Techniques in Computational Stochastic Dynamic Programming[1]

Floyd B. Hanson[2]

University of Illinois at Chicago
Chicago, Illinois 60607-7045

Excerpted Section

A. MARKOV CHAIN APPROXIMATION

Another approach to finite differences is the well developed *Markov Chain Approximation (MCA)* of Kushner [3, 4]. Recent developments are surveyed and further advanced by Kushner [5], and by Kushner and Dupuis [6], with special attention to methods for jump and reflected diffusions. This method applies a Markov chain approximation to continuous time, continuous state stochastic control problems by renormalizing finite differences forms as proper Markov chain transition probabilities. These transition probabilities arise when deriving finite difference versions of the dynamic programming equation. An important advantage of this method is that the Markov chain approximation facilitates convergence proofs for the numerical methods in terms of probabilistic arguments. Probabilistic interpretation of the approximation is a major motivation for the formulation of this method. Here, the MCA method is given a formal presentation, in the spirit of the SDP notation and formulation to facilitate comparison. The reader should refer to the above references for the greater detail, especially Kushner and Dupuis [6] for a multitude of variations and convergence proofs.

1. MCA Dynamic Programming Model Formulation

Consider the stochastic diffusion without Poisson jumps governed by the stochastic differential equation (SDE)

$$d\mathbf{X}(t) = \mathbf{F}(\mathbf{X}, \mathbf{U}, t)dt + G(\mathbf{X}, t)d\mathbf{W}(t), \tag{1}$$

where the notation is the same as in *the full paper [2]*. It is assumed that drift $\mathbf{F}$ and Gaussian coefficient G are bounded, continuous and Lipshitz continuous in the state $\mathbf{X}$, while $\mathbf{F}$ is uniformly so in the control $\mathbf{U}$. Further, let the expected cost objective functional be

$$\begin{aligned} \overline{V}(\mathbf{x}, \mathbf{u}, t) &= \text{Mean}\left[\int_t^{t_f} d\tau\, C(\mathbf{X}(\tau), \mathbf{U}(\mathbf{X}(\tau), \tau), \tau)\right. \\ &\quad \left. \mid \mathbf{X}(t) = \mathbf{x}, \mathbf{U} = \mathbf{u}\right] + \overline{Z}(\mathbf{x}, t_f), \end{aligned} \tag{2}$$

[1] This section on Markov Chain Approximation appeared in the chapter: F. B. Hanson, "Techniques in Computational Stochastic Dynamic Programming" in *Stochastic Digital Control System Techniques*, within series *Control and Dynamic Systems: Advances in Theory and Applications*, vol. 76, (C. T. Leondes, Editor), Academic Press, New York, NY, pp. 103-162, April 1996.

[2] Supported in part by National Science Foundation Grant DMS 93-0117, National Center for Supercomputing Applications, Pittsburgh Supercomputing Center, and Los Alamos National Laboratory's Advanced Computing Laboratory; written while on sabbatical in Division of Applied Mathematics at Brown University, Providence, RI 02912

with the same notation as in *the full paper [2]*. It is assumed that the instantaneous cost C and the salvage cost $\overline{Z}$ are bounded and continuous. Much also may depend on the boundary conditions. The final side condition is the same as that in *the full paper [2]* for SDP.

The optimal costs are defined as

$$v^*(\mathbf{x},t) = \inf_{\mathbf{u}} \left[\overline{V}(\mathbf{x},\mathbf{u},t)\right], \tag{3}$$

using the infimum (inf), instead of the less general minimum (min) in the spirit of [6], over all admissible controls.

Upon application of the principle of optimality, the dynamic programming equation for the optimal expected cost v^* is

$$\begin{aligned} 0 &= v_t^* + \mathcal{L}_x[v^*](\mathbf{x},t) \\ &= v_t^* + \inf_{\mathbf{u}} \Big[\mathbf{F}^T(\mathbf{x},\mathbf{u},t)\nabla_x[v^*] + \frac{1}{2}(GG^T)(\mathbf{x},t) : \nabla_x\nabla_x^T[v^*] \\ &+ C(\mathbf{x},\mathbf{u},t)\Big]. \end{aligned} \tag{4}$$

Since (4) is a backward equation and since we want to keep it simple, (4) is approximated with the Backward Euler approximation

$$\begin{aligned} v_{\kappa-1}^*(\mathbf{x}) &= v_\kappa^*(\mathbf{x}) + \Delta t_{\kappa-1} \cdot \inf_{\mathbf{u}} \Big[\mathbf{F}_\kappa^T(\mathbf{x},\mathbf{u})\nabla_x[v_\kappa^*] \\ &+ \frac{1}{2}(GG^T)_\kappa(\mathbf{x}) : \nabla_x\nabla_x^T[v_\kappa^*] + C_\kappa(\mathbf{x},\mathbf{u})\Big], \end{aligned} \tag{5}$$

for $\kappa = 1$ to K, with $v_\kappa^*(\mathbf{x}) \simeq v^*(\mathbf{x},t_\kappa)$, and similarly for $\mathbf{F}_\kappa$, G_κ and C_κ, while $t_\kappa = t_{\kappa-1} + \Delta t_{\kappa-1}$ is time in terms of the forward index κ. The final condition is $v_K^*(\mathbf{x}) = v^*(\mathbf{x},t_f) = \overline{Z}^*(\mathbf{x},t_f)$. (The correlation, in the case of constant time increments, between the forward index κ and the backward index k used in SDP is that $T_k = t_f - k\cdot\Delta T = (K-k)\cdot\Delta T = t_{K-k} = t_\kappa$, so the forward and backward time indices are related by $\kappa = K-k$. Hence, $T_0 = t_f = t_K$ and $T_K = 0 = t_0$, are the final and initial times, respectively, in either time direction.) The interpolation time increment is denoted by $\Delta t_{\kappa-1} = t_\kappa - t_{\kappa-1}$ here and has important roles to play for modeling and for numerical convergence.

However, since much of the literature on MCA, corresponding to much of the literature on stochastic control, is formulated as a stationary (time-independent) problem, the indices k and κ are treated as iteration indices for the time-independent problem. The time-independent problem may arise from ergodic problems or exit time problems or infinite horizon problems, for example.

2. MCA Local Consistency Conditions

The symbol h will indicate the the order of the spacing in the discretization of the state space for the Markov chain $\boldsymbol{\xi}_\kappa$ for discrete steps $\kappa \geq 0$ (note for the chain we use the forward time index κ instead of the backward time index k to give priority to the Markov chain properties usually defined in forward time, instead of completely forcing the backward time notion of SDP). The Markov chain transition probabilities are defined by

$$p(\boldsymbol{\xi}_\kappa,\boldsymbol{\eta}|\mathbf{u}_\kappa) = \text{Prob}[\boldsymbol{\xi}_{\kappa+1} = \boldsymbol{\eta}|\boldsymbol{\xi}_j,\mathbf{u}_j, j\leq\kappa]$$

for transitions of the Markov chain from stage $\boldsymbol{\xi}_\kappa$ to stage $\boldsymbol{\xi}_{\kappa+1} = \boldsymbol{\eta}$ under control policy $\mathbf{u}_\kappa$. These transition probabilities must satisfy the non-negativity ($p \geq 0$) and conservation ($\sum p = 1$) of probability properties of any proper probability law. Thus, the chain is defined on a finite state space $\mathcal{D}_\xi$ with discrete time parameter κ. Further, the Markov chain approximation increments $\Delta\boldsymbol{\xi}_\kappa \equiv \boldsymbol{\xi}_{\kappa+1} - \boldsymbol{\xi}_\kappa$, with $|\Delta\boldsymbol{\xi}_\kappa| = \mathcal{O}(h)$, must satisfy the *local consistency conditions*

$$\begin{aligned} E[\Delta\boldsymbol{\xi}_\kappa|\boldsymbol{\xi}_\kappa,\mathbf{u}_\kappa] &\equiv \sum_\eta (\boldsymbol{\eta}-\boldsymbol{\xi}_\kappa)\cdot p(\boldsymbol{\xi}_\kappa,\boldsymbol{\eta}|\mathbf{u}_\kappa) \\ &= \Delta t_\kappa \cdot [\mathbf{F}_\kappa(\boldsymbol{\xi}_\kappa,\mathbf{u}_\kappa) + o(1)] \\ &\text{and} \end{aligned} \tag{6}$$

$$\begin{aligned} \text{Covar}[\Delta\boldsymbol{\xi}_\kappa|\boldsymbol{\xi}_\kappa,\mathbf{u}_\kappa] &\equiv \sum_{\boldsymbol{\eta}}(\boldsymbol{\eta}-\boldsymbol{\xi}_\kappa-E[\Delta\boldsymbol{\xi}_\kappa])(\boldsymbol{\eta}-\boldsymbol{\xi}_\kappa-E[\Delta\boldsymbol{\xi}_\kappa])^T \\ &\cdot\ p(\boldsymbol{\xi}_\kappa,\boldsymbol{\eta}|\mathbf{u}_\kappa) \\ &= \Delta t_\kappa\cdot\left[(G_\kappa G_\kappa^T)(\boldsymbol{\xi}_\kappa)+o(1)\right], \end{aligned}$$

as $h\longrightarrow 0^+$, for $\kappa=0$ to $K-1$, consistent with the usual conditions for the first two infinitesimal moments of the stochastic diffusion approximation corresponding to the SDE (1). In (6), the conditioning on earlier states and controls $\{\boldsymbol{\xi}_j,\mathbf{u}_j,j<\kappa\}$ has been suppressed to simplify the notation, but is not meant to imply any assumption on earlier variables.

Here, Δt_κ is the local interpolation time increment, such that $\Delta t_\kappa\longrightarrow 0$ as $h\longrightarrow 0^+$, and such that the piecewise constant interpolated chain and control with continuous time parameter t are

$$\boldsymbol{\xi}(t)\equiv\boldsymbol{\xi}_\kappa \text{ and } \mathbf{u}(t)\equiv\mathbf{u}_\kappa \text{ for } t \text{ on } [t_\kappa,t_\kappa+\Delta t_\kappa),$$

which together with the local consistency conditions (6) are satisfied by the corresponding discrete time chain allow approximation of (1). For generality, $\Delta t_\kappa=\Delta t_\kappa(\boldsymbol{\xi}_\kappa,\mathbf{u}_\kappa)$ is permitted.

3. MCA Dynamic Programming Equation and Transition Probabilities Construction from Finite Differences

The MCA transition probabilities $p(\boldsymbol{\xi},\boldsymbol{\eta}|\mathbf{u})$ often are constructed from the finite difference in space of the parabolic, dynamic programming equation (5) (finite elements could be used as well). The usual mixed finite differences in space used in [6] are upwinded (forward and backward) for first order state derivatives,

$$\frac{\partial V}{\partial x_i}(\mathbf{x},t)\simeq\left\{\begin{array}{ll}\frac{V(\mathbf{x}+h_i\mathbf{e}_i,t)-V(\mathbf{x},t)}{h_i}, & F_i(\mathbf{x},\mathbf{u}(\mathbf{x},t),t)\geq 0\\ \frac{V(\mathbf{x},t)-V(\mathbf{x}-h_i\mathbf{e}_i,t)}{h_i}, & F_i(\mathbf{x},\mathbf{u}(\mathbf{x},t),t)<0\end{array}\right\}, \tag{7}$$

and central differences for the second order derivatives,

$$\frac{\partial^2 V}{\partial x_i^2}(\mathbf{x},t)\simeq\frac{V(\mathbf{x}+h_i\cdot\mathbf{e}_i,t)-2V(\mathbf{x},t)+V(\mathbf{x}-h_i\cdot\mathbf{e}_i,t)}{h_i^2}, \tag{8}$$

with $\mathbf{e}_i$ as the unit vector for the ith state component and $h_i=\mathcal{O}(h)$ as the size of the ith step. Here, we assume that the Gaussian noise is not correlated (i.e., we mean the coefficient is assumed to be diagonal), so the cross second derivatives are not needed. Also, the upwinding on first derivatives sacrifices accuracy (i.e., it has $\mathcal{O}(h_i)$ error rather than the smaller $\mathcal{O}(h_i^2)$ error of central differences as $h_i\longrightarrow 0^+$, so the larger error "numerically pollutes" the smaller) for potentially greater numerical stability in the case of convection dominated flows (i.e., the drift part $|F_i|/h_i$ is greater than the diffusion part $(GG^T)_{i,i}/h_i^2$). However, many refinements to fix up these problems are found in Kushner and Dupuis [6], such as higher order finite differences including those needed for cross second derivatives.

Upon substituting these finite differences, the dynamic programming equation (5) becomes

$$\begin{aligned} v^*(\mathbf{x},t_{\kappa-1}) &= \inf_{\mathbf{u}_{\kappa-1}}\left[p_\kappa(\mathbf{x},\mathbf{x},|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x},t_\kappa)\right. \\ &+ \sum_{i=1}^n p_\kappa(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x}+h_i\cdot\mathbf{e}_i,t_\kappa) \\ &+ \sum_{i=1}^n p_\kappa(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x}-h_i\cdot\mathbf{e}_i,t_\kappa) \\ &+ \left.\Delta t_{\kappa-1}\cdot C(\mathbf{x},\mathbf{u}_{\kappa-1},t_\kappa)\right], \end{aligned} \tag{9}$$

where the transition probabilities, now denoted by

$$p(\boldsymbol{\xi}_{\kappa-1},\boldsymbol{\xi}_\kappa|\mathbf{u}_{\kappa-1})\simeq p_\kappa(\boldsymbol{\xi},\boldsymbol{\eta}|\mathbf{u}_{\kappa-1})$$

and activated by the control vector $\mathbf{u}_{\kappa-1}$ from $t_{\kappa-1}$ to t_κ, are given by

$$p_\kappa(\mathbf{x},\mathbf{x},|\mathbf{u}_{\kappa-1}) = 1-\Delta t_{\kappa-1}\cdot\sum_{j=1}^{n}\left[\frac{(GG^T)_{j,j,\kappa}}{h_j^2}+\frac{|F_{j,\kappa}|}{h_j}\right], \tag{10}$$

$$p_\kappa(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1}) = \Delta t_{\kappa-1}\cdot\left[\frac{(GG^T)_{i,i,\kappa}}{2h_i^2}+\frac{[F_{i,\kappa}]_+}{h_i}\right], \tag{11}$$

$$p_\kappa(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1}) = \Delta t_{\kappa-1}\cdot\left[\frac{(GG^T)_{i,i,\kappa}}{2h_i^2}+\frac{[F_{i,\kappa}]_-}{h_i}\right], \tag{12}$$

for sufficiently small $\Delta t_{\kappa-1}$ so $p_\kappa(\mathbf{x},\mathbf{x},|\mathbf{u}_{\kappa-1})$ in (10) is non-negative. Here $[F]_\pm\equiv\max[\pm F,0]\geq 0$ (i.e., $[F]_+ + [F]_- = |F|$ and $[F]_+ - [F]_- = F$). The drift component is $F_{i,\kappa}=F_i(\mathbf{x},\mathbf{u}_{\kappa-1},t_\kappa)$ and the diffusion coefficient is $G_{i,i,\kappa}=G_{i,i}(\mathbf{x},t_\kappa)$, assuming that only the diagonal part is needed. While the explicit time-dependence of the dynamical system coefficients are evaluated at the later time t_κ (this is not a critical requirement) as would be from the Backward Euler approximation, the control is evaluated at the earlier time $t_{\kappa-1}$. The procedure generates feedback control at the discrete level. This is because the control policy, when known, would be set at the beginning for the period in forward time, although this control policy is determined by the dynamic program in backward time sequence. Kushner and Dupuis [6] call this type of MCA an explicit method, but since, as a backward procedure, it is explicit in the costs v_κ^* while implicit in the control $\mathbf{u}_{\kappa-1}$, it is really quasi-explicit in the case of the simple approximations used here. However, a number of implicit methods are also presented by Kushner and Dupuis [6].

In (10,11,12), the really important benefit of upwinding is seen, in that upwinding ensures that the drift coefficient $F_{i,\kappa}$ only enters the transition probabilities as an absolute value, guaranteeing the non-negativity property of the non-self transition probabilities in (11,12), and (10) as well, provided the interpolation increment is sufficiently small. Note that the requirement that the self transition term (10) must be non-negative as a probability puts restrictions on the time step $\Delta t_{\kappa-1}$, i.e.,

$$\Delta t_{\kappa-1}\leq\frac{1}{\sum_{i=1}^{n}\left[\frac{(GG^T)_{i,i,\kappa}}{h_i^2}+\frac{|F_{i,\kappa}|}{h_i}\right]}, \tag{13}$$

which is something like the generalized Courant Friedrichs and Lewy (CFL) condition *the full paper [2]* for SDP. This is more readily seen in the form:

$$\sigma_{MCA}=\Delta t_{\kappa-1}\cdot\sum_{i=1}^{n}\left[\frac{(GG^T)_{i,i,\kappa}}{h_i^2}+\frac{|F_{i,\kappa}|}{h_i}\right]\leq 1, \tag{14}$$

which is more like the "$\mathcal{L}_1$" norm version of the "$\mathcal{L}_2$" norm version of the generalized CFL condition in *the full paper [2]*. Thus the conditions for both methods require the same order of magnitude of the time step. The differences in quasi-norms is not significant given the approximations in both methods.

One can check that the transition probabilities do indeed satisfy the MCA local consistency conditions (6). For instance, the first infinitesimal moment of the MCA is calculated as follows,

$$\begin{aligned}
E[\Delta\boldsymbol{\xi}_\kappa|\boldsymbol{\xi}_\kappa=\mathbf{x}] &= p_\kappa(\mathbf{x},\mathbf{x}|\mathbf{u}_{\kappa-1})\cdot[0]\\
&+ \sum_{i=1}^{n}p_\kappa(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot[h_i\cdot\mathbf{e}_i]\\
&+ \sum_{i=1}^{n}p_\kappa(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot[-h_i\cdot\mathbf{e}_i]\\
&= \Delta t_{\kappa-1}\cdot\sum_{i=1}^{n}([F_{i,\kappa}]_+-[F_{i,\kappa}]_-)\cdot\mathbf{e}_i\\
&= \Delta t_{\kappa-1}\cdot\sum_{i=1}^{n}F_{i,\kappa}\cdot\mathbf{e}_i=\Delta t_{\kappa-1}\cdot\mathbf{F}(\mathbf{x},\mathbf{u}_{\kappa-1},t_\kappa),
\end{aligned}$$

demonstrating that the condition is satisfied for the first moment. The consistency condition can be similarly demonstrated for the second moment.

In the case of a stationary dynamic programming equation, i.e., the elliptic case, such as for an exit time problem, optimal stopping problem or infinite horizon, then the coefficients from the finite difference formula are renormalized to satisfy properties of transition probabilities. Renormalization is not essential for the non-stationary case described above. In the stationary case, renormalization is computationally useful in that it can help speed up the computation if wisely done. Typically, the stationary self-transition term $\widehat{p}_\kappa(\mathbf{x},\mathbf{x}|\mathbf{u})$ is set to zero and the central cost $v^*(\mathbf{x})$ is solved for as a function of the nontrivial transition costs, so the interpolation or iteration time $\widehat{\Delta t}$ is selected to be the resultant coefficient of the instantaneous cost $C(\mathbf{x},\mathbf{u})$. For example, suppose that the final horizon time $t_f = \tau_e$, the first exit time for the chain to reach to the boundary, then the non-stationary case equations (9,11,12) are reformulated (presumably before taking the infimum) for the stationary case as the iteration

$$\begin{aligned}\widehat{v}^*_{\hat\kappa+1}(\mathbf{x}) &= \inf_{\widehat{\mathbf{u}}_{\hat\kappa+1}}\Bigg[\sum_{i=1}^n \widehat{p}(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\widehat{\mathbf{u}}_{\hat\kappa+1})\cdot\widehat{v}^*_{\hat\kappa}(\mathbf{x}+h_i\cdot\mathbf{e}_i) \\ &+ \sum_{i=1}^n \widehat{p}(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\widehat{\mathbf{u}}_{\hat\kappa+1})\cdot\widehat{v}^*_{\hat\kappa}(\mathbf{x}-h_i\cdot\mathbf{e}_i) \\ &+ \widehat{\Delta t}\cdot C(\mathbf{x},\widehat{\mathbf{u}}_{\hat\kappa+1})\Bigg]. \end{aligned} \tag{15}$$

A new iteration index $\hat\kappa$ replaces the former κ which loses its meaning a time index in the stationary problem. The transition probabilities are now denoted by

$$p(\boldsymbol{\xi}_{\hat\kappa+1},\boldsymbol{\xi}_{\hat\kappa}|\widehat{\mathbf{u}}_{\hat\kappa+1}) \simeq \widehat{p}_{\hat\kappa}(\boldsymbol{\xi}_{\hat\kappa+1},\boldsymbol{\xi}_{\hat\kappa}|\widehat{\mathbf{u}}_{\hat\kappa+1}),$$

conditioned by the control vector $\widehat{\mathbf{u}}_{\hat\kappa+1}$ from iterations $\hat\kappa$ to $\hat\kappa+1$. The condition probabilities are now given by

$$\widehat{p}(\mathbf{x},\mathbf{x}|\widehat{\mathbf{u}}_{\hat\kappa+1}) \equiv 0,$$

$$\widehat{p}(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\widehat{\mathbf{u}}_{\hat\kappa+1}) = \widehat{\Delta t}\cdot\left[\frac{(GG^T)_{i,i}}{2h_i^2}+\frac{[F_i]_+}{h_i}\right], \tag{16}$$

$$\widehat{p}(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\widehat{\mathbf{u}}_{\hat\kappa+1}) = \widehat{\Delta t}\cdot\left[\frac{(GG^T)_{i,i}}{2h_i^2}+\frac{[F_i]_-}{h_i}\right]. \tag{17}$$

The starting interpolation time increment $\widehat{\Delta t}$ is given by the reciprocal of $[1-p(\mathbf{x},\mathbf{x}|\mathbf{u})]/\Delta t_{\kappa-1}$ from the non-stationary case as

$$\widehat{\Delta t} = \frac{1}{\sum_{j=1}^n\left[\frac{(GG^T)_{j,j}}{h_j^2}+\frac{|F_j|}{h_j}\right]}.$$

For this stationary case it is assumed that the SDE coefficients are autonomous, i.e., $F_i = F_i(\mathbf{x},\mathbf{u})$ and $G_{i,i} = G_{i,i}(\mathbf{x})$, for consistency. The former non-stationary interpolation time increment $\Delta t_{\kappa-1}$ cancels out of the discrete time dynamic programming equation (5).

4. MCA Dynamic Programming Equation Solution

The dynamic programming equation (9) can be solved for $v^*_{\kappa-1}$ and simultaneously for the optimal control vector approximation for MCA as the infimum or minimum argument:

$$\begin{aligned}\mathbf{u}^*(\mathbf{x},t_{\kappa-1}) &= \arg\inf_{\mathbf{u}_{\kappa-1}}\left[p_\kappa(\mathbf{x},\mathbf{x},|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x},t_\kappa)\right. \\ &+ \sum_{i=1}^n p_\kappa(\mathbf{x},\mathbf{x}+h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x}+h_i\cdot\mathbf{e}_i,t_\kappa) \\ &+ \sum_{i=1}^n p_\kappa(\mathbf{x},\mathbf{x}-h_i\cdot\mathbf{e}_i|\mathbf{u}_{\kappa-1})\cdot v^*(\mathbf{x}-h_i\cdot\mathbf{e}_i,t_\kappa) \\ &+ \left.\Delta t_{\kappa-1}\cdot C(\mathbf{x},\mathbf{u}_{\kappa-1},t_\kappa)\right]. \end{aligned} \tag{18}$$

Kushner and Dupuis [6] discuss many computational methods for solving problems like (9), such as Point-Jacobi, Gauss-Seidel, Successive-Over-Relaxation accelerations, Multigrid or Multilevel, and Domain Decomposition. They also discuss both the benefits and disadvantages of these methods. Please refer to their book for more information. The paper of Kushner and Jarvis [7] gives the recent state of MCA computations using parallel and vector supercomputers. Quadrat and coworkers [1] have used MCA in their expert system for solving stationary optimal control problems.

Jump diffusions or Poisson processes included with stochastic diffusions only requires a small amount of extra effort to calculate the modifications of the transition probabilities for MCA. The changes due diffusion and the jumps are separately accounted for since one is continuous and the other discontinuous.

Reflecting boundaries are handled for reflecting diffusions and reflecting jump diffusions by adding a border of extra discrete elements around the domain. The width of the border elements is $\mathcal{O}(h)$. Any chain that winds up in the border element if reflected back into the interior elements in a way consistent with the application and the stochastic noise, and locally consistent with the local direction of reflection. Non-normal reflections may be difficult to handle in multi-dimensions. The reflections are modeled by added auxiliary stochastic processes to the SDE (1) and these auxiliary stochastic processes are only activated when a chain hits the border element. This seems to be a proper way to handle the complexities of boundary conditions for stochastic processes, whether reflecting or otherwise. The recent paper of Kushner and Yang [8] gives a clear example of the MCA method with reflecting boundary conditions for telecommunication networks in a heavy traffic environment.

5. Similarities and Differences between MCA and SDP

The obvious difference between MCA and SDP, as presented here, is that MCA is is primarily applied to stationary problems, in spite for the dynamic model presentation here. The MCA method is extremely adaptable. Kushner and Dupuis [6] give variations for exit time problems, ergodic problems, jump diffusions, discounted costs, average costs, optimal filtering and many other applications. A major advantage or the method is that convergence proofs are facilitated by the probabilistic frame work, while they are generally difficult or not available for traditional numerical PDE methods presented for SDP here. However, the MCA method does introduce a indirect feature in the use of the Markov chain to solve a continuous time, continuous state optimal control problem. A principal similarity is that both MCA and SDP basically use Bellman's dynamic programming equations, although under quite different probabilistic and numerical interpretations. They are both variants of the stochastic dynamic programming algorithm. Although the control is usually the most important output for the control user, sometimes the expected optimal trajectories, i.e., using the optimal control, are desired to study the behavior of the dynamic system in the optimal state. There does not appear to be a formulation of MCA that covers the use of the forward equations for this purpose in Kushner and Dupuis [6], although simulations using random number generators are suggested.

ACKNOWLEDGEMENT

The author thanks his collaborators Dr. S.-L. Chung, Dr. H. H. Xu, Prof. K. Naimipour, Dr. D. J. Jarvis, J. J. Westman, C. J. Pratico and M. S. Vetter for their contributions. He also thanks his hosts, Prof. H. J. Kushner of Brown University and C. A. Shoemaker of Cornell University, for their hospitality and insights about computational control when visiting for a sabbatical year. In addition, he thanks Prof. P. G. Dupuis for his helpful comments.

I. REFERENCES

1. M. Akian, J. P. Chancelier and J. P. Quadrat, "Dynamic Programming Complexity and Applications," *Proceedings of 27th IEEE Conference on Decision and Control* **2**, pp. 1551-1558 (1988). *Mathematics of Computation* **33**, pp. 659-679 (1979).

2. F. B. Hanson, "Techniques in Computational Stochastic Dynamic Programming" in *Stochastic Digital Control System Techniques*, within series *Control and Dynamic Systems: Advances in Theory and Applications*, vol. 76, (C. T. Leondes, Editor), Academic Press, New York, NY, pp. 103-162, April 1996.

3. H. J. Kushner, "A Survey of Some Applications of Probability and Stochastic Control Theory to Finite Difference Methods for Degenerate Elliptic and Parabolic Equations," *SIAM Review* **18**, pp. 545-577, (1976).

4. H. J. Kushner, *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations*, Academic Press, New York, NY, 1977.

5. H. J. Kushner, "Numerical Methods for Stochastic Control in Continuous Time," *SIAM J. Control and Optimizations* **28**, pp. 999-1048 (1990).

6. H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control in Continuous Time*, Springer-Verlag, New York, NY, 1992.

7. H. J. Kushner and D. J. Jarvis, "Large-Scale Computations for High Dimension Control Systems," *Proceedings of 33rd IEEE Conference on Decision and Control* **1**, 6 pages (1994).

8. H. J. Kushner and J.-C. Yang, "A Numerical Method for Controlled Routing in Large Trunk Line Networks via Stochastic Control Theory," *ORSA Journal on Computing* **6** (3), pp. 300-316 (1994).