18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Saddle Point Method for JPDA and Related Filters

Roy Streit
Metron, Inc.
Reston, VA 20190 U.S.A.
r.streit@ieee.org

Abstract – The Bayes posterior probability distribution of many multitarget tracking filters can be written in terms of the mixed derivatives of an appropriately defined generating function. Using the Cauchy Residue Theorem for several complex variables, it is shown that these derivatives can be approximated using the saddle point method, a classical technique in analytic combinatorics. The method gives approximate weights for the particles of sequential Monte Carlo filter implementations. For JPDA the approximate particle weights are seen to be accurate to within one or two percent of the exact weights, after a global scaling factor is applied. The saddle point method has polynomial computational complexity in the number of targets and measurements, assuming that a bounded number of iterations are needed to find the saddle point.

Keywords: JPDA, particle filter, analytic combinatorics, saddle point method, Cauchy residue theorem.

1 Introduction

The purpose of this paper is to show that NP-hard measurement-to-target assignment problems that arise in many multitarget tracking filters can be circumvented by using a classical approximation technique called the saddle point method. It is applicable to filters whose Bayes posterior probability distribution can be written in terms of the mixed first-order derivatives of a generating function. The main focus of the paper is the joint probabilistic data association (JPDA) filter, especially sequential Monte Carlo (SMC) particle implementations. For discussion of particle JPDA implementation, see [6].

It is demonstrated that particle weights for the JPDA filter can be computed to within one or two percent of the exact particle weights, after a global scaling factor is applied. The scale factor is irrelevant to the SMC implementations of the JPDA filter.

The computational complexity of exact JPDA is known to be NP-hard. This is true even if clutter is not present. For detailed discussions, see [7, 8, 9].

It is shown here that the saddle point method for particle weight calculation is polynomial in the numbers of targets and measurements. Thus, for sufficiently large problems, it will be faster than exact methods.

1.1 Organization of the paper

Section 2 introduces the generating functional for the JPDA approach to multitarget tracking. Superposition of clutter and target-originated measurements is the main theoretical model. An important feature that distinguishes it from some other multitarget filters is that the target-originated measurements are superposed, not the targets themselves. Thus each target retains its own state space and has its own test function.

Section 3 exploits the fact that the probability generating functional (PGFL) is an analytic function of linear functionals. Functionals in this class can be reduced to multivariate analytic functions via Dirac delta function trains [10]. Particle weights are the mixed ordinary derivatives of the analytic function.

Section 4 formulates the mixed derivatives in terms of the Cauchy Residue Theorem of several complex variables. (This theorem is best known for the special case of one complex variable.) The Cauchy integrals, and hence the particle weights, are approximated by the saddle point method – a method often used in asymptotic analysis. The text [1] contains a wide-ranging discussion of analytic combinatorics and the saddle point method.

The didactic discussion in Section 5 entitled "Why Saddle Points?" provides an intuitive picture of why and how the saddle point method works.

Section 6 gives examples with Monte Carlo simulated data to study the accuracy of the approximate saddle point weights for JPDA.

Section 7 discusses the computational complexity of the saddle point method for particle JPDA.

Section 8 discusses the utility of the saddle point method for other multitarget tracking filters.

Section 9 gives concluding remarks, including comments on MCMC methods for approximate JPDA.

The Appendix gives a succinct derivation of the matrix permanent by the Cauchy Residue Theorem. The permanent arises in particle weight calculations for the special case of JPDA in the absence of clutter [9].

2 Analytic Combinatorics for JPDA

The JPDA approach to multitarget tracking maintains a different state space for different targets. These spaces are

commonly taken to be copies of the same space, but this is unnecessary. The target motion models and measurement likelihood functions can also be different for different targets. JPDA assumes that the number of targets, N, is known and fixed. Measurements, in contrast, are assumed to be taken by the same sensor and, thus, to lie in the same measurement space.

Here, to simplify notation, the usual scan time index is suppressed. Since JPDA is so well understood, this should not cause confusion. For simplicity, target state spaces are taken to be copies of the same state space, $S \subset \mathbb{R}^{\dim(S)}$. The measurement space is denoted by $Y \subset \mathbb{R}^{\dim(Y)}$.

The PGFL for the PDA filter is discussed in [11].

2.1 PGFL of the JPDA Filter

Because of these modeling choices, the PGFL for JPDA has $N+1$ test functions, one for each target and one for measurements. Test functions for targets and measurements are denoted by $h_1(x),\ldots,h_N(x)$ and $g(y)$, respectively. The PGFL of the target-originated measurement process for target n is denoted by $\Psi_{\mathrm{BMD}(n)}(h_n, g)$. The PGFL of the clutter process is denoted by $\Psi_C(g)$. JPDA assumes all $N+1$ processes to be mutually independent, and that all measurements are superposed in the measurement space. Hence, the PGFL of JPDA is the product of the PGFLs of the superposed processes:

$$\Psi_{\mathrm{JPDA}}(h_1,\ldots,h_N,g) = \Psi_{\mathrm{C}}(g)\prod_{n=1}^{N}\Psi_{\mathrm{BMD}(n)}(h_n,g). \quad (1)$$

Specific forms for each term are derived from the additional JPDA assumptions. In tracking problems the PGFLs of the superposed processes are time dependent.

2.2 Clutter Model

Clutter points are assumed independent and identically distributed. Let $p_\Lambda(y)$ denote the probability density function (PDF), and let Λ be the expected number of clutter points in the sensor region of regard. With the Poisson modeling assumption [13] that is typically used in JPDA, the clutter PGFL is

$$\Psi_{\mathrm{C}}(g) = \exp\left(-\Lambda + \Lambda\int_Y g(y)p_\Lambda(y)\,dy\right). \quad (2)$$

Non-Poisson clutter models are easily used instead. As a check, note that $\Psi_{\mathrm{C}}(1) = 1$.

2.3 Target-Originated Measurement Model

The predicted target PDF for target n is

$$\mu_n(x) = \int_S p_n(x \mid x')\,p_n^-(x')\,dx', \quad (3)$$

where $p_n^-(\cdot)$ is the PDF of target state at the previous scan that has been information updated with all measurements up to that time, and $p_n(x \mid \cdot)$ is the Markov transition function (i.e., target motion model) from the previous scan to the current scan.

Let $P_n^D(x)$ denote the probability of detection (at the current scan) for target n when it is in state x, and let $a_n^D(x) = 1 - P_n^D(x)$ and $b_n^D(x) = P_n^D(x)$. The probability generating function (PGF) of the number of target-originated measurements is

$$G_n(z \mid x) = a_n(x) + b_n(x)\,z\,. \quad (4)$$

The PGF is linear in the dummy variable z because it incorporates the "at most one measurement per target" rule. It follows (see [11]) that

$$\begin{aligned}&\Psi_{\mathrm{BMD}(n)}(h_n, g)\\ &= \int_S h_n(x)\mu_n(x)G_n\left(\int_Y g(y)p_n(y \mid x)\,dy\right)dx \quad (5)\\ &= \int_S h_n(x)\mu_n(x)\left(a_n^D(x) + b_n^D(x)\int_Y g(y)p_n(y \mid x)\,dy\right)dx.\end{aligned}$$

is the joint PGFL of the target and the target-originated measurement process for target n. As a check, note that $\Psi_{\mathrm{BMD}(n)}(1,1) = 1$. Since $\Psi_{\mathrm{BMD}(n)}$ is linear in h_n, it is referred to simply as the PGFL of the target-originated measurement model.

The PGFL for JPDA is found by substituting the clutter PGFL (2) and the PGFLs for the target-originated measurement processes (5) into the joint expression (1).

2.4 JPDA Marginal Target

JPDA works not in the full multitarget state space S^n but in the n marginal distributions. To close the Bayesian recursion, it approximates each of the marginal distributions in some fashion. This same marginalization approach is used here using an SMC particle implementation.

The marginal distribution for target n is the integral of the joint JPDA distribution over all target states except for target n. The PGFL of the marginal distribution is found by setting $h_j(\cdot) = 1$, $j \neq n$, in the joint PGFL (1), that is,

$$\begin{aligned}&\Psi_{\mathrm{JPDA}(n)}(h_n, g)\\ &\quad= \Psi_{\mathrm{C}}(g)\,\Psi_{\mathrm{BMD}(n)}(h_n, g)\prod_{j=1,\ j\neq n}^{N}\Psi_{\mathrm{BMD}(j)}(1, g). \quad (6)\end{aligned}$$

The PGFL for the correlated version of JPDA for, say two targets, is found by setting all but two of the h_js to one. The topic of central interest in this paper is evaluating the weight of an individual particle at state $x \in S$ using (6).

Given a set of measurements $\{y_1,\ldots,y_M\} \subset Y$, the weight of a particle is the sum of the likelihoods of the feasible measurement to target assignments, as seen through the marginal JPDA distribution. As is well known, this is an NP-hard calculation because the number of

feasible assignments grows rapidly in the number of targets and measurements.

3 Analytic Function for JPDA

The first step in the reduction (projection) a PGFL to a form suited to analysis and numerical calculation is to "kill the functionals" by substituting weighted sum of Dirac delta functions. A more general discussion and detailed methods can be found in [8].

The reduction cannot be performed on all functionals but it can be done here. By inspection the PGFLs (2) and (5) are analytic functions of linear functionals. In other words, test functions enter the PGFL as weighted integrals of the form $\int h(x)w(x)\,dx$ and not, e.g., as h^2 or e^h. The nonlinearities arise from the analytic functions involved. In the case of JPDA, these are exponentials and products of linear functions. The arguments of these analytic functions are linear functionals.

It follows that analytic functions of linear functionals can be evaluated at weighted sums of Dirac delta functions. This reduction can be proved by taking the limit of the PGFL acting on an appropriately defined sequence of test functions. It is important to take the limit *after* evaluating the integrals of the test sequence – this avoids the error of moving the limit inside the integral.

3.1 Delta Function Trains

Let the measurements $\{y_m : m = 1,\ldots,M\}$ be given. For ease of exposition, discussion of the special case $M = 0$ is omitted. For later use, define the positive constants and functions

$$\begin{aligned} a_0 &= -\Lambda \\ b_{n0} &\equiv b_{n0}(x_n) = \mu_n(x_n)\, a_n^D(x_n) \\ c_{n0} &= \int_S \mu_n(x)\, a_n^D(x)\,dx = \int_S b_{n0}(x)\,dx \\ a_m &= \Lambda p_\Lambda(y_m) \\ b_{nm} &\equiv b_{nm}(x_n) = \mu_n(x_n)\, b_n^D(x_n)\, p_n(y_m \mid x_n) \\ c_{nm} &= \int_S \mu_n(x)\, b_n^D(x)\, p_n(y_m \mid x)\,dx = \int_S b_{nm}(x)\,dx. \end{aligned} \tag{7}$$

Let $\beta = (\beta_1,\ldots,\beta_M) \in \mathbb{C}^M$. Define the weighted Dirac delta function train (or equivalently, a test sequence for it)

$$g(y) = \sum_{m=1}^M \beta_m\, \delta(y - y_m). \tag{8}$$

Substituting (8) into the clutter PGFL (2) gives

$$\begin{aligned} &\Psi_C\left(\sum_{m=1}^M \beta_m\, \delta(y - y_m)\right) \\ &\quad = \exp\left(-\Lambda + \Lambda \sum_{m=1}^M \beta_m\, p_\Lambda(y_m)\right) \\ &\quad = \exp\left(a_0 + \sum_{m=1}^M a_m \beta_m\right). \end{aligned} \tag{9}$$

Note that $\Psi_C(\beta)$ is an analytic function of the M complex variables β.

The PGFL (5) for target n is linear in $h_n(\cdot)$, hence the Dirac delta function train has only one term

$$h_n(x) = \alpha_n \delta(x - x_n), \quad \alpha_n \in \mathbb{C}. \tag{10}$$

Substituting (10) and (8) into (5) gives

$$\begin{aligned} &\Psi_{\mathrm{BMD}(n)}\left(\alpha_n \delta(x - x_n),\, \sum_{m=1}^M \beta_m\, \delta(y - y_m)\right) \\ &\quad = \alpha_n \mu_n(x_n)\left(a_n^D(x_n) + b_n^D(x_n) \sum_{m=1}^M \beta_m\, p_n(y_m \mid x_n)\right) \\ &\quad = \alpha_n \left(b_{n0} + \sum_{m=1}^M b_{nm} \beta_m\right). \end{aligned} \tag{11}$$

The PGFL of the marginal distribution of target n is

$$\begin{aligned} &\Psi_{\mathrm{BMD}(n)}\left(1,\, \sum_{m=1}^M \beta_m\, \delta(y - y_m)\right) \\ &\quad = \int_S \mu_n(x)\left(a_n^D(x) + b_n^D(x) \sum_{m=1}^M \beta_m\, p_n(y_m \mid x)\right) dx \\ &\quad = c_{n0} + \sum_{m=1}^M c_{nm} \beta_m. \end{aligned} \tag{12}$$

Substituting (9), (11), and (12) into (6) gives

$$\Psi_{\mathrm{JPDA}(n)}(\alpha_n, \beta) \equiv \alpha_n \Psi_{\mathrm{JPDA}(n)}(\beta), \tag{13}$$

where

$$\begin{aligned} \Psi_{\mathrm{JPDA}(n)}(\beta) &\equiv \exp\left(a_0 + \sum_{m=1}^M a_m \beta_m\right) \\ &\quad \times \left(b_{n0} + \sum_{m=1}^M b_{nm} \beta_m\right) \prod_{j=1,\, j\neq n}^N \left(c_{j0} + \sum_{m=1}^M c_{jm} \beta_m\right) \end{aligned} \tag{14}$$

is an analytic function of M complex variables β.

The marginal of the Bayes posterior PDF for target n at $x \in S$ is proportional to the cross-derivative of (13) evaluated at zero, that is, to the mixed derivative of order one with respect to each of the $M+1$ variables (evaluated at zero). Only the derivative of (13) with respect to α_n is trivial – it is the coefficient given by (14).

3.2 Particle Weights as Derivatives

The mixed first order derivative of (14) with respect to β at $\beta = 0 \in \mathbb{C}^M$ is proportional to the marginal of the Bayes posterior PDF for target n at $x \in S$. Its numerical value at x is therefore proportional to the weight of a particle at x. The dependence of these derivatives on x is due to the coefficients $b_{n0}(x)$ and $b_{nm}(x)$ in (7).

Expression (14) drives the computational complexity of JPDA through the complexity of its cross-derivative. The derivative can be found symbolically, and evaluated to find the numerical value of a particle at x. Alternatively, automatic differentiation (AD) can be employed to find the exact weight (via the chain rule) without first finding the symbolic derivative [12, 15]. However, both methods are exact, hence their computational complexity is exponential in problem size.

The root of the difficulty is that there is a one to one correspondence between the terms in the cross-derivative and the feasible measurement-to-target assignments. It therefore follows that approximation is the only way to upper bound the computational complexity of JPDA.

The analytic function (14) is closely related to matrix permanents. The connection was first noted in [9]. The Appendix gives a simple derivation using the Cauchy Residue Theorem.

4 Saddle Point Approximation

The proposed method has two essential elements. The first is to replace the cross-derivative of the analytic function $\Psi_{\text{JPDA}(n)}(\beta)$ by an exactly equivalent integral expression. This integral is the Cauchy Residue Theorem in several complex variables. It is reviewed in the first subsection below.

The Cauchy integral is then approximated using another classical tool called the saddle point method (sometimes called the method of stationary phase). The saddle point approximation requires finding the saddle point of the integrand. It is derived in the second subsection.

Readers are referred to Section 5 for insight into why it is important to find the saddle point is important. It also shows that the saddle point approximation is a quadratic fit to the integrand at the saddle point.

The discussion and derivation are presented in a form tailored to the needs of JPDA, but the saddle point method is not limited to JPDA. For discussion of the method, see the texts [1, 3, 5] and the introductory paper [16].

4.1 Multivariate Cauchy Residue Theorem

The classic form of Cauchy's theorem for one complex variable (see, e.g., [4]) is (in a traditional notation)

$$f(z) = \frac{1}{2\pi i} \oint_C \frac{f(\zeta)}{\zeta - z}\, d\zeta\,, \tag{15}$$

where $f : \mathbb{C} \to \mathbb{C}$ is analytic inside and on a simple closed contour C that contains the point $z \in \mathbb{C}$ in its interior. For functions $f : \mathbb{C}^m \to \mathbb{C}$ of several complex variables, (15) generalizes to the multiple integral [2]

$$\begin{aligned} &f(z_1,...,z_m) \\ &= \frac{1}{(2\pi i)^m} \oint_{C_1} \cdots \oint_{C_m} \frac{f(\zeta_1, \ldots, \zeta_m)}{(\zeta_1 - z_1)\cdots(\zeta_m - z_m)} d\zeta_1 \cdots d\zeta_m, \end{aligned} \tag{16}$$

where $z_j \in \mathbb{C}$ is interior to contour C_j.

The integrals in (16) can be done in any order, a result that follows from the Fubini-Tonelli theorem for real variables, together with the surprising fact (Hartog's Theorem) that a (complex) function analytic in each variable is analytic in all variables jointly [2].

Because f is analytic, derivatives can be found by differentiating under the integral. Thus, the derivative of (16) of order $k_1,...,k_m$ with respect to $z_1,...,z_m$ is

$$\begin{aligned} &f^{(k_1,\cdots,k_m)}(z_1,...,z_m) = \frac{k_1! \cdots k_m!}{(2\pi i)^m} \\ &\times \oint_{C_1} \cdots \oint_{C_m} \frac{f(\zeta_1, \ldots, \zeta_m)}{(\zeta_1 - z_1)^{k_1+1} \cdots (\zeta_m - z_m)^{k_m+1}}\, d\zeta_1 \cdots d\zeta_m. \end{aligned} \tag{17}$$

See [1, 2] for further discussion of the general topic.

4.2 Derivation of the Approximation

The weight of a particle at x for target n is proportional to the cross-derivative:

$$I \equiv I_n(x) = \frac{d^M}{d\beta_1 \cdots d\beta_M} \Psi_{\text{JPDA}(n)}(\beta)\Big|_{\beta_1 = \cdots = \beta_M = 0}\,. \tag{18}$$

Using the Cauchy integral (17) with $k_1 = \cdots = k_M = 1$, $a_1 = \cdots = a_M = 0 \in \mathbb{C}$, and circular contours $C(r_j)$ centered at zero with radii $r_j > 0$ gives

$$I = \frac{1}{(2\pi i)^M} \oint_{C(r_1)} \cdots \oint_{C(r_M)} \Psi(\beta_1,\cdots,\beta_M)\, \frac{d\beta_1 \cdots d\beta_M}{\beta_1 \cdots \beta_M}\,, \tag{19}$$

where $\Psi(\beta) \equiv \Psi(\beta_1,\cdots,\beta_M)$ is

$$\Psi(\beta) = \frac{\Psi_{\text{JPDA}(n)}(\beta_1,\cdots,\beta_M)}{\beta_1 \cdots \beta_M}. \tag{20}$$

The integral (19) evaluates to the same number for all choices of radii $r_j > 0$. The saddle point approximation is a shrewd way to choose these radii.

Note that $\Psi_{\text{JPDA}(n)}(\beta)$ is an entire function in each variable β_j, and its Taylor series has only nonnegative coefficients when expanded about the origin. This property characterizes an important subclass of analytic functions in complex analysis. Generating functions (GFs) fall into this class. As discussed in [1], Pringsheim's Theorem enables the search for dominant singularities to be restricted to the positive real line, a fact that is used here to determine the radii of the contours.

Using a Taylor series about a point $0 \neq \hat{\beta} \in \mathbb{C}^M$ gives

$$\begin{aligned} \Psi(\beta) &= e^{\phi(\beta)} \\ &= e^{\phi(\hat{\beta}) + \nabla_\beta \phi(\hat{\beta})(\beta - \hat{\beta}) + \frac{1}{2}(\beta - \hat{\beta})^T H(\hat{\beta})(\beta - \hat{\beta}) + h.o.t.}, \end{aligned} \tag{21}$$

where $h.o.t.$ stands for "higher order terms" and $H(\hat{\beta})$ is the $M \times M$ hessian matrix of ϕ,

$$H(\hat{\beta}) = \nabla_{\beta\beta}\, \phi(\beta)\Big|_{\beta=\hat{\beta}} = \nabla_{\beta\beta}\, \phi(\hat{\beta})\,. \tag{22}$$

Choose $\hat{\beta}$ so that

$$\nabla_\beta \phi(\beta)\Big|_{\beta=\hat{\beta}} = 0\,. \tag{23}$$

Such a point is called a saddle point of Ψ.

Let $\Psi(r)$ be the restriction of $\Psi(\cdot)$ to the positive real line. The saddle point is a positive vector that satisfies

$$\hat{r} = (\hat{r}_1,...,\hat{r}_M) = \underset{r_1>0,\,...,\,r_M>0}{\arg\min}\ \Psi(r_1,...,r_M)\,, \qquad (24)$$

To see this, a coordinate relaxation procedure is used. Fix the value of all variables except one, say $\beta_j = r_j e^{i\theta_j}$, to find a single-variable function, denoted $\tilde{\phi}(\beta_j)$. It has nonnegative coefficients when expanded about zero (*cf.* (7)), from which it follows, as shown in [1], that on the real line $\tilde{\phi}(r_j)$ is a convex function and the equation $\nabla_{r_j}\tilde{\phi}(r_j) = 0$ has at most one positive root. In this case the function $\Psi_{\text{JPDA}(n)}$ is entire but not a polynomial, from which it follows from the proof in [1] that $\tilde{\phi}(r_j)$ has exactly one positive root. This result holds for all (feasible) values of the other variables. Let an initial feasible point $r^{(0)} = (r_1^{(0)},...,r_M^{(0)}) \in \mathbb{R}^M_{>0}$ be specified. Note that $\Psi(r^{(0)}) < \infty$. Proceed by minimizing the convex function $\tilde{\phi}(r_j)$ with respect to r_j, keeping the other variables fixed. Applying this procedure to all M variables one at a time, in turn, and cycling through all of them repeatedly, yields a sequence of points with decreasing values of Ψ which converges to the saddle point satisfying (24).

Choose the contours $C(r_j)$ in (19) to have radii equal to the components saddle point vector. Let $\beta_j = \hat{r}_j e^{i\theta_j}$. Then $d\beta_j = i\hat{r}_j e^{i\theta_j}\, d\theta_j$ and the integral (19) becomes

$$I = \frac{1}{(2\pi)^M}\int_{-\pi}^{\pi}\cdots\int_{-\pi}^{\pi}\Psi(\hat{r}e^{i\theta})\,d\theta_1\cdots d\theta_M\,, \qquad (25)$$

where $\beta = \hat{r}e^{i\theta} = (\hat{r}_1 e^{i\theta_1},\cdots,\hat{r}_M e^{i\theta_M})$. Substituting β into (21) and dropping the higher order terms gives the first approximation

$$\Psi(\hat{r}e^{i\theta}) \cong \Psi(\hat{r})\,\exp\Big(\tfrac{1}{2}[\hat{r}(e^{i\theta}-1)]^T H(\hat{r})[\hat{r}(e^{i\theta}-1)]\Big). \qquad (26)$$

By Taylor series, $e^{i\theta}-1 = i\theta - \frac{1}{2}\theta^2 + \cdots$. Retaining only the linear term and substituting it into (26) gives a second approximation

$$\Psi(\hat{r}e^{i\theta}) \cong \Psi(\hat{r})\,\exp\Big(-\tfrac{1}{2}(\hat{r}\theta)^T H(\hat{r})(\hat{r}\theta)\Big). \qquad (27)$$

Substituting (27) into (25) and extending the integrals to all $\mathbb{R}^M$ gives a third (final) approximation

$$I \cong \frac{\Psi(\hat{r})}{(2\pi)^M}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} e^{-\frac{1}{2}(\hat{r}\theta)^T H(\hat{r})(\hat{r}\theta)}\,d\theta_1\cdots d\theta_M\,. \qquad (28)$$

The hessian matrix $H(\hat{r})$ is positive definite at the saddle point because $\Psi(r)$ is a minimum there, i.e., $\hat{r}$ satisfies (24). The integrand in (28) is therefore proportional to a multivariate Gaussian distribution and is straightforward to evaluate:

$$I \cong \frac{\Psi(\hat{r})}{(2\pi)^M}\,\frac{(2\pi)^{M/2}}{(\hat{r}_1\cdots\hat{r}_M)\sqrt{\det H(\hat{r})}}\,. \qquad (29)$$

Substituting from (20) and simplifying gives (29) as

$$I = \frac{1}{(2\pi)^{M/2}\sqrt{\det H(\hat{r})}}\,\frac{\Psi_{\text{JPDA}(n)}(\hat{r})}{(\hat{r}_1\ \cdots\ \hat{r}_M)^2}\,, \qquad (30)$$

where, using the first equation in (21),

$$\begin{aligned} H(\hat{r}) &= \nabla_{rr}\left\{\log\Psi_{\text{JPDA}(n)}(\hat{r}_1,\cdots,\hat{r}_M) - \textstyle\sum_{j=1}^{M}\log\hat{r}_j\right\} \\ &= \text{Diag}[\hat{r}_1^2,\cdots,\hat{r}_M^2] + \nabla_{rr}\,\log\Psi_{\text{JPDA}(n)}(\hat{r}_1,\cdots,\hat{r}_M). \end{aligned} \qquad (31)$$

The expressions (24) and (30)-(31) comprise the saddle point approximation to the JPDA weight of a particle at x for target n.

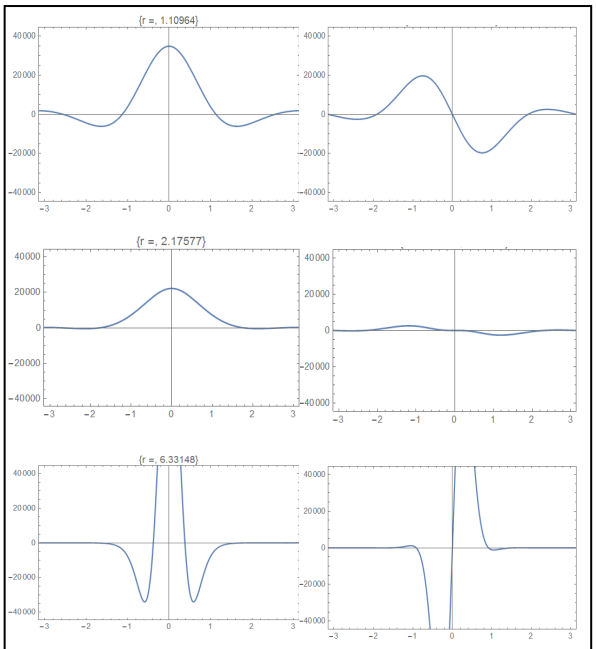

Figure 1. At the saddle point, the real part (left) of $f(z)$ is concentrated around zero and does not oscillate, while the imaginary part (right) is flat.

5 Why Saddle Points?

The role that saddle points play is illustrated in Figure 1 for one variable. It plots the real and imaginary parts of the function $f(z) = e^z(1+10z)^4 z^{-6}$ for $z = re^{i\theta}$ for three different choices of the radius r of a circular contour centered at zero. (The power z^{-6} was chosen to enhance the visual effectiveness of the example. Similar but less vivid behavior is exhibited for z^{-1}.) All plots range from $-\pi$ to π. The middle plots correspond to $\hat{r} = 2.18$, the saddle point in this example. The upper plots correspond to radius equal to $r = 1.1 < \hat{r}$ and the lower one to radius $r = 6.3 > \hat{r}$. In all three case, the imaginary parts integrate to zero and the real parts have the same integral.

These three specific choices for r pose no problem for numerical integration purposes (although the oscillations become very severe for sufficiently small and large radii). The purpose here, however, is to approximate the integrand and avoid numerical integration altogether. It is evident that the saddle point is the best choice.

Another example, this time in two real dimensions, is related to the JPDA expression (14) for $M = 2$ measurements and $N = 4$ targets. In a real JPDA problem these coefficients would be evaluated at a particular particle. Here, the constant terms were taken to be $a_0 = 0$, $b_{n0} = c_{j0} = 1$ and the remaining coefficients were chosen uniformly at random between $[0,1]$. The solid contour lines are for the real part of the exact integrand at the saddle point (25), and the dashed contours

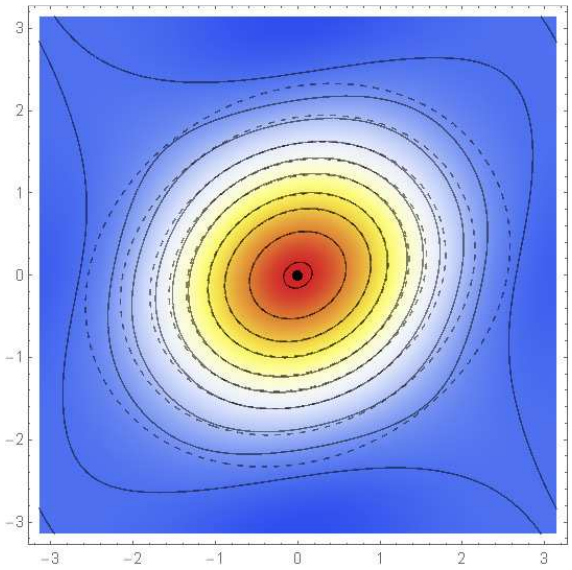

Figure 2. Contours of the exact integrand (solid) and approximate saddle point integrand (dashed). (Contours = 39, 35, 30, 26, 21, 17, 13, 8, 4, 0)

are of the saddle point approximating integrand (28). Both are plots from $-\pi$ to π of the two angles θ_1 and θ_2. The density plot corresponds to the exact integrand. The quadratic form character of the approximation is evident from Figure 2. The exact integrand is slightly negative in the outermost band between the contour lines for level zero. The saddle point approximation is never negative and extends to the entire plane.

6 Examples

The potential value of the saddle point method for JPDA was explored by evaluating the accuracy of the approximation on randomly generated JPDA style problems. The numerical evidence gathered is presented here. Further work using embedded code must await further investigation.

Particle weight calculations of the form (14) were generated for $N = 3, 5, \text{and } 7$ targets and for $M = 5, 10, \text{and } 15$ measurements. As in the second example of the previous section, the leading constants were taken to be either zero or one, while the coefficients of the β variables were uniformly randomly generated on the interval $[0,1]$. The number of trials was set to 1000.

For each combination of targets and measurements, the exact derivative was evaluated by using *Mathematica*® to find the symbolic derivative. This will give the exact particle weight – if the coefficients actually correspond to a real tracking problem.

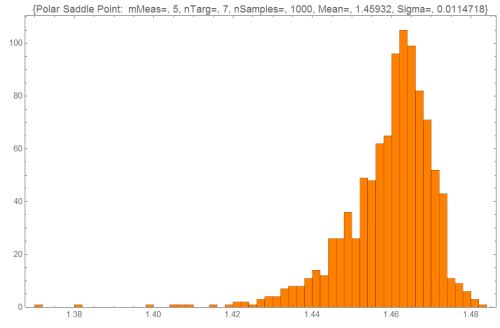

Figure 3. Ratio of saddle point approximation to exact derivative for 7 targets and 5 measurements.

For the same coefficients, the saddle point was found by minimizing the function (24) using *Mathematica*® (the FindMinimum routine). The numerical saddle point was used to evaluate the approximation (30).

In each trial, the ratio of the approximate to the exact weight was computed. The exact weight was calculated by evaluating the symbolic derivative. The histogram of the ratio is presented in Figure 3 for 7 targets and 5 measurements. Ideally, the mean will be exactly one. In this case the mean was about 1.46, that is, the approximation was 46% too large. However, the standard deviation about this mean was 0.011, that is, about 1.1%.

The factor of 1.47 implies that the particle weights are globally scaled too high by 47%. As is well known, such a global rescaling has no effect whatever on an SMC particle implementation. Indeed it is one of the most important properties of SMC methods.

The variation in the scale factor of 1.1% is a different matter. These errors are small compared to the other modeling errors that are inevitable in tracking problems.

The exact derivatives, i.e., the putative particle weights, span a wide range of values. Figure 4 gives a histogram for this case. The derivative ranges numerically from near zero to over 400.

Similar simulations were performed for the other combinations of number of targets and measurements. Table 1 tabulates the mean and standard deviations of the ratio of the saddle point approximation to the exact value. It is apparent from Table 1 that the change in the global scale factor is weakly dependent on the number of targets and moderately dependent on the number of measurements.

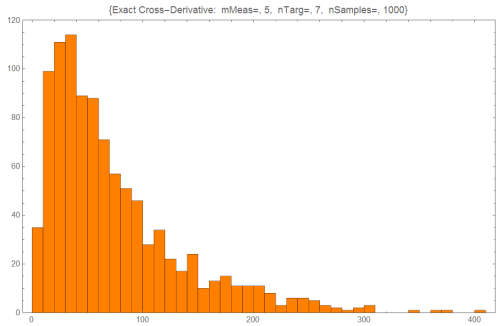

Figure 4. Histogram of exact derivatives for 7 targets and 5 measurements

$mMeas$ / $nTarg$	5	10	15
3	$\mu = 1.47$ $\sigma = 0.010$	$\mu = 2.24$ $\sigma = 0.0036$	$\mu = 3.37$ $\sigma = 0.0020$
5	$\mu = 1.45$ $\sigma = 0.015$	$\mu = 2.23$ $\sigma = 0.0048$	$\mu = 3.37$ $\sigma = 0.0028$
7	$\mu = 1.46$ $\sigma = 0.011$	$\mu = 2.21$ $\sigma = 0.0049$	$\mu = 3.36$ $\sigma = 0.0036$

Table 1. Ratio of saddle point approximation to the exact derivative: Mean and standard deviation

The scale does not affect SMC filters for JPDA, but the particle weight errors do. The standard deviations reported in Table 1 are, with one exception, less than 1%.

The question naturally arises as to whether or not small errors in the particle weights – before resampling – adversely affect JPDA tracking capability. A first study [14] of this question suggests that with particle weights perturbed by as much as 10% (uniformly distributed from 0% to 10%), the effect over many Monte Carlo trials is often not significant.

7 Computational Complexity

Saddle points can be found by solving the necessary conditions (23) directly, or by minimizing (24) via, e.g., relaxation, gradient descent, or even Nelder-Mead. Whatever method is chosen, it must be fast since the saddle point $\hat{r}$ depends on x, and SMC filters must find as many saddle points as there are particles.

The effort of finding the coefficients of the analytic function (14) is an overhead cost that is proportional to the product of the numbers of targets and measurements, NML, where L is the number of particles per target. This assumes the required integrals are evaluated (approximated) using the particle sets of each target.

Careful numerical analysis is needed to find the saddle point. Assuming a gradient descent method is used, and that the number of steps until convergence is bounded, the computational effort is proportional to a constant times the size of the gradient vector, M. This is needed for each particle and for each target, so the total effort of the saddle point calculations is $O(NML)$.

Computing a determinant using Gaussian elimination requires $O(M^3)$ effort. One determinant is needed for each particle, so the total effort is $O(LNM^3)$.

The total computational effort is the sum of the efforts of each part. This sum is dominated by $O(LNM^3)$ for serial computers. The constant hidden in the "big O" notation can vary widely in practice, depending on the implementation details, so it is hard to estimate how large a JPDA problem has to be for the saddle point approximation to be faster than an exact method, such as AD. (Also, for multicore or parallel computer architectures, the complexity bound can be reduced.)

For small problems the overhead of calculating the many coefficients (14) involved in JPDA may be a greater computational burden than the "pure" combinatorial costs of enumeration. The analytic function formulation can be used to quantify how much of the total effort is overhead and how much is combinatorial.

8 Generality of Saddle Point Method

The saddle point method can be applied to many generating functions $\Psi(\cdot)$, not just JPDA. Indeed, the argument of Section 3.2 carries through in the general case, so that the saddle point approximation (30) holds, but with $\Psi(\cdot)$ replacing the JPDA generating function.

However, the presence of the global scale factor that is evident in Table 1 limits this particular version of the saddle point method to SMC filters that approximate Bayes posterior probability density functions. This class of filters includes many classic filters, such as standard Bayes-Markov, PDA, JPDA, PMHT, IPDA, and JIPDA, as well as their multi-sensor and multi-scan (or, accumulated state density (ASD)) versions.

The saddle point method can be applied to SMC implementations of multitarget tracking filters that superpose targets in a common state space (and require a separate track extractor) provided the global scale factor is known or approximated. The global scale factor affects the estimated intensity function, i.e., the expected number of targets per unit state space. These filters are outside the scope of the present paper.

9 Concluding Remarks

Other methods for approximate JPDA with polynomial computational complexity have been proposed. The MCMC (Markov Chain Monte Carlo) method of [7] is especially interesting. The method can be extended to tracking filters other than JPDA. Other approximate methods are also discussed in [7].

The saddle point method differs from the randomized MCMC approximation method. It is a non-random direct numerical approach to the problem of computational complexity that arises in JPDA and a large number of multitarget tracking filters.

The Gaussian behavior of $\Psi(\cdot)$ at a saddle point is remarkable. This is due in large part to the way the Taylor series is manipulated, but the quality of the Gaussian fit is reminiscent of Central Limit Theorem behavior.

References

[1] P. Flagolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge Univ. Press, 2009.

[2] L. Bers, *Introduction to Several Complex Variables*, Lecture Notes, Courant Institute of Mathematical Sciences, New York University, 1964.

[3] N. Bleistein and R. A. Handelsman, *Asymptotic Expansion of Integrals*, Holt, Rinehart and Winston, New York, 1975. (Dover reprint edition, 1986)

[4] E. Hille, *Analytic Function Theory*, Vol. I, Ginn and Co., Boston, 1959.

[5] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1978.

[6] J. Vermaak, S. Godsill and P. Perez, "Monte Carlo filtering for multi-target tracking and data association," IEEE Trans. on Aerospace and Electronic Systems, T-AES 41(1), 309-332.

[7] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for Multi-Target Tracking," IEEE Trans. Automatic Control, T-AC 54(1), 2009, 481-497.

[8] L. Valiant, "The Complexity of Computing the Permanent," Theoret. Comp. Sci., vol. 8, 1979, 189-201.

[9] J. Uhlmann, "Matrix Permanent Inequalities for Approximating Joint Assignment Matrices in Tracking Systems," J. Franklin Institute, vol. 341, 2004, 569-593.

[10] R. Streit, "A Technique for Deriving Multitarget Intensity Filters Using Ordinary Derivatives," J. of Advances in Information Fusion, vol. 9(1), 2014, 3-12.

[11] R. Streit, "Generating Function Derivation of the PDA Filter," ISIF International Conference on Information Fusion, Salamanca, Spain, 2014.

[12] R. Streit and D. Krout, "Exact JPDA Particle Filter Using Automatic Differentiation," in preparation.

[13] R. Streit, *Poisson Point Processes—Imaging, Tracking, and Sensing*, Springer, New York, 2010.

[14] F. Govaers and C. Degen, Private Communication, February 18, 2015.

[15] A. Griewank, L. Lehman, H. Leovey, and M. Zilberman, "Automatic Evaluations of Cross-Derivatives," Mathematics of Computation, vol. 83, Jan. 2014, 251-274.

[16] C. Goutis and G. Casella, "Explaining the Saddlepoint Approximation," Working Paper 95-63, December 1995.

Appendix. Derivation of the Matrix Permanent by Cauchy's Theorem

Let $A = [a_{jm}]$ be an $M \times M$ matrix (real or complex). The GF of the matrix permanent, denoted by $\mathrm{Perm}(A)$, is

$$\Psi(\beta) = \prod_{j=1}^{M} \left(\sum_{m=1}^{M} a_{jm} \beta_m \right).$$

The form of this function is a special case of the JPDA expression (14). Let C be a circle centered at 0 with positive radius. Denote the cross-derivative of Ψ with respect to β at 0 by I. Using the multivariate version of Cauchy's Integral Theorem, I can be written (*cf.* (18))

$$(2\pi i)^M I = \oint_C \cdots \oint_C \prod_{j=1}^{M} \left(\sum_{m=1}^{M} a_{jm} \beta_m \right) \frac{d\beta_1 \cdots d\beta_M}{\beta_1^2 \cdots \beta_M^2}$$

$$= \sum_{m_1=1}^{M} \cdots \sum_{m_M=1}^{M} a_{1m_1} \cdots a_{Mm_M} \oint_C \cdots \oint_C \frac{\beta_{m_1} \cdots \beta_{m_M}}{\beta_1^2 \cdots \beta_M^2} d\beta_1 \cdots d\beta_M.$$

The M^M multiple integrals are non-zero if and only if the indices are distinct, i.e., they are a permutation of $\{1, 2, ..., M\}$. For these indices the multiple integral in the last expression is equal to $(2\pi i)^M$. Thus,

$$I = \sum_{\sigma \in \mathrm{Sym}(M)} a_{1\sigma_1} \cdots a_{M\sigma_M} \equiv \mathrm{Perm}[A],$$

where $\sigma = (\sigma_1, ..., \sigma_M) \in \mathrm{Sym}(M)$, where $\mathrm{Sym}(M)$ is the set of all permutations on the first M positive integers.