Parametric Policy Iteration:
An Efficient Algorithm for Solving Multidimensional DP Problems?

John Rust, *Yale University*

April, 2000

1. Introduction

This note provides a brief description of an algorithm I call *parametric policy iteration,* and the results of some computational experiments with the method that suggest it could be a promising method for breaking the curse of dimensionality of dynamic programming in problems where the value function belongs to a "nice" space.

2. Review of the DP Problem

Consider an infinite horizon dynamic programming problem where the state $s \in S \subset R^d$. Bellman's equation is

$$V(s) = \max_{a \in A(s)} [u(s,a) + \beta \int V(s')p(s'|s,a)ds']. \tag{2.1}$$

The *optimal policy* $\alpha(s)$ is the solution to:

$$\alpha(s) = \underset{a \in A(s)}{argmax}[u(s,a) + \beta \int V(s')p(s'|s,a)ds']. \tag{2.2}$$

In abstract terms $V = \Gamma(V)$ is the unique fixed point to the *Bellman operator* $\Gamma : B \to B$, where B is a Banach space of functions from S to R and the Bellman operator is given by

$$\Gamma\Big(V\Big)(s) \equiv \max_{a \in A(s)} [u(s,a) + \beta \int V(s')p(s'|s,a)ds']. \tag{2.3}$$

Most previous work has focused on proving approximation theorems based on some type of *Discretized Bellman operator:*

$$\Gamma_N\Big(V\Big)(s) \equiv \max_{a \in A(s)} \left[u(s,a) + \beta \sum_{i=1}^{N} V(s_i)p_N(s_i|s,a)\right], \tag{2.4}$$

where: p_N is a discrete probability distribution over a finite *grid* $\{s_1, \ldots, s_N\}$ in S, where $S = [0,1]^d$ for simplicity. In that case Γ_N has a dual interpretation, it can be regarded as a contraction mapping on R^N, (this is where the computation is done, resulting in a fixed point $V_N \in R^N$), but it is also a valid contraction mapping $\Gamma_N : B \to B$. This latter feature makes it easy to prove approximation bounds since the function $\Gamma(V_N)$ can be regarded as an element of B, and thus is a natural candidate as an approximation to $V = Gamma(V)$.

This note considers an alternative way of approximating V, namely as a linear combination of a set of *basis functions* $\{\rho_1(s), \rho_2(s), \ldots, \rho_k(s)\}$. These functions may not literally be a basis for B, but should have the property that this collection is *ultimately dense* in B in the sense that for any $V \in B$ we have:

$$\underset{k\to\infty}{lim} \; \underset{\theta_1,\ldots,\theta_k}{inf} \; \underset{s \in S}{sup} |V(s) - \sum_{i=1}^{k} \theta_i \rho_i(s)| = 0.$$

It is possible that families of functions that are nonlinear in the parameters θ could be considered also, such as neural net and wavelet bases. In this note I restrict attention to bases which are linear in parameters for simplicity, since as we will see below it vastly simplifies the problem of determining the optimal values of θ: they will be the solution to a simple least squares problem which is easy to compute. If the basis is a nonlinear function of θ then we will have to solve a nonlinear least squares problem, which could be more time consuming and it may be difficult or impossible to prove that the algorithm breaks the curse of dimensionality (Barron's results on the properties of neural nets as a means of breaking the curse of dimensionality of approximating certain classes of functions notwithstanding, there is the computational problem of finding a globally minimizing θ vector and this is where a curse of dimensionality could arise).

3. Policy Iteration for Continuous and Discrete MDPs

To understand the PPI algorithm I first review the infinite dimensional version of policy iteration. Puterman and Shin (1978) proved the convergence of this algorithm, showing that it is basically equivalent to an infinite-dimensional version of Newton's method for solving the nonlinear equation

$$(I - \Gamma)(V) = 0.$$

The algorithm consists of alternating *policy valuation* and *policy improvment* steps.

Policy Iteration (Infinite-Dimensional Version)

1. **Policy Valuation Step:** given an initial guess of policy α compute V_α, the value function implied by policy α:

$$V_\alpha(s) = u(s, \alpha(s)) + \beta E_\alpha V_\alpha(s),$$

where E_α is the *Markov operator* corresponding to α:

$$E_\alpha V(s) = \int V(s')p(s'|s, \alpha(s))ds'.$$

There is a unique solution to the linear operator equation defining V_α (Fredholm integral equation of the second kind):

$$V_\alpha = u_\alpha + \beta E_\alpha V = (I - \beta E_\alpha)^{-1} u_\alpha,$$

where $(I - \beta E_\alpha)$ exists and has the following geometric series or *Neumann series* representation:

$$(I - \beta E_\alpha)^{-1} = \sum_{t=0}^{\infty} [\beta E_\alpha]^t.$$

2. **Policy Improvement Step:** Compute improved policy α' using V_α:

$$\alpha'(s) = \underset{a \in A(s)}{argmax}[u(s, a) + \beta \int V_\alpha(s')p(s'|s, a)ds'].$$

If the Policy iteration algorithm converges, it is easy to see that the policy α^* that it converges to, and the corresponding value function V_{α^*} are solutions to Bellman's equation, and thus are the solution to the DP problem. It is well known that policy iteration always converges in a finite number of steps from any starting point if the state space S and action sets $A(s)$, $s \in S$ are finite sets. Puterman and Shin provided sufficient conditions for policy iteration to converge when S and $A(s)$ contain a continuum of points.

One strategy for approximating the solutions to continuous state DP problems is via discretation that results on an approximate MDP problem on a finite state space $S_N = \{s_1, \ldots, s_N\}$, and the use of policy iteration for the finite MDP on S_N. This results in

Policy Iteration (Finite-Dimensional Version)

1. **Policy Valuation Step:** given an initial guess of policy α compute $V_{\alpha,N}$, the value function (in R^N) implied by policy α:

$$V_{\alpha,N}(s) = u(s, \alpha(s)) + \beta E_{\alpha,N} V_{\alpha,N}(s),$$

where $E_{\alpha,N}$ is the *discrete Markov operator* corresponding to α:

$$E_{\alpha,N}V(s) = \sum_{i=1}^{N} V(s_i)p_N(s_i|s,\alpha(s)).$$

There is a unique solution to the linear system of equations defining $V_{\alpha,N} \in R^N$

$$V_{\alpha,N} = u_\alpha + \beta E_{\alpha,N}V_{\alpha,N} = \Big(I - \beta E_{\alpha,N}\Big)^{-1} u_\alpha,$$

where $(I - \beta E_{\alpha,n})$ exists and has the following geometric series representation:

$$(I - \beta E_{\alpha,N})^{-1} = \sum_{t=0}^{\infty} [\beta E_{\alpha,N}]^t.$$

where I is the $N \times N$ identity matrix and $E_{\alpha,N}$ is the $N \times N$ Markov transition matrix with (i,j) entry given by:

$$E_{\alpha,N}[i,j] = \Big[p_N(s_i|s_j,\alpha(s_j))\Big].$$

2. **Policy Improvement Step:** Compute improved policy α' using V_α:

$$\alpha'(s) = \underset{a\in A(s)}{argmax}[u(s,a) + \beta \sum_{i=1}^{N} V_{\alpha,N}(s_i)p_N(s_i|s,a)].$$

4. Parametric Policy Iteration

This algorithm is basically the same as the infinite dimensional version of policy iteration, except that we approximately solve each policy valuation step by approximating the solution V_α as a linear combination of k basis functions $\{\rho_1,\ldots,\rho_k\}$. Thus, suppose we set

$$V_\alpha(s) \simeq \sum_{i=1}^{k} \theta_i\rho_i(s).$$

Then the equation for V_α

$$V_alpha(s) = u(s,\alpha(s)) + \beta \int V_\alpha(s')p(s'|s,\alpha(s)ds'$$

is transformed into a linear equation with k unknown parameters $\theta \equiv \{\theta_1,\ldots,\theta_k\}$:

$$\sum_{i=1}^{k} \theta_i\rho_i(s) = u(s,\alpha(s)) + \beta \int \sum_{i=1}^{k} \theta_i\rho_i(s')p(s'|s,\alpha(s))ds' \tag{4.1}$$

Suppose we evaluate the above equation at a set of N points in S, with $N \geq k$. Then define the $(N \times k$ matrices P and EP with elements $P_{j,k}$ and $EP_{j,k}$ given by

$$P_{j,k} = \rho_k(s_j)$$
$$EP_{j,k} = \int \rho_k(s')p(s'|s_j, \alpha(s_j)) \tag{4.2}$$

Further define the $(N \times 1)$ vector y with $j - th$ element y_j given by

$$y_j = u(s_j, \alpha(s_j)).$$

and let the $(N \times k)$ matrix X be given by

$$X = (P - \beta EP)$$

Then the system of equations (4.1) can be written in matrix form as

$$y = X\theta.$$

If $N = k$ and X is invertible the solution for θ is simply

$$\theta = y/X = X^{-1}y.$$

If $N > k$ we have an overdetermined system and in general there is no $\theta \in R^k$ that allows us to exactly solve $y = X\theta$. However we can form an approximate solution using the ordinary least squares estimator (OLS), i.e. the value of θ that mininizes the distance $||y - X\theta||^2$, is given by

$$\theta = y/X = (X'X)^{-1}X'y \tag{4.3}$$

Definition: *The parametric policy iteration* algorithm is same as the infinite dimensional policy iteration algorithm described above, execpt that the policy valuation step is solved by approximating V_α as a linear combination of basis functions using the OLS estimator given in equation (4.3).

In general we will not be able to exactly integrate the basis functions and must use a quadrature rule to approximate the elements of EP given in equation (4.2)Thus, the PPI algorith requires the following choices:

1. The quadrature rule for computing the elements of EP.

2. The sample points $\{s_1, \ldots, s_N\}$ at which P and EP are evaluated at.

3. The set of basis functions $\{\rho_1, \ldots, \rho_k\}$.

Note that the policy improvement step would only be done on the same N points $\{s_1, \ldots, s_N\}$. Thus only NK numerical integrations and N maximizations are required for each policy valuation and policy improvement step, so if N and k can be chosen to be small, it is possible to find an approximation solution to the DP problem with amazingly few computations, provided the basis functions $\{\rho_1(s), \ldots, \rho_k(s)\}$ are sufficiently easy to evaluate at each $s \in S$. The resulting solution is defined by a parameter vector θ^* that enables us to evaluate $V(s) = \sum_{k=1} \theta_i^* \rho_i(s)$ very rapidly at any $s \in S$. Evaluating the corresponding decision $\alpha(s)$ at that point would require an approximate solution to

$$\alpha(s) = \max_{a \in A(s)} \left[u(s,a) + \beta \int \sum_{i=1}^{k} \theta_i^* \rho_i(s') p(s'|s,a) ds' \right].$$

In many cases this can be done quite rapidly, or alternatively, using the values $\{\alpha(s_i)\}$, $j = 1, \ldots, N$ from the last step of policy iteration, it might be possible to interpolate values of $\alpha(s)$ for $s \in \{s_1, \ldots, s_N\}$ if the decision rule is sufficiently smooth.

5. Numerical experiments

This section presents some numerical experiments that shows that in cases where V_α lies in a a "good space" for which there is a set of basis functions $\{\rho_1, \ldots, \rho_k\}$ which provide a good approximation to V_α, the PPI algorithm can work very well. However in cases where the "wrong" basis is chosen, it can be difficult or impossible to provide a close approximation to V_α even as $k \to \infty$.

I focus on the policy valuation step since this is the key to the algorithm. Thus, I ignore the dependence on a for now and consider two test problems for which analytic solutions can be derived:

Case 1

$$V(s) = ln(s) + \beta \int V(s') p(s'|s) ds'.$$

If $p(s'|s)$ is a lognormal density with $ln(s') = a + bln(s) + \epsilon$ where $\epsilon \simeq N(0, \sigma^2)$ with $\sigma^2 > 0$ and $b \in (0,1)$ the solution is given by

$$V(s) = \frac{\beta a}{(1-\beta)(1-b\beta)} + \frac{ln(x)}{(1-b\beta)}$$

Case 2

$$V(s) = u(s) + \beta \int V(s')p(s'|s).$$

If $p(s'|s)$ is a normal density with $s' = a + bs + \epsilon$ with $\epsilon \simeq N(0, \sigma^2)$ and $b \in (0, 1)$ and $u(s)$ is a polynomial, then $V(s)$ is also a polynomial with degree equal to the highest degree of $u(s)$. For example, if

$$u(s) = k_0 + k_1 s + k_2 s^2 + k_3 s^3$$

we have

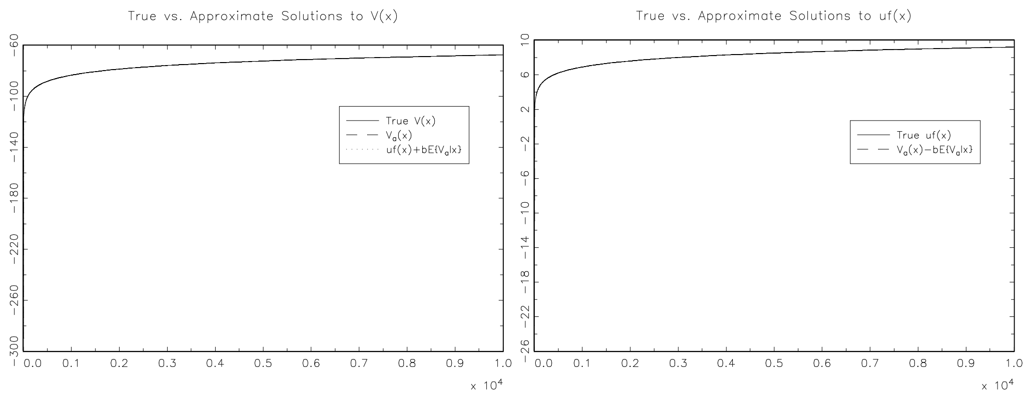
$$V(s) = \gamma_0 + \gamma_1 s + \gamma_2 s^2 + \gamma_3 s^3$$

where

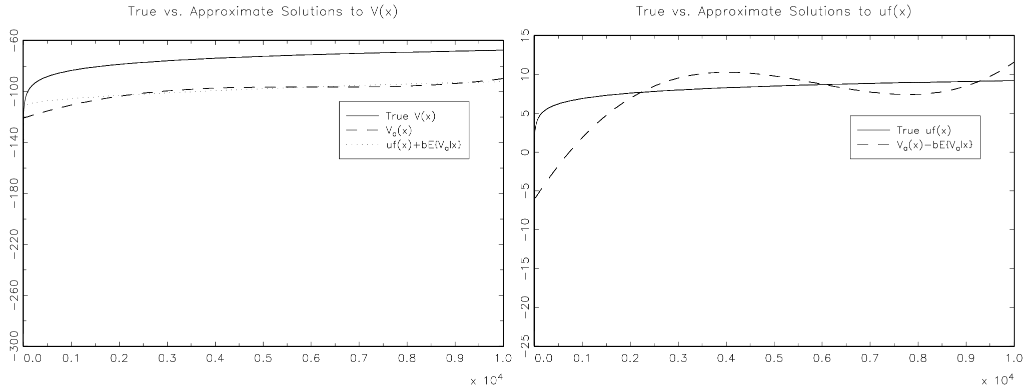
$$\begin{aligned}
\gamma_3 &= k_3/(1 - \beta b^3) \\
\gamma_2 &= (k_2 + 3\beta k_3 a b^2)/(1 - \beta b^2) \\
\gamma_1 &= (k_1 + 2ab\beta k_2 + 3\beta b a^2 k_3)/(1 - \beta b) \\
\gamma_0 &= (k_0 + a\beta k_1 + \beta k_2(\sigma^2 + a^2) + \beta k_3 a^3)/(1 - \beta)
\end{aligned}$$

Below I plot analytical versus approximate solutions using the following basis functions:

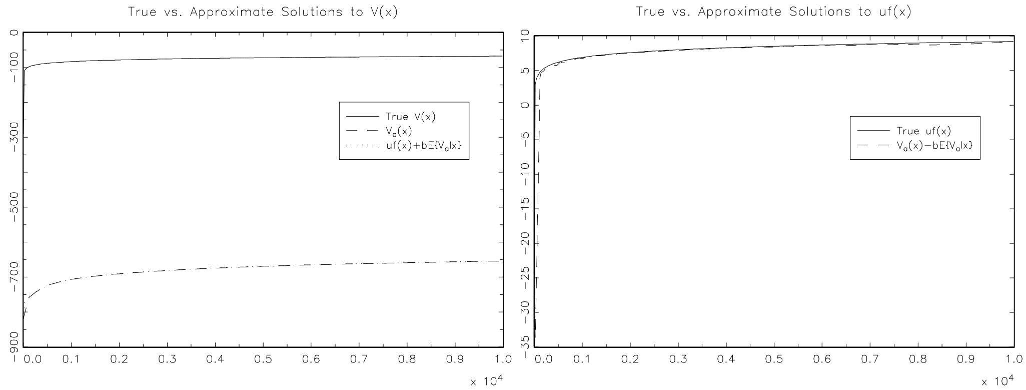
1. poly-log basis $\rho_i(s) = [ln(s)]^i$.
2. standard polynomial basis $\rho_i(s) = s^i$.
3. Chebyshev polyonmial basis $\rho_i(s) = cos(i\, cos^{-1}\,((2s - \underline{s} - \overline{s})/(\overline{s} - \underline{s}))$.
4. linear spline basis, $\rho_i(s)$ is a tent function equal to 1 at grid point s_j, 0 for $s < s_{j-1}$ or $s > s_{j+1}$, and linear with slope $1/(s_j - s_{j-1})$ on the interval $(s_{j-1}, s_j]$ and linear with slope $-1/(s_{j+1} - s_j)$ on the interval $(s_j, s_{j+1}]$.



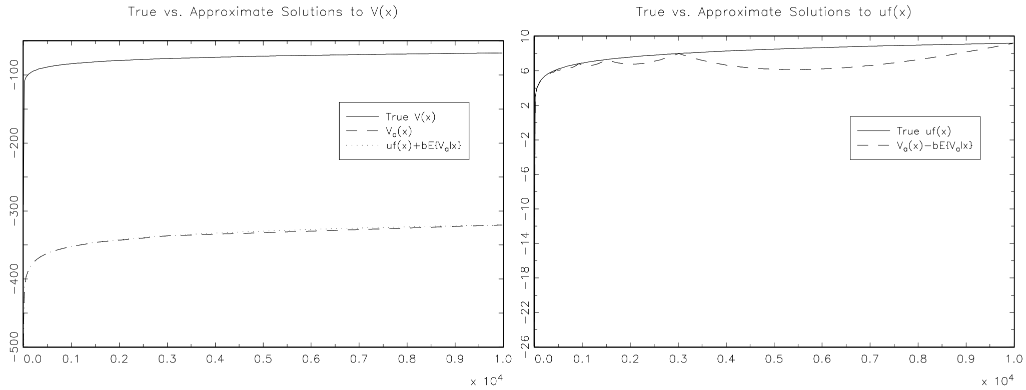
Case 1 Solution: Poly Log Basis



Case 1 Solution: Chebyshev Basis



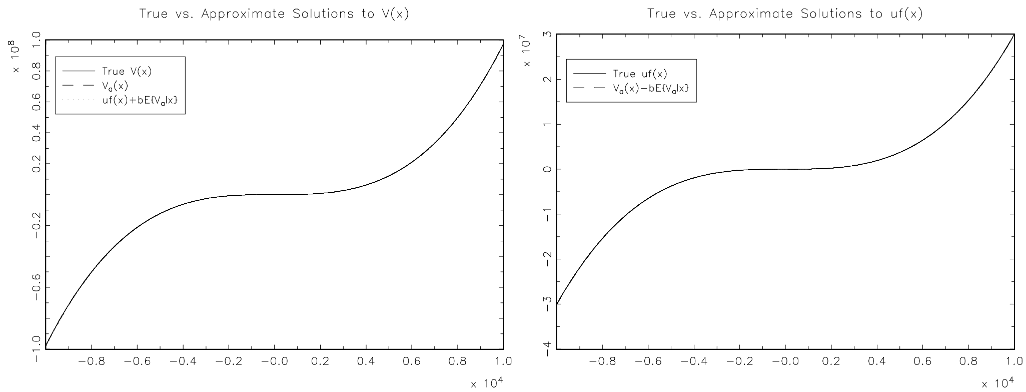
Case 1 Solution: Linear Spline Basis (uniform grid)



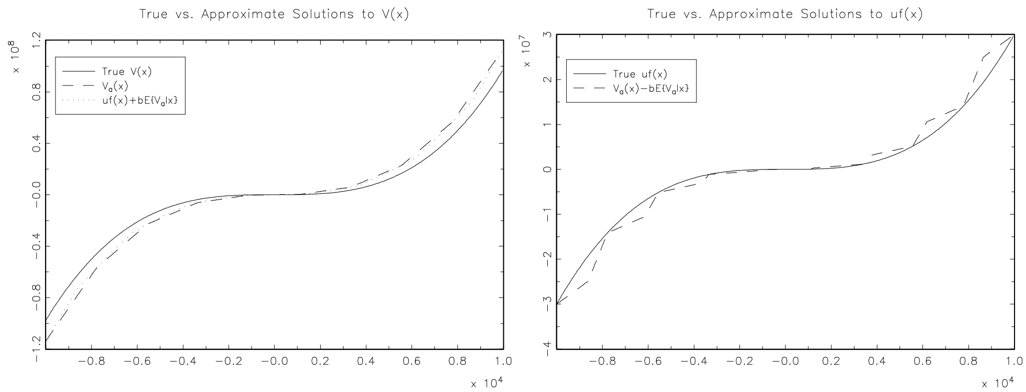
Case 1 Solution: Linear Spline Basis (uneqally spaced grid)

6. Closed-form solutions to Bellman equations

We now consider two classes of examples of continuous-state, continuous-decision Markov decision processes that are related to the linear functional equations given above.



Case 2 Solution: Chebyshev Basis



Case 2 Solution: Linear Spline Basis

The first example is Phelp's model of optimal consumption and savings, and a second is a family of linear-quadratic-gaussian control problems.

Example 1: Consider the problem of optimal consumption and savings analyzed by Phelps (1962). In this case the state variable s denotes a consumer's current wealth, and the decision d is how much to consume in the current period. Since consumption is a continuous decision, we will use c_t rather than d_t to denote the values of the control variable, and let w_t to denote the state variable wealth. The consumer is allowed to save, but is not allowed to borrow against future income. Thus, the constraint set is $D(w) = \{c\,|0 \leq c \leq w\}$. The consumer can invest his savings in a single risky asset with random rate of return $\tilde{R}$ which is *IID* with distribution F. Thus, $p(dw_{t+1}|w_t, c_t) = F((dw_{t+1}/(w_t - c_t))$. Let the consumer's utility function be given by $u(w, c) = log(c)$. Then Bellman's equation for this problem is given by:

$$V^*(w) = \max_{0\leq c\leq w}[log(c) + \beta \int_0^\infty V^*(R(w-c))F(dR)] \qquad (6.1)1$$

Working backward from an initial conjecture $V = 0$ we see that at each stage V_t has the form, $V_t(w) = A_t \, log(w) + B_t$ for constants A_t and B_t. Thus, it is reasonable to conjecture that this form holds in the limit as well. Inserting the conjectured functional form $V^*(w) = A_\infty \, log(w) + B_\infty$ into ((6.1)1) and solving for the unknown coefficients A_∞ and B_∞ we find:

$$\begin{aligned} A_\infty &= 1/(1-\beta) \\ B_\infty &= log(1-\beta)/(1-\beta) + \beta \, log(\beta)/(1-\beta)^2 + \beta E\{log(\tilde{R})\}/(1-\beta)^2, \end{aligned} \tag{6.2)1}$$

and the optimal decision rule or *consumption function* is given by:

$$\delta^*(w) = (1-\beta)w. \tag{6.3)2}$$

Thus, the logarithmic specification implies that a strong form of the *permanent income hypothesis* holds in which optimal consumption is independent of the distribution F of investment returns.

Example 3: Sometimes one can derive a differential equation for V and in certain cases one can derive analytical solutions to this differential equation and use it to characterize the optimal decision rule. Consider, for example, the problem of optimal replacement of durable assets analyzed in Rust (1985,1986). In this case the state space $S = R_+$, where s_t is interpreted as a measure of the accumulated utilization of the durable (such as the odometer reading on a car). Thus $s_t = 0$ denotes a brand new durable good. At each time t there are two possible decisions {keep,replace} corresponding to the binary constraint set $D(s) = \{0, 1\}$ where $d_t = 1$ corresponds to selling the existing durable for scrap price $\underline{P}$ and replacing it with a new durable at cost $\overline{P}$. Suppose the level of utilitization of the asset each period has an exogenous exponential distribution. This corresponds to a transition probability p is given by:

$$p(ds_{t+1}|s_t, d_t) = \begin{cases} 1 - exp\{-\lambda(ds_{t+1} - s_t)\} & \text{if } d_t = 0 \text{ and } s_{t+1} \geq s_t \\ 1 - exp\{-\lambda(ds_{t+1} - 0)\} & \text{if } d_t = 1 \text{ and } s_{t+1} \geq 0 \\ 0 & otherwise. \end{cases} \tag{6.4}$$

Assume the per-period cost of operating the asset in state s is given by a function $c(s)$ and that the objective is to find an optimal replacement policy to minimize the expected discounted costs of owning the durable over an infinite horizon. Since minimizing a function is equivalent to maximizing its negative, we can define the utility function by:

$$u(s_t, d_t) = \begin{cases} -\, c(s_t) & \text{if } d_t = 0 \\ -\,[\,\overline{P} - \underline{P}\,] - c(0) & \text{if } d_t = 1. \end{cases} \tag{6.5}$$

Bellman's equation takes the form:

$$
\begin{aligned}
V^*(s) = max\Big[& -c(s) + \beta \int_s^\infty V^*(s')\lambda\, exp\{-\lambda(s'-s)\}ds', \\
& -[\,\overline{P} - \underline{P}\,] - c(0) + \beta \int_0^\infty V^*(s')\lambda\, exp\{-\lambda(s')\}ds'\Big].
\end{aligned}
\tag{6.6}
$$

Observe that V^* is a non-increasing, continuous function of s and that the second term on the right hand side of ((6.6)), the value of replacing the durable, is a constant independent of s. Note also that $\overline{P} > \underline{P}$ implies that it is never optimal to replace a brand-new durable $s = 0$. Let γ be the smallest value of s such that the agent is indifferent between keeping and replacing. Differentiating Bellman's equation ((6.6)), it follows that on the *continuation region,* $[0, \gamma)$, V^* satisfies the differential equation:

$$
V^{*'}(s) = -c'(s) + \lambda c(s) + \lambda(1-\beta)V^*(s). \tag{6.7}
$$

This is known as a *free boundary value problem* since the boundary condition:

$$
V^*(\gamma) = [\,\overline{P} - \underline{P}\,] + V^*(0) = -c(\gamma) + \beta V^*(\gamma) = \frac{-c(\gamma)}{1-\beta}, \tag{6.8}
$$

is determined endogenously. Equation ((6.7)) is a linear first order differential equation which can be integrated to yield the following closed-form solution for V^*:

$$
V^*(s) = max\Big[\frac{-c(\gamma)}{1-\beta}, \frac{-c(\gamma)}{1-\beta} + \int_s^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)(y-s)}]dy\Big], \tag{6.9}
$$

where γ is the unique solution to:

$$
[\,\overline{P} - \underline{P}\,] = \int_0^\gamma \frac{c'(y)}{1-\beta}[1 - \beta e^{-\lambda(1-\beta)y}]dy. \tag{6.10}
$$

It follows that the optimal decision rule is given by:

$$
\delta^*(s) = \begin{cases} 0 & \text{if } s \in [0, \gamma] \\ 1 & \text{if } s > \gamma. \end{cases} \tag{6.11}
$$

Example 3: Consider the following linear-quadratic-gaussian (LQG) control problem whose solution is given in the following theorem.

Theorem: *Let $S = R$ and $A(s) = S$, $\forall s \in S$. Consider and MDP with the following utility function and transition density:*

$$
u(s, a) = [\lambda_2 a^2 + \lambda_1 a + \lambda_0] + s[\rho_0 + \rho_1 a] + \mu s^2 \tag{6.12}
$$

$$p(s'|s,a) = \frac{1}{\sqrt{2\pi}\sigma}\, exp\left\{-(s' - \kappa_0 - \kappa_1 a - bs)^2/(2\sigma^2)\right\}$$
$$\sigma = [\eta_0 + \eta_1 a] \tag{6.13}$$

where:

$$\mu < 0, \quad \lambda_2 < 0, \quad \textit{and} \quad \rho_1^2 - 4\mu\lambda_2 < 0.$$

Then $V(s)$ is given by:

$$V(s) = \max_{a\in A(s)} \left[u(s,a) + \beta\int V(s')p(s'|s,a)ds'\right]$$
$$= \gamma_0 + \gamma_1 s + \gamma_2 s^2 \tag{6.14}$$

and the optimal decision rule $\alpha(s)$ is given by:

$$\alpha(s) = f_0 + f_1 s \tag{6.15}$$

where:

$$f_1 = -\frac{\rho_1 + 2\beta\gamma_2\kappa_1 b}{2\left[\lambda_2 + \beta\gamma_2(\eta_1^2 + \kappa_1^2)\right]}$$
$$f_0 = -\frac{\lambda_1 + \beta\gamma_1\kappa_1 + 2\beta\gamma_2(\eta_0\eta_1 + \kappa_0\kappa_1)}{2\left[\lambda_2 + \beta\gamma_2(\eta_1^2 + \kappa_1^2)\right]} \tag{6.16}$$

where:

$$\gamma_2 = \frac{-k_1 - \sqrt{k_1^2 - 4k_2k_0}}{2k_2}$$
$$\gamma_1 = \frac{\rho_0 + 2\beta\gamma_2\kappa_0 b + f_1\left[\lambda_1 + 2\beta\gamma_2(\eta_0\eta_1 + \kappa_0\kappa_1)\right]}{1 - \beta(\kappa_1 f_1 + b)}$$
$$\gamma_0 = \frac{\lambda_2 f_0^2 + \lambda_1 f_0 + \lambda_0 + \beta\gamma_1(\kappa_0 + \kappa_1 f_0) + \beta\gamma_2\left[(\eta_0 + \eta_1 f_0)^2 + (\kappa_0 + \kappa_1 f_0)^2\right]}{(1-\beta)} \tag{6.17}$$

where:

$$k_0 = \rho_1^2 - 4\mu\lambda_2$$
$$k_1 = 4\left[\lambda_2(1 - \beta b^2) - \mu\beta(\eta_1^2 + \kappa_1^2) + \rho_1\kappa_1 b\beta\right]$$
$$k_2 = 4\left[\beta(\kappa_1^2 + \eta_1^2)(1 - \beta b^2) + \beta^2\kappa_1^2 b^2\right]. \tag{6.18}$$