On Rydén's EM Algorithm for Estimating MMPPs

William J. J. Roberts, *Member, IEEE*, Yariv Ephraim, *Fellow, IEEE*, and Elvis Dieguez, *Member, IEEE*

***Abstract*—Two aspects of Rydén's expectation-maximization algorithm for estimating the parameter of a Markov modulated Poisson process are addressed. First, a scaling procedure is developed for the forward–backward recursions that circumvents the need for customized floating-point software. Second, evaluation of integrals of matrix exponentials is facilitated by applying a result due to Van Loan. For an MMPP of order four, a speedup of over two orders of magnitude was observed.**

***Index Terms*—Expectation-maximization (EM) algorithm, Markov-modulated Poisson process, maximum likelihood estimation.**

I. Introduction

A Markov-modulated Poisson process (MMPP) is a Poisson process with variable rate [5]. The rate is determined by an underlying continuous-time finite-state homogeneous Markov chain. MMPPs have many applications, for example, in medicine [14, Chap. 7], computer networks [6], and finance [11]. These processes have been thoroughly studied. Good surveys of their properties can be found in [4], [12], and [13].

Estimation of the parameter of an MMPP is of interest in many applications. The parameter comprises the initial state distribution and generator of the underlying Markov chain as well as the rates of the observed process. The number of states of the underlying chain is usually assumed known. A good survey of estimation techniques that have been applied to this problem can be found in [12]. In 1996, Rydén [13] developed an expectation-maximization (EM) algorithm for maximum-likelihood (ML) estimation of the parameter of an MMPP. Contrary to an earlier version [12], the algorithm has explicit E and M steps. Two computational difficulties with this algorithm were noted in [13]. The first is the need for a scaling procedure to stabilize the forward–backward recursions used to estimate the underlying Markov chain posterior state probabilities. The second difficulty concerns the evaluation of integrals of matrix exponentials. In [13], customized floating-point software was used to overcome the numeric instability of the forward–backward recursions, and the integrals of matrix exponentials were evaluated after the matrices have been diagonalized.

In this letter, we address these two computational aspects of the algorithm. We develop a scaling procedure for the forward–backward recursions, and we apply some results from control theory, developed by Van Loan [15], for evaluation of the integrals. We demonstrate that the application of Van Loan's result yields considerable computational speedup over the approach used in [13].

Manuscript received November 7, 2005; revised December 21, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Stefano Buzzi.

W. J. J. Roberts is with Atlantic Coast Technologies, Inc., Silver Spring, MD 20904-2545 USA (e-mail: robertsb@acoast.com).

Y. Ephraim and E. Dieguez are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: yephraim@gmu.edu).

Digital Object Identifier 10.1109/LSP.2006.871709

The techniques developed here are also applicable for parameter estimation of Markov arrival processes (MAPs). These processes are related to MMPPs and are often used in queuing theory. An EM algorithm for MAP parameter estimation was developed by Breuer in [1]. Estimation of the posterior state probability of an MMPP, which is an integral part of the EM algorithm, was studied in [8] using transformation of measures and the generalized Bayes' rule. Unlike our approach, the complexity resulting from evaluation of matrix exponential integrals was reduced in [8] using a discretization procedure.

The remainder of this letter is organized as follows. In Section II, we present a summary of the EM algorithm of Rydén [13]. In Section III, we present our main results. In Section IV, we present numeric results. Concluding comments are provided in Section V.

II. Background

In this section, we provide a summary of the EM algorithm of Rydén [13]. We follow the presentation of [13] and use similar notation. Uppercase letters are used to denote random variables and lowercase letters to denote their realizations. Let $\{X(t), t \geq 0\}$ denote the underlying continuous-time homogeneous Markov chain with state space, say, $\{1, \dots, r\}$. The number of states r constitutes the order of the chain, which is also referred to as the order of the MMPP. Let $Q = \{q_{ij}, i, j = 1, \dots, r\}$ denote the generator of the Markov chain, and define $q_i = -q_{ii}$. Let $\{N(t), t \geq 0\}$ denote the observed process. This is a variable-rate Poisson process with rate λ_i when $X(t) = i$. Let $\Lambda = \mathrm{diag}\{\lambda_1, \dots, \lambda_r\}$.

Let $T_k, k = 1, 2, \dots$ denote the time of the kth Poisson event. For simplicity, assume that a zeroth Poisson event occurs at $T_0 = 0$. We can now define the kth inter-event time as $Y_k = T_k - T_{k-1}$ for $k = 1, 2, \dots$. The zeroth event plays no further role and will not be counted by $\{N(t), t \geq 0\}$. Consider the embedded discrete-time Markov chain $\{X(T_k)\}_{k=1}^{\infty}$ obtained from sampling the continuous-time chain at the Poisson event times. It follows that $\{X(T_k), Y_k\}_{k=1}^{\infty}$ is a Markov renewal process with transition probability matrix $F(y) = \{F_{ij}(y)\}$, where $F_{ij}(y) = P(Y_k \leq y, X(t_{k-1} + y) = j | X(t_{k-1}) = i)$. The $r \times r$ transition density matrix, obtained from differentiation of $F_{ij}(y)$ w.r.t. y, is given by [5]

$$f(y) = \exp\left((Q - \Lambda)y\right)\Lambda. \tag{1}$$

The MMPP is assumed stationary and identifiable up to state permutations. The process $\{X(T_{k-1}), Y_k\}$ is stationary if the Markov chain $\{X(T_k)\}$ is initiated with its stationary distribution, say, π, which satisfies $\pi F(\infty) = \pi$. A unique stationary distribution exists if Q is irreducible and Λ contains at least one positive λ_i. The process is identifiable in the above sense if all rates $\{\lambda_i\}$ are distinct.

Let $\phi = \{Q, \Lambda\}$ denote the parameter of the stationary MMPP. Suppose that we observe the MMPP in an interval

$[0, T]$ and that the nth Poisson event (excluding the zeroth event) occurs at time T. The likelihood function of the sample path $\{N(u), 0 \le u \le T\}$ can be expressed in term of the likelihood function of the sequence $Y^n = \{Y_1, \ldots, Y_n\}$. The latter is given by

$$p(y^n) = \pi \prod_{k=1}^{n} f(y_k)\mathbf{1} \tag{2}$$

where y^n is a realization of Y^n, and $\mathbf{1}$ is an $r \times 1$ vector with all elements equal to one.

The EM algorithm of [13] attempts to find an ML estimator of ϕ from y^n. The algorithm was expressed in terms of the following statistics. Let $I(\cdot)$ denote the indicator function. For $i \neq j$, let

$$m_{ij} = \int_0^T \lim_{\Delta \downarrow 0} \frac{1}{\Delta} I\left(X(t-\Delta) = i, X(t) = j\right) dt \tag{3}$$

denote the number of jumps of $\{X(t)\}$ from state i to state j in $[0, T]$. Let

$$D_i = \int_0^T I\left(X(t) = i\right) dt \tag{4}$$

denote the time $\{X(t)\}$ spent in state i during $[0, T]$. Let t_k denote a realization of T_k, and let

$$n_i = \sum_{k=1}^{n} I\left(X(t_k) = i\right) \tag{5}$$

denote the number of events that occurred while $\{X(t)\}$ was in state i. Given an estimate of the parameter ϕ, say, $\phi = (Q, \Lambda)$, Rydén showed that a new estimate, say, $\{(\hat{q}_{ij}, \hat{\lambda}_i), i, j = 1, \ldots, r\}$, is obtained from

$$\hat{q}_{ij} = \frac{\hat{m}_{ij}}{\hat{D}_i}, \quad i \neq j \tag{6}$$

$$\hat{\lambda}_i = \frac{\hat{n}_i}{\hat{D}_i} \tag{7}$$

where $\hat{m}_{ij}$, $\hat{n}_i$, and $\hat{D}_i$ are conditional mean estimates under ϕ, of m_{ij}, n_i, and D_i, respectively, given $\{N(u), 0 \le u \le T\}$. For (3), the conditional mean results in the integral of the density of $\{X(t) = i, X(t) = j\}$ given $\{N(u), 0 \le u \le T\}$. For (4) and (5), the conditional means are the integrals of the appropriate conditional probabilities. Note that the initial distribution π is not estimated, since its estimation significantly complicates the algorithm, while π has negligible long-term effects on the likelihood function.

Using conditional independence of the increments $\{N(u), 0 \le u < t\}$ and $\{N(u), t \le u \le T\}$ given either $\{X(t-) = i, X(t) = j\}$ or $\{X(t) = j\}$, Rydén showed that the conditional mean estimates of (3) and (5), respectively, can be expressed as follows:

$$\begin{aligned}\hat{m}_{ij} = {} & \frac{q_{ij}}{P\left(N(u), 0 \le u < T\right)} \\ & \times \int_0^T P\left(N(u), 0 \le u < t, X(t-) = i\right) \\ & \times P\left(N(u), t \le u \le T | X(t) = j\right) dt \end{aligned} \tag{8}$$

$$\begin{aligned}\hat{n}_i = {} & \frac{1}{P\left(N(u), 0 \le u < T\right)} \\ & \times \sum_{k=1}^{n} P\left(N(u), 0 \le u < t_k, X(t_k) = i\right) \\ & \times P\left(N(u), t_k \le u \le T | X(t_k) = i\right) \end{aligned} \tag{9}$$

where $P(\cdot)$ denotes the density of the indicated sample path. The conditional mean estimate of (4) coincides with $\hat{m}_{ii}/q_{ii}$ in (8) when $X(t-) = i$ is replaced by $X(t) = i$. This substitution does not change the density since $\{X(t)\}$ is continuous in probability. Clearly, these conditional means depend on the forward–backward densities defined analogously to the case of hidden Markov processes (HMPs) with discrete-time Markov chains (see, e.g., [3]).

Rydén introduced the following forward–backward densities and showed that they can be recursively evaluated. The forward density under ϕ is given by

$$P(N(u), 0 \le u < t, X(t-) = i) = \left(\pi \prod_{k=1}^{N(t)} f(y_k)\right) \bar{F}\left(t - t_{N(t)}\right) \mathbf{1}_i \tag{10}$$

where $\mathbf{1}_i$ denotes an $r \times 1$ vector of which the ith element is one and all remaining elements are zero, and $\bar{F}(t)$ denotes an $r \times r$ transition matrix whose (i, j) element is the probability that when $X(0) = i$, then $X(t) = j$ and no events were seen in $(0, t]$. This matrix is given by

$$\bar{F}(t) = \exp\left((Q - \Lambda)t\right). \tag{11}$$

The backward density under ϕ is given by

$$\begin{aligned} & P\left(N(u), t \le u \le T | X(t) = j\right) \\ & \qquad = \mathbf{1}_j' f\left(t_{N(t)+1} - t\right) \left(\prod_{k=N(t)+2}^{n} f(y_k)\mathbf{1}\right) \end{aligned} \tag{12}$$

where $'$ denotes matrix transpose, and the convention $\prod_{l=\nu_1}^{\nu_2}(\cdot) = I$ is used whenever $\nu_1 > \nu_2$. To recursively evaluate the forward and backward densities, let $L(0) = \pi$ and $R(n+1) = \mathbf{1}$. Then, the bracketed term in (10) can be evaluated as $L(k) = L(k-1)f(y_k)$ for $k = 1, \ldots, n$. Similarly, the bracketed term in (12) can be evaluated as $R(k) = f(y_k)R(k+1)$ for $k = n, \ldots, 1$.

The forward–backward densities tend toward zero or infinity exponentially fast as $n \to \infty$. This observation follows from an ergodic theorem for the sample entropy of the observation process (see Leroux [9]). Thus, the forward–backward recursions may not provide the correct values of the densities when the computer's dynamic range is limited. This phenomenon, commonly referred to as the "numeric instability" of the forward–backward recursions, is well known for HMPs with discrete-time Markov chains [3]. The recursions were thus implemented in [13] using customized floating-point software. Here we follow on a suggestion in [13] and develop a scaling procedure for the forward–backward recursions, which can then be implemented using standard routines in Matlab.

Implementation of (8) and (9) using (1) and (11) requires evaluation of matrix exponential integrals. In [13], the matrices involved were first diagonalized and then integrated. Since these matrices are not symmetric, inversion of matrices is required. Another approach, due to Klemm *et al.* [7], required using a randomization expansion for the matrix exponentials and truncated series expansion in incomplete β-functions. Here we invoke a result from control theory, due to Van Loan [15], and compare it with Rydén's approach.

III. Main Results

In this section, we present our main results. We address the scaling of the forward–backward recursions and the evaluation of matrix exponential integrals.

A. Scaling of Forward–Backward Recursions

The proposed scaling procedure is motivated by the form of $\hat{m}_{ij}$. The approach is equally applicable to the expressions for $\hat{D}_i$ and $\hat{n}_i$. To evaluate $\hat{m}_{ij}$, we first substitute (10) and (12) into (8), as was done in [13]. Then, we perform the integration over disjoint intervals of lengths $\{y_k\}$ and replace $P(N(u), 0 \leq u < T)$ by $p(y^n) = \prod_{l=1}^{n} p(y_l|y^{l-1})$, where $p(y_1|y_0) = p(y_1)$. Letting $c_k = p(y_k|y^{k-1})$, $k = 1, \ldots, n$, we obtain

$$\hat{m}_{ij} = \sum_{k=1}^{n} \frac{q_{ij}}{c_k} \left(\pi \prod_{l=1}^{k-1} \frac{f(y_l)}{c_l} \right) \times \int_{t_{k-1}}^{t_k-} \bar{F}(t - t_{k-1}) \mathbf{1}_i \mathbf{1}_j' f(t_k - t) dt \left(\prod_{l=k+1}^{n} \frac{f(y_l)}{c_l} \mathbf{1} \right). \quad (13)$$

Now, we *redefine* $L(k)$ and $R(k)$ to allow recursive evaluation of the bracketed products in (13), respectively. Specifically, we define

$$L(k) = \pi \prod_{l=1}^{k} \frac{f(y_l)}{c_l}, \quad k = 1, \ldots, n$$
$$R(k) = \prod_{l=k}^{n} \frac{f(y_l)}{c_l} \mathbf{1}, \quad k = n, \ldots, 1 \quad (14)$$

with $L(0) = \pi$ and $R(n+1) = \mathbf{1}$. From (2), we have

$$c_k = \pi \prod_{l=1}^{k-1} \frac{f(y_l)}{c_l} f(y_k) \mathbf{1} = L(k-1) f(y_k) \mathbf{1}. \quad (15)$$

Hence, using (14) and (15), we have

$$L(k) = \frac{L(k-1) f(y_k)}{L(k-1) f(y_k) \mathbf{1}}, \quad R(k) = \frac{f(y_k) R(k+1)}{L(k-1) f(y_k) \mathbf{1}}. \quad (16)$$

Using (2), it can be seen that the jth element of $L(k)$, say, $L(k)_j$, is equal to $L(k)_j = \Pr(X(t_k) = j|y^k)$. Thus, the recursion for $L(k)$ in (16) is a forward recursion for the probability of the state at time t_k given the observations $\{N(t), 0 \leq t \leq t_k\}$. As such, the value of $L(k)_j$ cannot exceed one. To interpret $R(k)$, let $y_k^n = \{y_k, \ldots, y_n\}$. For the jth element of $R(k)$, we have from (2) and (14) that

$$R(k)_j = \frac{p\left(y_k^n | X(t_{k-1}) = j\right)}{p\left(y_k^n | y^{k-1}\right)}. \quad (17)$$

Using conditional independence similarly to that applied in (9), we obtain

$$\begin{aligned} R(k)_j &= \frac{p\left(y_k^n | X(t_{k-1}) = j\right) p(y^{k-1})}{\sum_{i=1}^{r} p\left(y_k^n | X(t_{k-1}) = i\right) p\left(y^{k-1}, X(t_{k-1}) = i\right)} \\ &= \frac{p\left(y_k^n | X(t_{k-1}) = j\right)}{\sum_{i=1}^{r} p\left(y_k^n | X(t_{k-1}) = i\right) \Pr\left(X(t_{k-1}) = i | y^{k-1}\right)} \\ &< \frac{1}{\Pr\left(X(t_{k-1}) = j | y^{k-1}\right)} = \frac{1}{L(k-1)_j} \end{aligned} \quad (18)$$

provided that $L(k-1)_j$ is bounded away from zero. Thus, the scaled $R(k)_j$ is upper bounded, and $R(k)_j L(k-1)_j < 1$.

The scaled recursions for $L(k)$ and $R(k)$ in (16) are equivalent to the scaled recursions developed by Levinson *et al.* [10] for HMPs with discrete-time Markov chains. The scaling procedure in [10], however, is algorithmic, and hence, its cumulative effects are not immediately evident. Devijver [2] correctly argued that, at the k-step, the embedded procedure of [10] effectively scales the forward recursion by $p(y^k)$ and the backward recursion by $p(y_{k+1}^n | y^k)$. Since $p(y^k) = \prod_{l=1}^{k} c_l$ and $p(y_{k+1}^n | y^k) = \prod_{l=k+1}^{n} c_l$, the scale factors for the forward and backward recursions in the embedded scheme of [10] coincide with the explicit factors of $\{c_k\}$ used here. The approach taken here justifies the backward scaling factor proposed in [10] since it naturally follows from factorization of $p(y^n)$. The coincidence of the scaling factors for HMPs and MMPPs of course is not surprising if one considers their state space representations (see, e.g., [3] and [8]). Some additional comments on the forward–backward scaling procedure of [10] can be found in [3].

To complete the description of the algorithm using the recursions (16), we now comment on the evaluation of $\hat{n}_i$ and $\hat{D}_i$. Let $\hat{n}$ denote the $r \times 1$ vector with the ith element given by $\hat{n}_i$. From (8) and (9), it is easy to see that

$$\hat{n} = \sum_{k=1}^{n} L(k)' \odot R(k+1) \quad (19)$$

where $\odot$ denotes term-by-term multiplication of the two matrices. Furthermore, $\hat{D}_i = \hat{m}_{ii}/q_{ii}$. Finally, the log-likelihood of the MMPP is given by

$$\log p(y^n) = \sum_{t=1}^{n} \log(y_t | y^{t-1}) = \sum_{t=1}^{n} \log c_t. \quad (20)$$

B. Integrals of Matrix Exponentials

We now turn to evaluation of the integral of matrix exponentials in (13). Let $\hat{m}$ denote an $r \times r$ matrix whose entries are given by $\{\hat{m}_{ij}\}$. The transpose of this matrix may be written as

$$\hat{m}' = Q' \odot \sum_{k=1}^{n} \frac{1}{c_k} \int_{t_{k-1}}^{t_k-} f(t_k - t) R(k+1) L(k-1) \bar{F}(t - t_{k-1}) dt. \quad (21)$$

Substituting (1) and (11) into (21), and using $y_k = t_k - t_{k-1}$, yields

$$\hat{m} = Q \odot \sum_{k=1}^{n} \frac{\mathcal{I}_k'}{c_k}$$

where

$$\mathcal{I}_k = \int_{0}^{y_k} e^{(Q-\Lambda)(y_k - y)} \Lambda R(k+1) L(k-1) e^{(Q-\Lambda)y} dy. \quad (22)$$

TABLE I
RATIOS OF EXECUTION TIMES USING VAN LOAN'S AND RYDEN'S METHODS FOR THE DEFINITE INTEGRAL OF MATRIX EXPONENTIAL

	r = 2	r = 3	r = 4
Calculation of I_k only	20.1	94.8	301.0
Overall algorithm	10.0	50.5	198.9

The integral (22) can be efficiently evaluated using an approach developed by Van Loan [15, Theorem 1]. Applying this approach to our problem, we define the $2r \times 2r$ block-triangular matrix

$$C_k = \begin{bmatrix} Q - \Lambda & \Lambda R(k+1)L(k-1) \\ 0 & Q - \Lambda \end{bmatrix}. \tag{23}$$

It follows that $\mathcal{I}_k$ is the $r \times r$ upper-right block of the matrix $e^{C_k y_k}$. This integral evaluation, as well as evaluation of other related integrals, was shown in [15] to follow from equating terms in the equation $(d/dy)e^{C_k y} = C_k e^{C_k y}$. Thus, evaluation of certain integrals of matrix exponentials can be replaced by evaluation of a matrix exponential of higher dimension. The latter can be efficiently performed using diagonal Padé's approximation with repeated squaring, as recommended in [15]. This approximation is used by the Matlab function "expm." Thus, the desired integral may be compactly evaluated using a single readily available Matlab routine. In contrast, Rydén [13] evaluated this integral by first diagonalizing the exponent matrix and then integrating over exponents of the eigenvalues. Inversion of matrices was required in this process.

IV. Numerical Results

We have implemented the re-estimation formulas (6), (7) using the scaled forward–backward recursions (16) and Van Loan's approach for evaluation of the integrals of matrix exponentials. This setup was compared to Rydéns original EM algorithm in terms of numeric accuracy and computational speed. All routines were implemented using only standard features of the Matlab version 7.0 programming language installed on a Pentium M processor running the Windows XP operating system.

We have simulated MMPPs with the same parameter values as those given in [13]. For $n \leq 10$, no numeric instability of the unscaled forward–backward recursions was apparent. Thus, we have used this small value of n to verify that the scaled forward–backward recursions and Van Loan's approach were properly implemented. For the larger values of $n = 5000$ and $n = 30\,000$ used in [13], the numerical instability of the unscaled forward–backward recursions was evident. For these large values of n, we found that the scaled forward–backward recursions and Van Loan's approach implemented here resulted in similar parameter estimates as those obtained by Rydén [13] using customized floating-point software and matrix diagonalization. The specialized software, however, is not readily available in standard Matlab. The slight deviation between ours and Rydén's parameter estimates was due to the different sample paths of the MMPP used in both cases.

In implementing Van Loan's approach, we have used the Matlab function "expm." For comparison, we have implemented Rydén's approach using the Matlab function "eig" for eigen-decomposition of a matrix. In both cases, the scaled forward–backward recursions were used. Results were obtained for $r = 2, 3, 4$. The MMPP parameter for $r = 2$ and $r = 3$ were those used in cases 2A and 3A of [13], respectively. For $r = 4$, $\{\lambda_i\}$ and $\{q_{ij}\}_{i \neq j}$ were generated as independent uniform random variables over [0, 1]. The computational speedup factors are summarized in Table I. These results show a clear performance advantage, which increases with r. It should be noted, however, that computational performance may be heavily influenced by the particulars of the implementation, programming language, operating system, and hardware used. The speedup factors reported in Table I were validated using a second independently implemented Matlab (version 7.0) code running on an Athlon XP 2700 personal computer. With the exception of the forward–backward recursions, vectorization was used throughout that code.

V. Comments

We have addressed scaling of the forward–backward recursions and evaluation of integrals of matrix exponentials. Both are crucial computational aspects of parameter estimation of MMPPs. The scaling procedure developed here turned out to be equivalent to that commonly used in HMPs with discrete-time Markov chains. Our procedure, however, is intrinsic to the re-estimation formulas. Thus, it clarifies in a straightforward manner several aspects of the original scaling procedure. For the integrals of matrix exponentials, we have invoked a result from control theory that facilitated the evaluation of these integrals. The use of Van Loan's result provided significant speedup of the algorithm.

References

[1] L. Breuer, "An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure," *Ann. Oper. Res.*, vol. 112, pp. 123–138, 2002.

[2] P. A. Devijver, "Baum's forward-backward algorithm revisited," *Pattern Recognit. Lett.*, vol. 3, pp. 369–373, 1985.

[3] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1518–1569, Jun. 2002.

[4] W. Fischer and K. Meier-Hellstern, "The Markov-modulated Poisson process (MMPP) cookbook," *Perform. Eval.*, vol. 18, pp. 149–171, 1992.

[5] D. S. Freed and L. A. Shepp, "A Poisson process whose rate is a hidden Markov chain," *Adv. Appl. Prob.*, vol. 14, pp. 21–36, 1982.

[6] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplier performance," *IEEE J. Sel. Areas Commun.*, vol. SAC-4, no. 6, pp. 856–868, Sep. 1986.

[7] A. Klemm, C. Lindemann, and M. Lohmann, "Modeling IP traffic using the batch Markovian arrival process," *Perform. Eval.*, vol. 54, pp. 149–173, 2003.

[8] V. Krishnamurthy and S. Dey, "Reduced spatio-temporal complexity MMPP and image-based tracking filters for maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1277–1291, Oct. 2003.

[9] B. G. Leroux and M. J. Puterman, "Maximal-penalized-likelihood estimation for independent and Markov dependent mixture models," *Biometrics*, vol. 48, pp. 545–558, Jun. 1992.

[10] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, Apr. 1983.

[11] D. W.-C. Miao, Markov Modulated Poisson Processing in Credit Risk Modeling Univ. Oxford, Oxford, U.K., 2005, Transfer Rep..

[12] T. Rydén, "Parameter estimation for Markov modulated Poisson processes," *Commun. Statist. Stoch. Models*, vol. 10, no. 4, pp. 795–829, 1994.

[13] ——, "An EM algorithm for estimation in Markov-modulated Poisson processes," *Comput. Stat. Data Anal.*, vol. 21, pp. 431–447, 1996.

[14] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*. New York: Springer-Verlag, 1991.

[15] C. F. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Trans. Autom. Control*, vol. AC-23, no. 3, pp. 395–404, Mar. 1978.