha = e^{-\rho}$. Let τ^* be the stopping time that produces the optimal expected reward

$$\sup_{\tau > 0} \mathrm{E}[\alpha^{\tau} G(x_{\tau}) | x_0 = x].$$

3.1.1 Approximations

Because the state space $\Re^{100}$ of the process x_t is quite large, we cannot hope to compute an optimal stopping policy to produce τ^*. Instead, we compute a suboptimal policy with the stopping time τ dependent upon features collected from x. Our experiment replicates the one reported in [26]. The features, represented by a set of 16 basis functions $\Phi = [\ \phi_1 \ \cdots \ \phi_{16}\ ]$, span a space of much lower dimension than $\Re^{100}$. In [24], a stopping policy is derived from a linear combination of these basis functions. In [24], a stopping policy is derived from a linear combination of these basis functions, with the weights produced from temporal-difference learning algorithm.

We repeat the experiment, recalculating the weights using the fixed point Kalman filter, to show that the weights produced by the fixed point Kalman filter converges to their steady state values much sooner than those produced by temporal difference learning.

Let $j = i/50 - 1$. The basis functions were chosen heuristically, using arguments that can be found in [24]. For reference, the functions were as follows:

$$\begin{aligned} \phi_1(x) &= 1 \\ \phi_2(x) &= G(x) \\ \phi_3(x) &= \min_{i=1,\ldots,100} x(i) - 1 \end{aligned}$$

$$
\begin{aligned}
\phi_4(x) &= \max_{i=1,\ldots,100} x(i) - 1 \\
\phi_5(x) &= \frac{1}{50} \operatorname*{argmin}_{i=1,\ldots,100} x(i) - 1 \\
\phi_6(x) &= \frac{1}{50} \operatorname*{argmin}_{i=1,\ldots,100} x(i) - 1 \\
\phi_7(x) &= \frac{1}{100} \sum_{i=1}^{100} \frac{x(i) - 1}{\sqrt{2}} \\
\phi_8(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{3}{2}} j \\
\phi_9(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{5}{2}} \left(\frac{3j^2 - 1}{2} \right) \\
\phi_{10}(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{7}{2}} \left(\frac{5j^2 - 3j}{2} \right) \\
\phi_{11}(x) &= \phi_2(x)\phi_3(x) \\
\phi_{12}(x) &= \phi_2(x)\phi_4(x) \\
\phi_{13}(x) &= \phi_2(x)\phi_7(x) \\
\phi_{14}(x) &= \phi_2(x)\phi_8(x) \\
\phi_{15}(x) &= \phi_2(x)\phi_9(x) \\
\phi_{16}(x) &= \phi_2(x)\phi_{10}(x)
\end{aligned}
$$

However, for our experiment, we worked only with basis function $\phi_1, \ldots, \phi_{10}$. When the full array of basis functions $\phi_1, \ldots, \phi_{16}$ was tested with the fixed point Kalman filter, it appeared that the matrix H_i^{-1} tended towards singularity, suggesting that $\phi_{11}, \ldots, \phi_{16}$ were contained in the span of $\phi_1, \ldots, \phi_{10}$.

3.1.2 Experimental Results

Using our ten basis functions, we generated a sequence of weight vectors $r_0, r_1, \ldots, r_{1\times 10^7}$ by initializing $r_0 = 0$ and running ten million iterations of TD(0) with a constant step size of $\gamma_t = .001$, and a second sequence of weight vectors by initializing r_0 as before and running ten million iterations of the fixed point Kalman filter with a nonconstant step size of $\gamma_t = .01\frac{10000}{10000+t}$. For reference, we mention that the estimate $(F\Phi r_t)(x_t) + w_{t+1}$ (for both TD and the fixed point Kalman filter) is given by

$$(F\Phi r_t)(x_t) + w_{t+1} = \alpha \max((\Phi r_t)(x_{t+1}), x_t(100)).$$

At each iteration, the feature weights r_t were recorded. As r_t evolved, the performance of the policy was periodically evaluated by drawing 1000 test trajectories. For the policy at that iterate, the discounted payoff was computed for each of the test trajectories, and the average discounted payoff was taken as measure of the performance of the current policy. For each policy that was tested, the variance of the resulting empirical average was estimated to be less than .001.

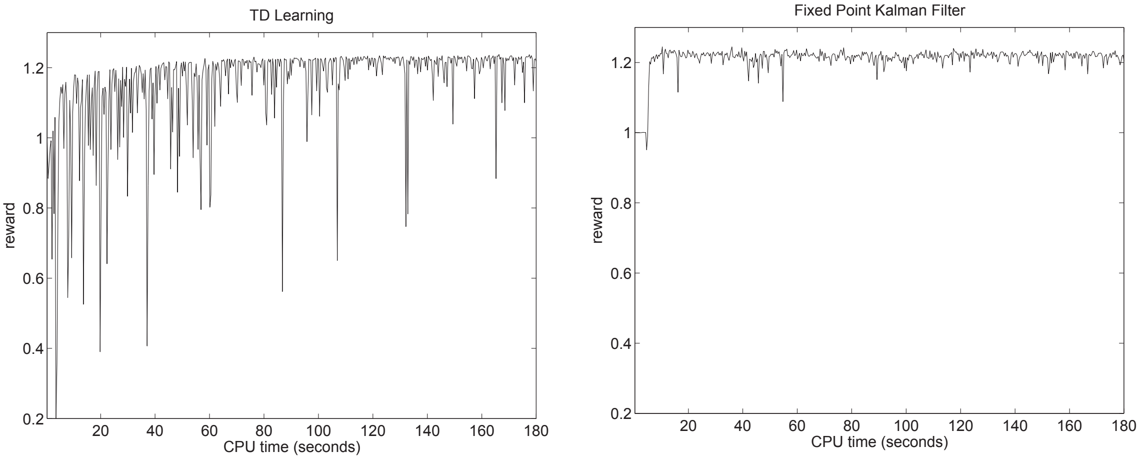

Figure 1: The average discounted payoff of two evolving policies for options pricing.

For both TD and the fixed point Kalman Filter, a range of step size sequences were tested, of the form $\gamma_t = a\frac{b}{b+t}$, for $a \in \{10^1, 10^0, 10^{-1}, \ldots, 10^{-5}\}$, and for $b \in \{10^3, 10^4, \infty\}$ (with $b = \infty$ meaning that $\gamma_t = a$ for all t). The schedule whose policy performance converged fastest was chosen. For the fixed point Kalman filter, γ_t was set to zero for the first twenty thousand iterations, allowing the matrix H_t to adapt (from its initialized value $H_0 = 100I$) before the weights r_t were allowed to change.

Figure 1 plots the performance of the evolving policy given by the feature weight trajectories under the fixed point Kalman filter and under TD(0). For this application, it is uncertain what the optimal policy is. As a result, it is difficult to rate the optimality of the policy given by TD(0) or by the fixed point Kalman filter. In [26], however, the resulting policy was shown to outperform a baseline policy. The fixed point Kalman filter produces payoffs that converge faster than those given by the TD(0) policy – in fact, the payoffs converge almost immediately when the policy is produced using the fixed point Kalman filter.

In order to quickly adapt from the initial weight values, the stepsize was chosen to be large and to decay slowly. As a result, although the payoffs converge rapidly using the Kalman filter, the feature weights themselves are not as well behaved. Figure 2 plots the feature weights as they are adapted by the fixed point Kalman filter and by TD(0). The weights of the fixed point Kalman filter oscillate noisily. However, the fluctuation in the weights does not seem to affect the policy. It could be that the while oscillations in r_t were large, the resulting change in $(\Phi r_t)(x)$ was small for states that were commonly visited. It seems that the values of the weights could possibly be oscillating about a fixed point; however, it is unclear from the figure. The weights produced by TD(0) are less noisy, and it seems clear that the TD(0) weights have not reached their steady state values yet.

To produces less noisy weights, the Kalman filter was rerun with a different stepsize schedule. The difference was that this time, γ_t was fixed to zero for the first two-hundred thousand iterations, giving H_t more time to adapt. The result was faster,

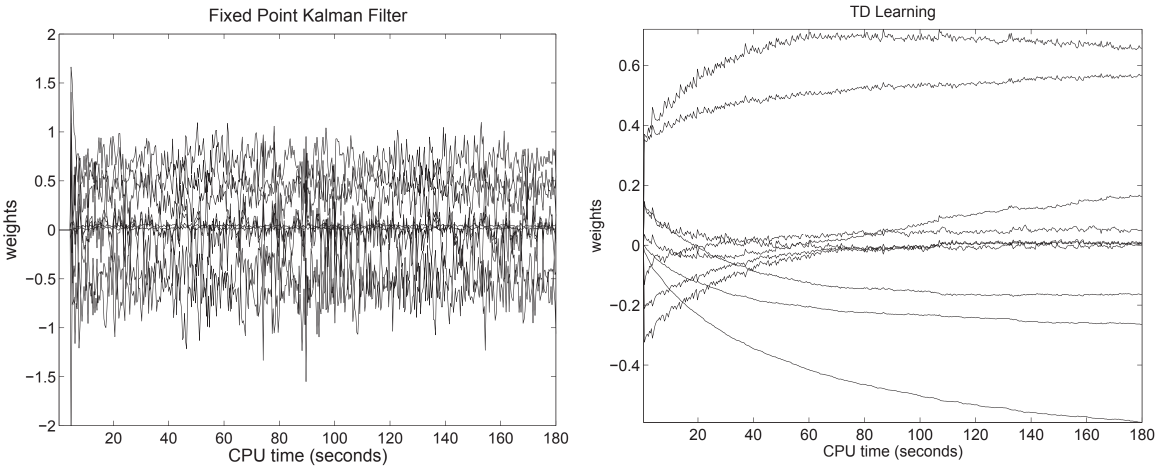

Figure 2: The feature weights of two evolving policies for options pricing.

smoother convergence of the weights to their steady-state values. The policy and weights of the alternate stepsize schedule for the fixed point Kalman filter are shown in Figure 3.

The options pricing experiment was repeated, for the same range of stepsize schedules, and using the same ten basis functions, but with a poorly scaled eleventh basis function added to the feature set. This basis function was random noise with an expected value of 455. The behavior of the Kalman filter remained unchanged; however, unless much smaller step-sizes were used, the weights generated by temporal difference learning would blow up to the point of overflow. As a result, as shown in Figure 4, the best choice of stepsize for TD produced a much-degraded convergence rate. The poor performance of the policy given by TD is explained by the magnitude of the weights, shown in Figure 5. Because a smaller stepsize is chosen, the weights adapt much more slowly in temporal-difference learning with the exception of the weight corresponding to the poorly scaled feature. In contrast, the fixed point Kalman filter still converges rapidly to the policy performance attained in the previous tests. The reason for this is that the Kalman filter iterates automatically rescales the basis functions by the matrix H_t , where H_t is depends on the empirical statistics of each basis function. In this example, the eleventh basis function is statistically much larger than, and uncorrelated with, the other basis functions. As a result H_t gives low weight to the eleventh basis function without changing the weights of the other basis functions.

It is worth noting that in each of the examples, most of the adaptation for the fixed point Kalman filter was unnecessary, since the weights would oscillate noisily without changing the policy. For this study, the extra simulations were illuminating for pedagogical reasons, in that they showed the policy could be stable even when the weights are noisy. In practice, however, a stopping criterion for the filter would have been convenient, if such a rule could be found.

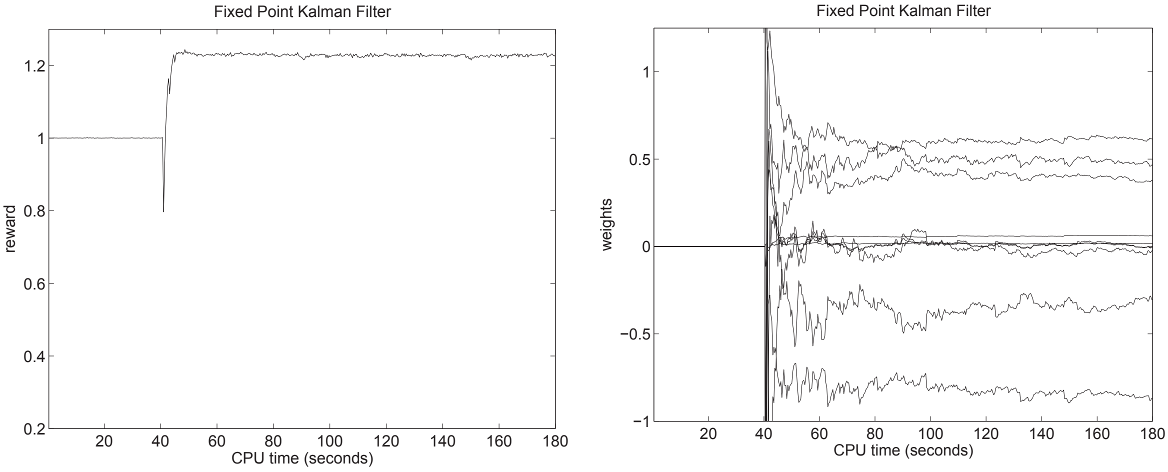

Figure 3: The average discounted payoff, and the underlying feature weights, of an alternate evolving policy for the Fixed Point Kalman Filters.

3.2 Queueing Networks

As a second experiment, we applied temporal difference learning and the fixed point Kalman filter to the problem of designing a controller for a queueing network. A queueing network consists of d queues, each of which can be thought of as a state variable $x^{(i)}, i = 1, \ldots, d$, taking on values in $\{0, 1, 2, \ldots\}$. The state variable $x^{(i)}$ represents the number of jobs in the queue; when a job arrives at queue i then the state variable $x^{(i)}$ is incremented by one, and when a job leaves queue i then $x^{(i)}$ is decremented by one.

For our discussion of queueing networks, we depart from the most general formulations, and define time as a series of discrete instances $\{t = 1, 2, 3, \ldots\}$ (although continuous time formulations are also possible). At each instance of t, the controller chooses to service some of the jobs in the queues. Generally, not all the resident jobs in a queueing network can be serviced at once, and the probability of a job leaving or arriving at a queue at time t depends on how many jobs were chosen for service at time t and what queues those jobs resided in.

Let U denote the space of all possible service allocations, and let u_t be the particular decision made by the controller at time t. Let $x_t X$ be a d dimensional vector where $x_t^{(i)}$ denotes the length of queue i at time t and X denotes the space of all possible queue lengths. Then $x_1, x_2, \ldots$ is a Markov decision process; x_t depends probabilistically only on the previous state x_{t-1} and the control decision u_{t-1} .

At each instance t, a cost is imposed based on the current queue lengths x_t. The goal is to design a controller that maps states to actions so as to minimize the expected discounted cost. We write the expected discounted cost as $\mathrm{E}\left[\sum_{i=1}^{\infty} \alpha^i c(x_i)|x_0\right]$, α is the discount factor, x_0 is the initial state, and $c(x_i)$ is the cost when the queue lengths are given by x_t.

In many respects, the problem of designing a controller for a queueing network is

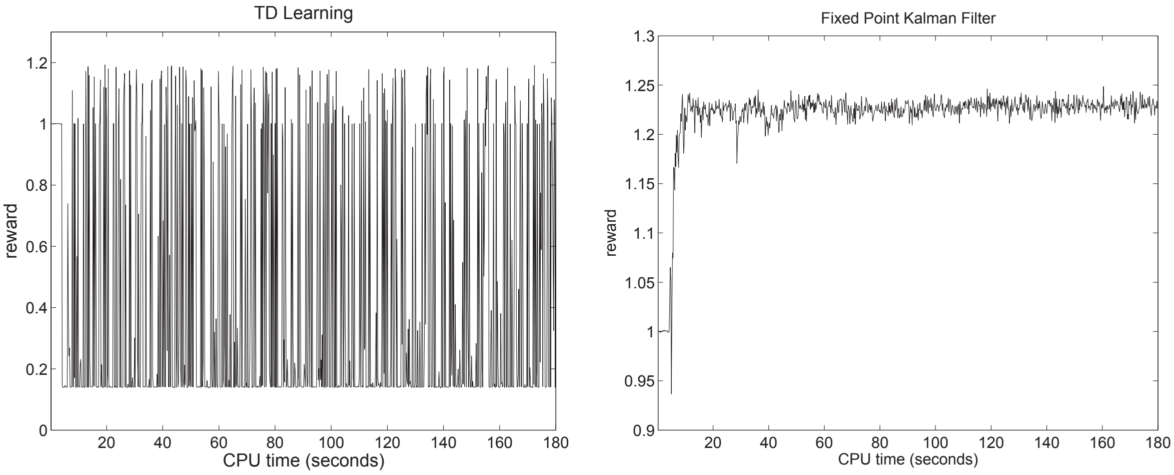

Figure 4: The average discounted payoffs of two evolving policies for options pricing, with degraded basis functions.

similar to the optimal stopping problem from the previous section. The costs $c(x_i)$, for example, can be thought of as negative rewards. However, the action space is quite different for the two problems. As a result, the contraction F for problems such as the queueing network will be different than the contraction seen in the optimal stopping problem.

3.2.1 Finding the Optimal Policy without Special Problem Structure

For such a queueing system, even without any additional problem structure, the optimal control policy can be found by calculating the fixed point of a dynamic programming operator F that is also a contraction (although not in the sense of the $\|\cdot\|_D$ norm). We define the contraction F here. Let $X \times U$ be the space of all allowable state action pairs. Let $Q : X \times U \rightarrow \Re$ be a function mapping state action pairs to real values. Let $p_{x,u,y}$ be the probability of transitioning from state x to state y given action u. Then the contraction F is defined pointwise by

$$FQ(x,u) \quad = \quad c(x) + \alpha \sum_{y \in X} p_{x,u,y} \min_u Q(y,u).$$

Let Q^* be the fixed point of F. If J^* is the value function satisfying Bellman's equation

$$J^*(x) = \min_u c(x) + \alpha \sum_{y \in X} p_{x,u,y} J^*(y),$$

then the function Q^* is defined by

$$Q^*(x,u) = c(x) + \alpha \sum_{y \in X} p_{x,u,y} J^*(y),$$

and Q^* satisfies

$$\min_u Q^*(x,u) = J^*(x),$$

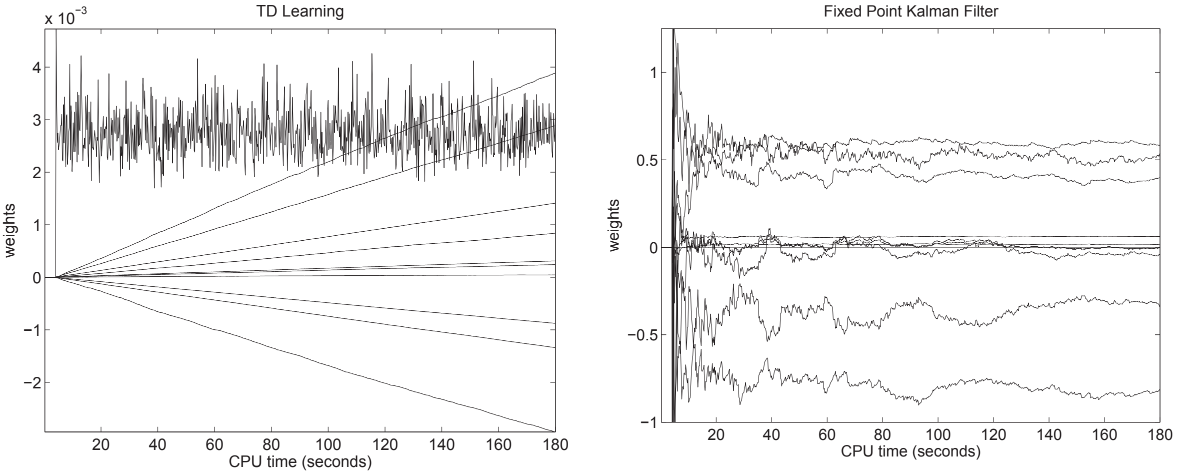

Figure 5: The feature weights for two evolving policies for options pricing, with degraded basis functions.

so that the optimal control policy for the queueing network is to choose u_t at each time t to solve

$$\min_u Q^*(x_t, u).$$

The reader may find detailed discussions of Q-learning in [7]. We state here that, as with the value function J^*, the enormous (possibly infinite) size of $X \times U$ precludes the exact computation of the fixed point Q^* of F. Instead, we can try to approximate the fixed point F using the methods described in this paper, temporal difference learning and the fixed point Kalman filter. Since there exists no norm $\|\cdot\|_D$ such that F satisfies the assumptions of Theorem 1, the theoretical results of this paper do not apply.

3.2.2 Klimov's Problem: a Queueing Network with Special Structure

For our experiment we will work with a special case of the queueing problem known as *Klimov's problem.* For this problem, there optimal controller has already been derived [27]. We will use TD and the Fixed Point Kalman Filter to generate new controllers without any foreknowledge of the optimal control. For this problem, we can easily find the optimal bounds on the performance of our controller that we will produce through fixed point approximation methods.

a Description of Klimov's Problem In Klimov's problem (also known as Cashier's Nightmare), we are given a queueing network with d queues, and the following additional assumptions.

1. Exactly one job is serviced at any time t.
2. The serviced job does not leave the network; instead it is placed in a new queue.
3. All other jobs remain in their queues.

As a result, the total number of jobs always remains constant at some initial value that we denote by k.

To complete the description of the network, we formulate the state transition probabilities and costs. The cost $c(x)$ at state x is equal to $g'x$ for some vector of costs g. This means that the cost is given by a weighted sum of the queue lengths. We will introduce $\hat{x}$ as an alternate representation of the state that allows for a simpler description of the state transition probabilities. Since the number of jobs remain constant, we can enumerate the jobs $1, \ldots, k$. Let $\hat{x}_t^{(i)} \in \{1, \ldots, d\}$ denote the queue $1, \ldots, d$ that the ith job resides in at time t. Let $\hat{x}_t \in \{1, \ldots, d\}^k$ denote a state representation where the ith element of $\hat{x}_t$ is given by $\hat{x}_t^{(i)}$. Note that x_t can be found from $\hat{x}_t$.

We now introduce the state transition probabilities of $\hat{x}$. Let $P \in \Re^{d \times d}$ be a matrix whose i, jth entry denotes transition probability p_{ij} from queue i to queue j, for $i, j \in \{1, \ldots, d\}$. Let $\hat{u}_t \in \{1, \ldots, k\}$ denote the job being serviced at time t. To reduce notational clutter, we will denote $\hat{u}_t$ by u and $\hat{x}_t^{(u)}$ by l. Then, given $\hat{x}_t$, $\hat{x}_{t+1}$ is given by the following rule: for all $i \neq u$,

$$\hat{x}_{t+1}^{(i)} = \hat{x}_t^{(i)}.$$

For $i = u$,

$$\hat{x}_{t+1}^{(i)} = j \text{ w.p. } p_{lj}.$$

We note that a more general formulation of Klimov's problem can be found in the original solution [27]. In that paper, we note that the inter-service waiting time is a exponential random variable (whereas in our case it is always taken to be 1), and so time is defined by a Poisson process $\{t = t_1, t_2, t_3, \ldots\}$, with the goal to minimize $\mathrm{E}[\sum_{i=1}^{\infty} \alpha^{t_i} c(x_{t_i}) | x_0]$. However, the continuous and discrete time versions of Klimov's problem both produce the same controller, if the discount factor and cost per state are related by a factor that depends on the interarrival times $\mathrm{E}[t_{i+1} - t_i]$ of the continuous time formulation (For a more complete discussion of continuous and discrete time dynamic programming problems, see [5] – specifically Vol. 2, Section 5.1).

The Optimal Controller For the queueing network as described in Klimovs problem, the optimal service policy can be constructed without knowledge of the value function. The optimal policy in this case can be stated as a priority list of queues, where at each iteration the non-empty queue with the highest priority on the list is serviced. As a result, the lowest priority queue is never serviced unless it is the only non-empty queue in the network. If all jobs reside in the lowest priority queue, then the optimal policy is to service any job in that queue, until a single job leaves the lowest priority queue. If all jobs except one reside in the lowest priority network, then the optimal policy will be to service that job continuously, until all jobs reside in the lowest priority queue. As a result, the stationary distribution under the optimal policy assigns non-zero probability to only those states where at most one job resides outside the lowest priority queue. The lowest priority queue will be the one with the lowest cost.

The exact priority of the queues is computable and will depend on the costs and transition probabilities of the queueing network. However, since the lowest priority queue is known to be the one with lowest cost, it is simple to construct a policy that achieves the optimal stationary distribution, without computation of the actual priorities. The optimal stationary distribution is achieved by any policy that services the lowest priority queue only if no other non-empty queues exist.

3.2.3 Experimental Parameters

Although the optimal policy can be found without solving for the Value Function in the case of Klimov's problem, for our experiments we will try to approximate the optimal policy of a network satisfying the assumptions of Klimov's problem, using fixed point approximation techniques. The goal of our experiment is to solve a problem where exact fixed point calculation fail, but also where the performance of our approximate policy can be quantified in comparison to the optimal policy.

For our specific experiment, we picked $k = 200$ jobs and $d = 100$ queues, and randomly chose the transition probabilities over the uniform simplex. The vector of costs g was chosen to be $[.99, .98, .97, ..., .01, 0.0]$. The discount factor was chosen to be .996.

3.2.4 Approximating the Optimal Policy

Under exact fixed point calculation methods, we would be required to search for the fixed point of F over the space of all functions of the form $X \times U \to \Re$, where $X \times U$ is the set of all possible state-action pairs. However, the enormous size of $X \times U$ precludes such a computationally expensive method. Instead, we will use temporal difference learning and the fixed point Kalman filter to approximate the fixed point of F.

Our approximation was taken from the linear hull of 100 basis functions, one function for each queue. We enumerate the basis functions as $\phi^{(i)} : X \times U \to \Re, i = 1, \ldots, 100$. Let $x^{(i)}, i = 1, \ldots, 100$ denote the current length of queue i. We chose $\phi^{(i)}$ to be equal to the expected length of queue i at the next time step, given current length $x^{(i)}$ and current action u. The motivation was simply that the basis functions should depend on the queue lengths and the action chosen.

For $(u, i) \in \{1, \ldots, 100\}^2$, let p_{ui} be the $(u, i)th$ element of the transition matrix P, which gives the probability that if queue u is serviced, a job from queue u transitions to queue i. Let $I(x^{(u)})$ be the indicator function returning 1 if $x^{(u)}$ is non-zero. Let $I_u(i)$ be the indicator function returning 1 if $i = u$. Then,

$$\phi^{(i)}(x, u) = x^{(i)} + (p_{ui} - I_u(i))I(x^{(u)}).$$

3.2.5 Experimental Results

Two million iterations of TD(0) and the fixed point Kalman filter were run, producing two weight trajectories r_t. We print the formula for the fixed point Kalman filter for reference:

$$r_{t+1} \quad = \quad r_t + \gamma_t H_t \phi(x_t, u_t)((F\Phi r_t)(x_t, u_t) + w_{t+1} - (\Phi r_t)(x_t, u_t)),$$

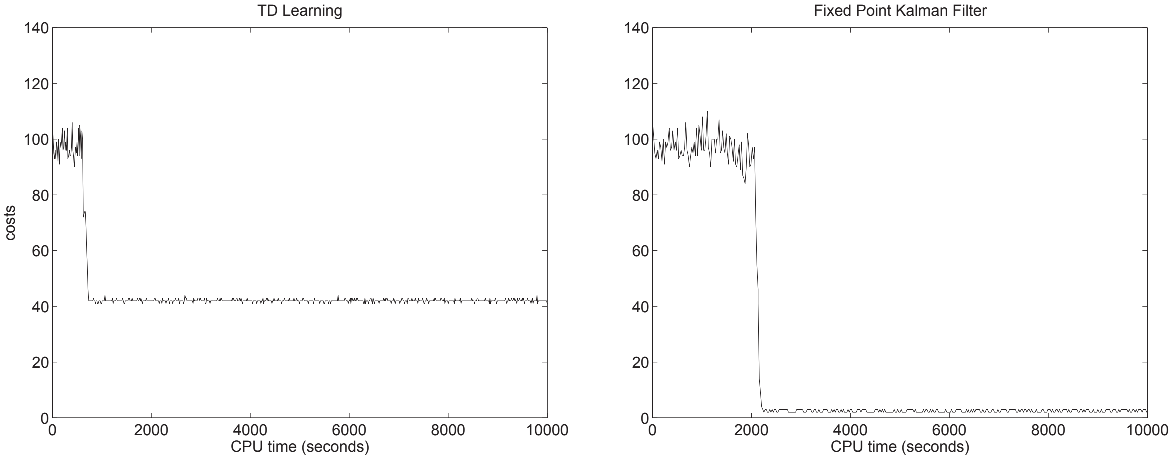

Figure 6: The cost of the states that were visited while training two policies for servicing a queueing network.

where

$$H_t = \left(\frac{1}{t} \sum_{s=1}^{t} \phi(x_s, u_s) \phi'(x_s, u_s) \right)^{\dagger},$$

the estimate $(F\Phi r_t)(x_t, u_t) + w_{t+1}$ is given by

$$(F\Phi r_t)(x_t) + w_{t+1} = \sum_{i=1}^{d} c(i) l(i) + \alpha \min_u ((\Phi r_t)(x_t, u)).$$

To speed up convergence, we used some random state exploration. With probability .0003, u_t was chosen randomly (with uniform probability) from all non-empty queues. Otherwise, u_t was given by

$$u_t = \operatorname*{argmin}_{u: x^{(u)} > 0} (\Phi r_t)(x_t, u).$$

At each iteration, the cost $c(x_t)$ was recorded. A range of constant stepsizes were tested $(10^{(0)}, 10^{-1}, \ldots, 10^{(-8)})$. For the fixed point Kalman filter, the cost $c(x_t)$ converges soonest with a stepsize of .01; for temporal-difference learning, $c(x_t)$ converged fastest with a stepsize of .00001.

Figures 6 plots the costs $c(x_t)$ incurred while training the sequence of weights r_t. As shown, the Kalman filter produces a policy achieving optimal costs, while TD(0) produces a non-optimal policy. For problems where the contraction is not with respect to a weighted Euclidean norm $\| \cdot \|_D$, the convergence theorems do not apply, so it is possible for the algorithms to converge to differing policies (or to not converge at all) We comment that the optimality of the Kalman filter policy was verifiable for this experiment only because in Klimovs problem, the optimal policy is well-known: all jobs except one should permanently reside in the cheapest queue.

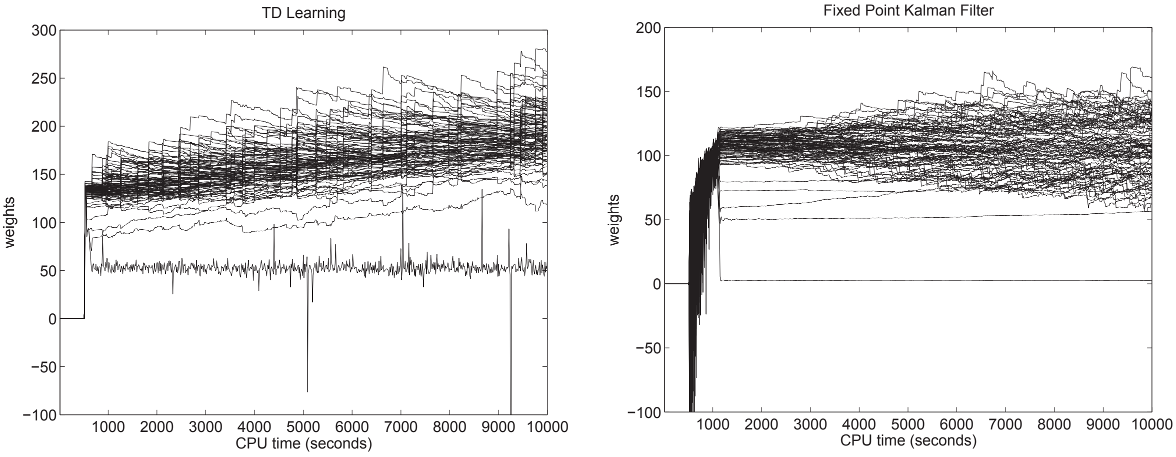

Figure 7: The feature weights of two evolving policies for servicing a queueing network.

Figure 7 shows the actual trajectory of each of the elements of r_t as they are adapted by temporal difference learning and the fixed point Kalman filter. The behavior of the weights appears to be quite complex.

For both the TD(0) and Kalman filter experiments, the weights were allowed to adapt for $2,000,000$ iterations. We note that the average CPU time elapsed per iteration of the fixed point Kalman filter was .0225 seconds. The average CPU time elapsed per TD(0) iterations was .0138 seconds (the total computations took roughly 12 hours for each algorithm and each choice of stepsize). However, the iterations could have been halted much sooner, after 2000 seconds, with no change in the policy. Note that in order to reduce the elapsed running time, we used a recursive rank-1 update formula to calculate H_t from H_{t1}:

$$\frac{H_t}{t} = \frac{H_{t-1}}{t-1} - \frac{H_{t-1}\phi(x_t)\phi' H_{t-1}}{(t-1)^2 + (t-1)\phi(x_t)H_{t-1}\phi(x_t)}$$

We used this update for both the options pricing and queueing system experiments.

3.2.6 A Smaller Queueing Problem

We also used the fixed point Kalman filter and TD(0) to produce approximate value functions for a much smaller queueing problem. The problem is the same as before, except that the number of jobs has been reduced from 100 to 10, with the number of features correspondingly reduced to ten also. In this case, both algorithms converge to value functions that give the optimal policy, in roughly the same time, as shown in Figures 8 and 9. The trajectory of the weights is especially interesting for the TD-learning algorithm in this example.

Our final comment is that these tests were run using MATLAB on a Sun Blade 1000 (the options pricing example) and a Sun ULTRA 60 workstation (the queueing system example).

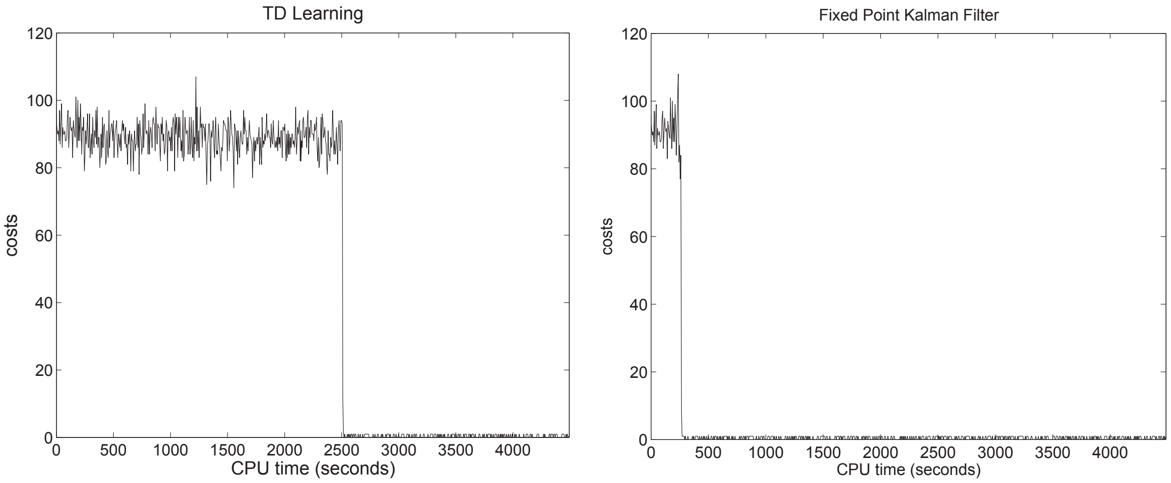

Figure 8: The cost of the states that were visited while training two policies for servicing a small queueing network.

4 Conclusion

The fixed point Kalman filter can be interpreted in several different ways. First, it can be viewed as a generalization of the Kalman filter suitable for approximating the fixed point of a D–norm. Second, the algorithm can also be thought of a variation on TD(0) in which basis functions are adaptively rescaled to be orthonormal for better performance. Finally, the algorithm can be viewed as a relaxation of LSTD that can be efficiently implemented when approximating the fixed point of a a non–linear operator.

We proved a convergence theorem for the fixed point Kalman filter, which parallels similar results for TD. We have also carried out some simulations, which thus far show that the fixed point Kalman filter outperforms TD(0) in problems where the behavior of both algorithms are well understood. The results also suggest that the fixed point Kalman filter is more robust than TD(0) against basis functions that are "ill–conditioned" (i.e., far from D-orthonormal).

After these cursory simulations, we suggest that in cases where the state space evolves with a stationary distribution π, the behavior of both TD and the fixed point Kalman filter are well understood, and the fixed point Kalman filter can be expected to outperform TD learning. In the more general case of a Markov decision process, the behavior of neither algorithm is well–understood, and it is not clear when the fixed point Kalman filter will actually converge faster than TD learning (if it should converge at all). However, the general idea behind the fixed point Kalman filter – that adaptive algorithms should account for the scaling and orthogonality of their basis functions – is a practical one, as demonstrated by the poor behavior of TD with the addition of a single ill-chosen basis function. In cases where near-optimal policies (involving large numbers of basis functions) are sought for complex problems, we may expect significant advantages from the fixed point Kalman filter.

In the broader literature on TD, there exist many application areas where TD

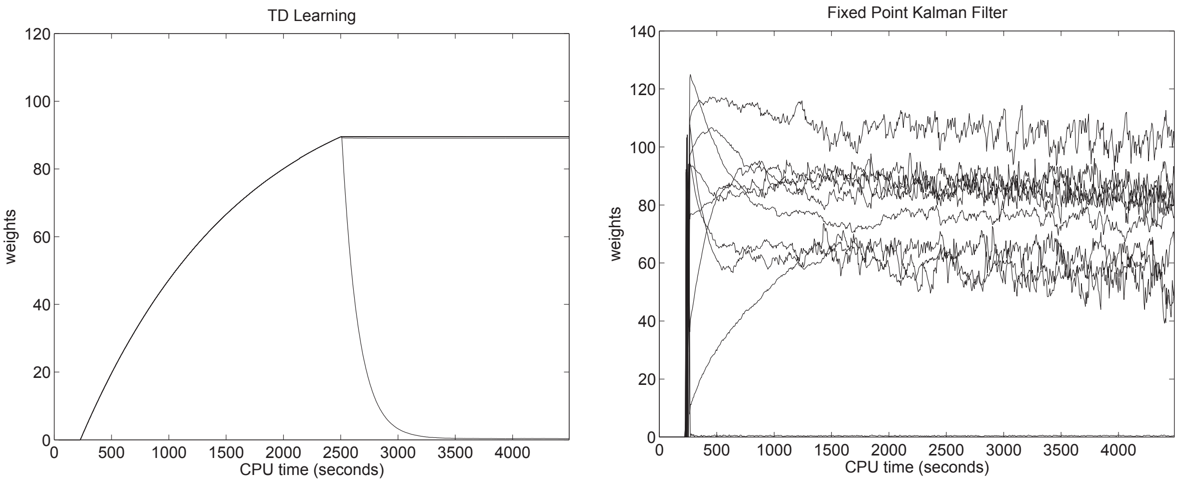

Figure 9: The feature weights of two evolving policies for servicing a queueing network.

and the fixed point Kalman filter can be applied, but only without any theoretical rationale for their convergence. However, TD has already been tested on many such problems with good experimental results. These problems include a world class computer backgammon player [22], channel allocation for cellular networks [6], job shop scheduling [30], and inventory control [25]. On the other hand, problems have also been constructed for which TD has been shown not to converge [14]. We expect that similar claims, both positive and negative, will hold for the fixed point Kalman filter.

A Proof of Theorem 1

In this appendix, we prove Theorem 1, which asserts that θ_t converges to 0, with probability one. Without loss of generality, we will as an intermediate step prove the convergence of θ_t using one additional assumption about the sequence Γ_t. In particular, we will first prove the convergence of θ_t under the working assumption that there exists a (deterministic) scalar $\overline{\Gamma}$ such that $\sup_t \|\Gamma_t\|_D \leq \overline{\Gamma}$, with probability 1. The majority of this appendix is devoted to this proof, showing that for any $\overline{\Gamma}$, θ_t converges to 0 with probability 1.

While this additional assumption will greatly facilitate the analysis, we note that it does not actually hold for the fixed point Kalman filter. Therefore, the assumption is lifted using the following argument: Since Γ_t converges to I (Assumption 3), for any scalar $\overline{\Gamma}$ the probability $\epsilon_{\overline{\Gamma}}$ that $\sup_t \|\Gamma_t\| \geq \overline{\Gamma}$ goes to 0 as $\overline{\Gamma} \to \infty$.[4] Given a scalar $\overline{\Gamma}$ it is easy to construct a $\mathcal{F}_t$–measurable process $\tilde{\Gamma}_t$ such that $\tilde{\Gamma}_t = \Gamma_t$, for all t, for sample paths satisfying $\sup_t \|\Gamma_t\|_D \leq \overline{\Gamma}$. Proving that θ_t converges to 0 for such processes establishes that θ_t converges to 0 for a set of sequences Γ_t, generated by the original

[4] If $\epsilon_{\overline{\Gamma}}$ converges to a positive number as $\overline{\Gamma} \to \infty$, then with some nonzero probability, $\sup_t \|\Gamma_t\|_D = \infty$ with that probability. However, $\sup_t \|\Gamma_t\|_D$ must be less than ∞ with probability one, by Assumption 3. As a result, $\epsilon_{\overline{\Gamma}}$ must converge to zero.

process, of probability $1 - \epsilon_{\overline{\Gamma}}$. Since $\epsilon_{\overline{\Gamma}}$ can be set arbitrarily close to 0, such a proof establishes convergence of θ_t to 0 with probability one.

We will use the following variant of the Supermartingale Convergence Theorem, which can be found in [3] (Lemma 2 on page 344):

Theorem 3 *Let Z_i ,A_i, B_i, and Y_i be finite nonnegative random variables adapted to an increasing sequence of σ-fields $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots$ such that $\sum A_i < \infty$ and $\sum B_i < \infty$ with probability one. If*

$$\mathrm{E}[Z_{i+1}|\mathcal{G}_i] \leq (1 + A_i)Z_i + B_i - Y_i, \tag{5}$$

for all i, then

$$\sum Y_i < \infty \quad \text{and} \quad Z_i \to Z < \infty,$$

for some scalar $Z < \infty$, with probability 1.

We will associate various terms in our stochastic approximation algorithm with variables involved in Theorem 3, as we now explain. First, let $\overline{x} \in S$ be a distinguished state (arbitrarily chosen). Let t_i be the time of the ith visit to state $\overline{x}$. Let $\mathcal{G}_i$ be the σ–field generated by information available up to time t_i. Let $Z_i = \|\theta_{t_i}\|_D$. We denote the difference between $\theta_{t_{i+1}}$ and θ_{t_i} by

$$s_{i+1} = \sum_{t=t_i}^{t_{i+1}-1} \gamma_t \Gamma_t h(\theta_t, x_t, x_{t+1}).$$

We also define an approximation to s_{i+1}:

$$\tilde{s}_{i+1} = \sum_{t=t_i}^{t_{i+1}-1} \gamma_{t_i} \Gamma_{t_i} h(\theta_{t_i}, x_t, x_{t+1}).$$

Note that s_i and $\tilde{s}_i$ are $\mathcal{G}_i$–measurable. In the next subsection, we will prove the following lemma:

Lemma 1 *There exists a scalar $c > 0$ and a (random) index i^* such that*

$$\theta'_{t_i} D E[\tilde{s}_{i+1}|\mathcal{G}_i] \leq -c\gamma_{t_i} \|\theta_{t_i}\|_D^2,$$

for all $i \geq i^*$.

Let

$$Y_i = -\left[2\theta'_{t_i} D E[\tilde{s}_{i+1}|\mathcal{G}_i]\right]^- .$$

Note that Y_i is nonnegative. We will also later prove the following lemma:

Lemma 2 *There exist finite nonnegative random variables A_i and B_i, adapted to $\mathcal{G}_i$, such that $\sum A_i < \infty$, $\sum B_i < \infty$, and*

$$A_i Z_i + B_i \geq 2\theta'_{t_i} D E[s_{i+1} - \tilde{s}_{i+1}|\mathcal{G}_i] + \left[2\theta'_{t_i} D E[\tilde{s}_{i+1}|\mathcal{G}_i]\right]^+ + E[s_{i+1}^2|\mathcal{G}_i],$$

with probability one.

Hence, choosing A_i and B_i to be variables identified by this lemma, Equation 5 is satisfied by our definitions:

$$\begin{aligned}
E[Z_{i+1}|\mathcal{G}_i] &= E\left[\|\theta_{t_{i+1}}\|_D^2|\mathcal{G}_i\right] \\
&= E\left[\|\theta_{t_i} + s_{i+1}\|_D^2|\mathcal{G}_i\right] \\
&= E\left[\|\theta_{t_i}\|_D^2 + 2\theta'_{t_i} D s_{i+1} + \|s_{i+1}\|_D^2|\mathcal{G}_i\right] \\
&= E\left[\|\theta_{t_i}\|_D^2 + 2\theta'_{t_i} D \tilde{s}_{i+1} + 2\theta'_{t_i} D(s_{i+1} - \tilde{s}_{i+1}) + \|s_{i+1}\|_D^2|\mathcal{G}_i\right] \\
&\leq (1 + A_i) Z_i + B_i - Y_i.
\end{aligned}$$

Since conditions of Theorem 3 are satisfied, we have $\sum Y_i < \infty$ and $Z_i \to Z < \infty$, for some scalar Z. Lemma 1 implies that there exists a (random) index i^* such that $Y_i \geq c\gamma_{t_i} Z_i$, for all $i \geq i^*$. Together with the fact that $\sum Y_i < \infty$ and $\sum \gamma_{t_i} = \infty$, this implies that $Z = 0$. In other words, the subsequence $\theta_{t_i}, i = 1, \ldots$ converges to 0. Furthermore, since the distinguished state was arbitrary, convergence of the subsequence holds for any choice of $\overline{x}$. Since there are a finite number of states in S, it follows that the entire sequence θ_t converges to 0.

We have just proven Theorem 1. Our proof not quite self-contained, though – it made use of Lemmas 1 and 2, which we will prove in the following subsections.

A.1 Proof of Lemma 1

By Assumption 5, there exists a scalar $c_1 > 0$ such that $\theta' D\overline{h}(\theta) \leq -c_1\|\theta\|_D^2$ for all θ. Furthermore, by Assumptions 5 and 4, there exists a scalar c_2 such that $\|\overline{h}(\theta)\|_D \leq c_2\|\theta\|_D$. Hence, we have

$$\begin{aligned}
\theta'_{t_i} D E[\tilde{s}_{i+1}|\mathcal{G}_i] &= \theta'_{t_i} D E\left[\sum_{t=t_i}^{t_{i+1}-1} \gamma_{t_i} \Gamma_{t_i} h(\theta_{t_i}, x_t, x_{t+1})|\mathcal{G}_i\right] \\
&= \gamma_{t_i} E[t_{i+1} - t_i] \theta'_{t_i} D \Gamma_{t_i} \overline{h}(\theta_{t_i}) \\
&= \gamma_{t_i} E[t_{i+1} - t_i] \left(\theta'_{t_i} D\overline{h}(\theta_{t_i}) + \theta'_{t_i} D(\Gamma_{t_i} - I)\overline{h}(\theta_{t_i})\right) \\
&\leq \gamma_{t_i} E[t_{i+1} - t_i] \left(-c_1\|\theta_{t_i}\|_D^2 + c_2\|\Gamma_{t_i} - I\|_D \|\theta_{t_i}\|_D^2\right).
\end{aligned}$$

The lemma follows from the fact that $\|\Gamma_{t_i} - I\|_D$ converges to 0.

A.2 Proof of Lemma 2

Recall our working assumption that there is a (deterministic) scalar $\overline{\Gamma}$ such that $\sup_t \|\Gamma_t\|_D \leq \overline{\Gamma}$ with probability one. We begin with a helpful lemma that places a bound on how quickly θ_t can change over time.

Lemma 3 *There exists a scalar C such that*

$$\|\theta_{t+\tau} - \theta_t\|_D \leq \tau\gamma_t\overline{\Gamma} \prod_{j=1}^{\tau-1} (1 + \gamma_{t+j}\overline{\Gamma}C)(1 + \|\theta_t\|_D),$$

for all t and $\tau \geq 1$.

Proof: By Assumption 4, there exists a scalar C such that

$$\|h(\theta, x, y)\|_D \leq C(1 + \|\theta\|_D).$$

Let $\Delta_{t,\tau} = \|\theta_{t+\tau} - \theta_t\|_D$ and $a_{t,\tau} = \gamma_{t+\tau}\overline{\Gamma}C$. We then have $\Delta_{t,0} = 0$ and

$$\begin{aligned}
\Delta_{t,\tau+1} &\leq \Delta_{t,\tau} + \|\theta_{t+\tau+1} - \theta_{t+\tau}\|_D \\
&\leq \Delta_{t,\tau} + \gamma_{t+\tau}\|\Gamma_{t+\tau}\|_D C(1 + \|\theta_{t+\tau}\|_D) \\
&\leq \Delta_{t,\tau} + \gamma_{t+\tau}\overline{\Gamma}C(1 + \|\theta_t\|_D + \Delta_{t,\tau}) \\
&= (1 + a_{t,\tau})\Delta_{t,\tau} + a_{t,\tau}(1 + \|\theta_t\|_D).
\end{aligned}$$

It follows that

$$\begin{aligned}
\Delta_{t,\tau} &\leq \sum_{j=1}^{\tau}\prod_{k=1}^{j-1}(1 + a_{t,\tau-k})a_{t,\tau-j}(1 + \|\theta_t\|_D) \\
&\leq \tau\gamma_t\overline{\Gamma}\prod_{j=1}^{\tau-1}(1 + a_{t,j})(1 + \|\theta_t\|_D).
\end{aligned}$$

qed

According to Lemma 3, it follows that $\|\theta_t\|_D$ can only increase by a factor of $(1 + \gamma_{t+j}\overline{\Gamma}C)$ (modulo the linear term) at each time step. Note that this factor approaches 1 as t grows. One implication of this and the assumption that $\sum \gamma_t^2 < \infty$ is that the growth of $\|\theta_t\|_D$ is sub-exponential. This is articulated in the following corollary, which we state without proof (the proof is straightforward).

Corollary 1 *For any nonnegative scalars a, p, and q, there exists a scalar c such that*

$$\sum_{\tau=0}^{\infty} e^{-a\tau}\tau^p\|\theta_{t+\tau} - \theta_t\|_D^q \leq \gamma_t^q c(1 + \|\theta_t\|_D^q),$$

for all t.

Recall that our goal is to establish existence of finite nonnegative random variables A_i and B_i, adapted to $\mathcal{G}_i$, such that $\sum A_i < \infty$, $\sum B_i < \infty$, and

$$A_i Z_i + B_i \geq 2\theta_{t_i}' D\mathrm{E}[s_{i+1} - \tilde{s}_{i+1}|\mathcal{G}_i] + \left[2\theta_{t_i}' D\mathrm{E}[\tilde{s}_{i+1}|\mathcal{G}_i]\right]^+ + E[s_{i+1}^2|\mathcal{G}_i],$$

with probability one. For shorthand, let

$$\begin{aligned}
\delta_i &= 2\theta_{t_i}' D\mathrm{E}[s_{i+1} - \tilde{s}_{i+1}|\mathcal{G}_i] + \left[2\theta_{t_i}' D\mathrm{E}[\tilde{s}_{i+1}|\mathcal{G}_i]\right]^+ + \mathrm{E}[s_{i+1}^2|\mathcal{G}_i], \\
\delta_i^{(1)} &= \mathrm{E}\left[\sum_{t=t_i}^{t_{i+1}-1}\gamma_t\Gamma_t\left(h(\theta_t, x_t, x_{t+1}) - h(\theta_{t_i}, x_t, x_{t+1})\right)|\mathcal{G}_i\right], \\
\delta_i^{(2)} &= \mathrm{E}\left[\sum_{t=t_i}^{t_{i+1}-1}\gamma_{t_i}(\Gamma_t - \Gamma_{t_i})h(\theta_{t_i}, x_t, x_{t+1})|\mathcal{G}_i\right], \\
\delta_i^{(3)} &= \mathrm{E}\left[\sum_{t=t_i}^{t_{i+1}-1}(\gamma_t - \gamma_{t_i})\Gamma_t h(\theta_{t_i}, x_t, x_{t+1})|\mathcal{G}_i\right], \\
\delta_i^{(4)} &= \left[2\theta_{t_i}' D\mathrm{E}[\tilde{s}_{i+1}|\mathcal{G}_i]\right]^+, \\
\delta_i^{(5)} &= \mathrm{E}[s_{i+1}^2|\mathcal{G}_i],
\end{aligned}$$

and note that

$$\delta_i = \sum_{k=1}^{5} \delta_i^{(k)}.$$

For each $\delta^{(k)}$, we will prove a lemma that there exist sequences $A_i^{(k)}$ and $B_i^{(k)}$ such that $\sum A_i^{(k)} < \infty$, $\sum B_i^{(k)} < \infty$, and $|\delta_i^{(k)}| \leq A_i^{(k)} \|\theta_{t_i}\|_D + B_i^{(k)}$. Letting $A_i = \sum_{k=1}^{5} A_i^{(k)}$ and $B_i = \sum_{k=1}^{5} B_i^{(k)}$, it is easy to see that Lemma 2 follows from these lemmas, which we state and prove in the remainder of this section.

Lemma 4 *There exist sequences $A_i^{(1)}$ and $B_i^{(1)}$ such that $\sum A_i^{(1)} < \infty$, $\sum B_i^{(1)} < \infty$, and $|\delta_i^{(1)}| \leq A_i^{(1)} \|\theta_{t_i}\|_D + B_i^{(1)}$, for all i.*

Proof: By Assumption 4, there exists a scalar c_1 such that $\|h(\theta, x, y) - h(\overline{\theta}, x, y)\|_D \leq c_1 \|\theta - \overline{\theta}\|_D$, for all x, y. Hence,

$$\begin{aligned} |\delta_i^{(1)}| &= \left| E\left[\sum_{t=t_i}^{t_{i+1}-1} \gamma_t \Gamma_t \left(h(\theta_t, x_t, x_{t+1}) - h(\theta_{t_i}, x_t, x_{t+1}) \right) |\mathcal{G}_i \right] \right| \\ &\leq \gamma_{t_i} \overline{\Gamma} c_1 E\left[\sum_{t=t_i}^{t_{i+1}-1} \|\theta_t - \theta_{t_i}\|_D |\mathcal{G}_i \right]. \end{aligned}$$

Since $\Pr\{t_{i+1} - t_i \geq \tau\} \leq a e^{-b\tau}$ for some $a, b \geq 0$, Corollary 1 implies that there exists a scalar c_2 such that

$$E\left[\sum_{t=t_i}^{t_{i+1}-1} \|\theta_t - \theta_{t_i}\|_D |\mathcal{G}_i \right] \leq \gamma_{t_i} c_2 (1 + \|\theta_{t_i}\|_D),$$

for all i. Letting $A_i^{(1)} = B_i^{(1)} = \gamma_{t_i}^2 \overline{\Gamma} c_1 c_2$, the result follows from the fact that $\sum \gamma_{t_i}^2 < \infty$. **qed**

Lemma 5 *There exist sequences $A_i^{(2)}$ and $B_i^{(2)}$ such that $\sum A_i^{(2)} < \infty$, $\sum B_i^{(2)} < \infty$, and $|\delta_i^{(2)}| \leq A_i^{(2)} \|\theta_{t_i}\|_D + B_i^{(2)}$, for all i.*

Proof: By Assumption 4, there exists a scalar c such that $\|h(\theta, x, y)\|_D \leq c(1 + \|\theta\|_D)$ for all θ, x, y. Furthermore, by Assumption 3, there is a deterministic nonincreasing sequence β_i and a scalar $c_{\overline{\Gamma}}$ such that

$$\|\Gamma_{t-1} - \Gamma_t\|_D \leq \beta_t c_{\overline{\Gamma}},$$

for all t. We therefore have

$$\begin{aligned} |\delta_i^{(2)}| &= \left| E\left[\sum_{t=t_i}^{t_{i+1}-1} \gamma_{t_i} (\Gamma_t - \Gamma_{t_i}) h(\theta_{t_i}, x_t, x_{t+1}) |\mathcal{G}_i \right] \right| \\ &\leq E\left[\sum_{t=t_i}^{t_{i+1}-1} \gamma_{t_i} \|\Gamma_t - \Gamma_{t_i}\|_D |\mathcal{G}_i \right] c(1 + \|\theta_{t_i}\|_D) \end{aligned}$$

$$
\begin{aligned}
&\le\ E\left[(t_{i+1}-t_i)\sum_{t=t_i}^{t_{i+1}-1}\gamma_{t_i}\|\Gamma_{t-1}-\Gamma_t\|_D|\mathcal{G}_i\right]c(1+\|\theta_{t_i}\|_D)\\
&\le\ E\left[(t_{i+1}-t_i)\sum_{t=t_i}^{t_{i+1}-1}\gamma_{t_i}\beta_t c_{\overline{\Gamma}}|\mathcal{G}_i\right]c(1+\|\theta_{t_i}\|_D)\\
&\le\ E\left[(t_{i+1}-t_i)^2|\mathcal{G}_i\right]\gamma_{t_i}\beta_{t_i}c_{\overline{\Gamma}}c(1+\|\theta_{t_i}\|_D).
\end{aligned}
$$

By Assumption 3, $\sum \gamma_{t_i}\beta_{t_i} < \infty$. The result follows. **qed**

Lemma 6 *There exist sequences $A_i^{(3)}$ and $B_i^{(3)}$ such that $\sum A_i^{(3)} < \infty$, $\sum B_i^{(3)} < \infty$, and $|\delta_i^{(3)}| \le A_i^{(3)}\|\theta_{t_i}\|_D + B_i^{(3)}$, for all i.*

Proof: By Assumption 4, there exists a scalar c such that $\|h(\theta,x,y)\|_D \le c(1+\|\theta\|_D)$ for all θ, x, y.

$$
|\delta_i^{(3)}| \le E\left[\sum_{t=t_i}^{t_{i+1}-1}(\gamma_{t_i}-\gamma_t)\Gamma_t h(\theta_{t_i},x_t,x_{t+1})|\mathcal{G}_i\right] \le E\left[\sum_{t=t_i}^{t_{i+1}-1}(\gamma_{t_i}-\gamma_t)|\mathcal{G}_i\right]\overline{\Gamma}c(1+\|\theta_{t_i}\|_D).
$$

Let

$$
A_i^{(3)} = B_i^{(3)} = E\left[\sum_{t=t_i}^{t_{i+1}-1}(\gamma_{t_i}-\gamma_t)|\mathcal{G}_i\right]\overline{\Gamma}c.
$$

Let $\tau = t_2 - t_1$. Recalling that γ_t is a decreasing sequence, converging to zero, we have

$$
\begin{aligned}
\sum_{i=1}^{\infty}A_i^{(3)} &= \sum_{i=1}^{\infty}E\left[\sum_{t=t_i}^{t_{i+1}-1}(\gamma_{t_i}-\gamma_t)|\mathcal{G}_i\right]\\
&\le \sum_{t=0}^{\infty}E\left[\sum_{\overline{\tau}=0}^{\tau-1}(\gamma_t-\gamma_{t+\overline{\tau}})\right]\\
&= E\left[\sum_{\overline{\tau}=0}^{\tau-1}\sum_{t=0}^{\infty}(\gamma_t-\gamma_{t+\overline{\tau}})\right]\\
&= E\left[\sum_{\overline{\tau}=0}^{\tau-1}\sum_{t=0}^{\infty}\sum_{k=0}^{\overline{\tau}-1}(\gamma_{t+k}-\gamma_{t+k+1})\right]\\
&= E\left[\sum_{\overline{\tau}=0}^{\tau-1}\sum_{k=0}^{\overline{\tau}-1}\sum_{t=0}^{\infty}(\gamma_{t+k}-\gamma_{t+k+1})\right]\\
&\le E\left[\sum_{\overline{\tau}=0}^{\tau-1}\sum_{k=0}^{\overline{\tau}-1}\sum_{t=0}^{\infty}(\gamma_t-\gamma_{t+1})\right]\\
&= E\left[\sum_{\overline{\tau}=0}^{\tau-1}\overline{\tau}\gamma_0\right]\\
&\le \gamma_0 E[\tau^2].
\end{aligned}
$$

The result follows. **qed**

Lemma 7 *There exists a sequence $B_i^{(4)}$ such that $\sum B_i^{(4)} < \infty$ and $|\delta_i^{(4)}| \leq B_i^{(4)}$, for all i.*

Proof: Note that $\delta_i^{(4)} \geq 0$. Lemma 1 implies that $\delta_i^{(4)}$ is positive only a finite number of times. Hence, $\sum \delta_i^{(4)} < \infty$. Letting $B_i^{(4)} = \delta_i^{(4)}$, the result follows. **qed**

Lemma 8 *There exist sequences $A_i^{(5)}$ and $B_i^{(5)}$ such that $\sum A_i^{(5)} < \infty$, $\sum B_i^{(5)} < \infty$, and $|\delta_i^{(5)}| \leq A_i^{(5)} \|\theta_{t_i}\|_D + B_i^{(5)}$, for all i.*

Proof: By Assumption 4, there exists a scalar c_1 such that $\|h(\theta, x, y)\|_D^2 \leq c_1(1+\|\theta\|_D^2)$, for all θ, x, y. Hence,

$$\begin{aligned}
\left|\delta_i^{(5)}\right| &= E\left[\|s_{i+1}\|_D^2 | \mathcal{G}_i\right] \\
&= E\left[\left\|\sum_{t=t_i}^{t_{i+1}-1} \gamma_t \Gamma_t h(\theta_t, x_t, x_{t+1})\right\|_D^2 | \mathcal{G}_i\right] \\
&\leq \gamma_{t_i}^2 \overline{\Gamma}^2 E\left[\sum_{t=t_i}^{t_{i+1}-1} \|h(\theta_t, x_t, x_{t+1})\|_D^2 | \mathcal{G}_i\right] \\
&\leq \gamma_{t_i}^2 \overline{\Gamma}^2 E\left[\sum_{t=t_i}^{t_{i+1}-1} c_1(1 + \|\theta_t\|_D^2) | \mathcal{G}_i\right].
\end{aligned}$$

Since $\Pr\{t_{i+1} - t_i \geq \tau\} \leq ae^{-b\tau}$ for some $a, b \geq 0$, Corollary 1 implies that there exists a scalar c_2 such that

$$\left|\delta_i^{(5)}\right| \leq \gamma_{t_i}^2 \overline{\Gamma}^2 c_2(1 + \|\theta_{t_i}\|_D^2),$$

for all i. The result follows from the fact that $\sum \gamma_{t_i}^2 < \infty$. **qed**

References

[1] Bellman, R. and Dreyfuss, S. 1959. "Functional Approximations and Dynamic Programming", *Mathematical Tables and Other Aids to Computation*, Vol. 13, pp 247-251.

[2] Barto, A. and Crites, R. 1996. "Improving Elevator Performance Using Reinforcement Learning", *Advances in Neural Information Processing Systems*, Vol. 8, pp. 1017-1023.

[3] Benveniste, A., Métivier, M. and Priouret, P. 1991. *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, New York.

[4] Bertsekas, D. P.. 1995. *Nonlinear Programming*, Athena Scientific.

[5] Bertsekas, D. P.. 1995. *Dynamic Programming and Optimal Control*, Athena Scientific.

[6] Bertsekas, D.P., and Singh, S. 1997. "Reinforcement Learning for Dynamic Channel Allocation in Cellular Telephone Systems", *Advances in Neural Information Processing Systems*, Vol. 9, The MIT Press, p. 974.

[7] Bertsekas, D. P. and Tsitsiklis, J.N. 1995. *Neuro-Dynamic Programming*, Athena Scientific.

[8] V. Borkar. 1995. *Probability Theory: An Advanced Course*, Springer-Verlag, New York.

[9] J. Boyan. 2002. "Technical Update: Least-squares Temporal Difference Learning," *Machine Learning*, Vol 49(2), pp. 233-246.

[10] J. Boyan. 1999. "Least-squares Temporal Difference Learning",Proceedings of the Sixteenth International Conference (ICML) on Machine Learning.

[11] S.J. Bradtke and A.G. Barto. 1996. "Linear least-squares algorithms for temporal-difference learning." *Machine Learning*, Vol. 22, pp. 33-57.

[12] Choi, D. S. and Van Roy, B. 2001. "A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning," Proceedings of the International Joint Conference on Machine Learning.

[13] Dayan, P.D. 1992. "The Convergence of TD(λ) for General λ," *Machine Learning*, Vol. 8, pp. 341-362.

[14] de Farias, D.P. and Van Roy, B. 2000. " On the Existence of Fixed Points for Approximate Value Iteration and Temporal-Difference Learning," *Journal of Optimization Theory and Applications*, Vol. 105, No. 3.

[15] Gurvits, L., Lin, L.J. and Hanson, S.J. 1994. "Incremental Learning of Evaluation Functions for Absorbing Markov Chains: New Methods and Theorems," preprint.

[16] Karatzas, I., and Shreve, S.E. 1998. *Methods of Mathematical Finance*, Springer-Verlang.

[17] Lagoudakis, M. and Parr, R. 2001. "Model-Free Least-Squares Policy Iteration," *Neural Information Processing Systems (NIPS-14).*

[18] Nedic, A. and Bertsekas, D. P. 2001. "Policy Evaluation Algorithms with Linear Function Approximation," Technical Report LIDS-P-2537, MIT Laboratory for Information and Decision Systems.

[19] Pineda, F. 1997. "Mean-Field Analysis for Batched TD(λ)," *Neural Computation* pp. 1403–1419.

[20] Sutton, R.S. 1988. "Learning to Predict by the Method of Temporal Differences," *Machine Learning*, Vol. 3, pp. 9-44.

[21] Tadić, V. 2001. "On the Convergence of Temporal-Difference Learning with Linear Function Approximation," *Machine Learning*, Vol. 42, pp. 241-267.

[22] Tesauro, G. 1995. "Temporal Difference Learning and TD-Gammon", *Communications of the ACM*, Vol. 38, No. 3.

[23] Tsitsiklis, J.N. and Van Roy, B. 1997. "An Analysis of Temporal-Difference Learning with Function Approximation," *IEEE Transactions on Automatic Control*, Vol. 42, pp 674-690.

[24] J. N. Tsitsiklis and B. Van Roy. 1999. Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing High-Dimensional Financial Derivatives. *IEEE Transactions on Automatic Control*, Vol. 44, No. 10, pp. 1840-1851.

[25] Van Roy, B., Bertsekas, D.P., Lee, Y. and Tsitsiklis, J.N. 1997. “A Neuro-Dynamic Programming Approach to Retailer Inventory Management,” Proceedings of the IEEE Conference on Decision and Control.

[26] Van Roy, B. 1998. *Learning and Value Function Approximation in Complex Decision Processes*, Ph.D. dissertation, MIT.

[27] Varaiya, P., Walrand, J., and Buyokkoc, C. 1985. ”Extensions of the Multiarmed Bandit Problem: The Discounted Case.” *IEEE Transactions on Automatic Control*, Vol 30, No. 5.

[28] Warmuth, M. and Schapire, R. 1997. “On the Worst-case Analysis of Temporal-difference Learning Algorithms,” *Journal of Machine Learning*, Vol. 22 (1,2,3), pp. 95-121.

[29] Warmuth, M. and Forster, J. “Relative Loss Bounds for Temporal-Difference Learning,” to appear in *Journal of Machine Learning.*

[30] Zhang, W. and Dietterich, T.G. 1995. “A reinforcement learning approach to Job-shop Scheduling”, Proceedings of the International Joint Conference on Artificial Intelligence.