Exact Association Probability for Data with Bias and Features

JAMES P. FERRY

A crucial prerequisite to data fusion is data association: i.e., the specification of which data arise from the same source. The Bayesian approach to association pioneered by Mori and Chong is based on principled probability formulas, which thus provide reliable confidence estimates for association hypotheses, in contrast to approaches that rely on costs which can only be heuristically transformed into probabilities. This paper extends the Bayesian approach in several ways. It presents a general derivation of association probability between any number of sensors for arbitrary data types, then derives specific results for kinematic and non-kinematic cases. The kinematic case includes bias and is novel in three ways. First, it is a proper Bayesian approach to bias which integrates over all bias hypotheses rather than selecting one. Second, it handles bias on an arbitrary number of sensors. Third, the formula is exact: previous treatments of even the unbiased case involve an integral approximation which is not needed here. The treatment of features allows for several complex phenomena, including feature behavior which depends on object type, and noisy and/or missing feature data. A rigorous verification procedure is used to demonstrate that the implementation of these formulas produces correct probabilities.

Manuscript received December 04, 2008; revised August 20, 2009; released for publication December, 13, 2009.

Refereeing of this contribution was handled by Shozo Mori.

Authors' address: Metron, Inc., Reston, VA, E-mail: ferry@metsci.com.

1. INTRODUCTION

The data association problem arises whenever multiple sensors are trained on a common region containing multiple objects, which, in turn produce multiple measurements on each sensor. Having multiple views provides more information about the state of the objects in the region, provided the sensor data can be fused correctly. In order to fuse the data, however, it is necessary to know which measurements on different sensors arose from the same object. This is the data association problem, and it is roughly forty years old [24, 25]. In many applications, these "measurements" are not raw sensor measurements, but the posterior state estimates given by a single-sensor tracker such as a Kalman filter, and one speaks of "measurement-to-track" or "track-to-track" associations. Solving the association problem is necessary in Multiple Hypothesis Tracking [23], and much effort has gone into the development of algorithms to find the best association given track estimates with Gaussian error covariances [4]. The key difference between raw measurement data and track posteriors, however, is that the former lack inter-sensor correlation (for given object states). Such correlations are important in the track case, however, particularly if the posterior distributions have been influenced by previous inter-sensor data fusion. Indeed, even the posteriors of single-sensor trackers are correlated due to common process noise [3]. The scope of this paper is limited to situations in which such inter-sensor correlations are absent or have been compensated for.

The association problem was originally formulated in terms of *costs* with statistically motivated definitions —one found or devised a credible cost function and used it to seek low-cost associations. In 1990 Chong *et al.* introduced a more rigorous framework for assessing the quality of associations [7]. Mori and Chong extended this work in a series of papers in the early 2000s [16, 17, 18, 19, 20]. For the important case of two sensors with Gaussian kinematic data, they replaced the heuristic framework of costs with a rigorous, Bayesian reformulation of the association problem. In doing so, they gave a meaningful definition of the *probability* of an association, and argued that rather than seeking the association with minimal cost, one should seek the MAP association—i.e., the one with Maximal *A posteriori* Probability. In practice, the MAP method looks similar to the older method. The probabilities can be converted to costs, and one ends up computing the same quantities as before with one subtle difference: Mori and Chong showed that the correct cost threshold is not constant, but depends on the covariance matrices of the two measurements involved. Hence it came to be known as an *adaptive* threshold.

In 2002, Stone *et al.* generalized the work of Mori and Chong to non-kinematic data with the XMAP (eXtended Maximal *A posteriori* Probability) method [27]. In this and later work [6, 9, 11], association probabil-

ity formulas have been derived that take into account continuous and non-continuous data types beyond the purely kinematic, primarily for the case of two sensors.

This paper encompasses and extends the previous XMAP work, beginning with a general derivation for association probability in an abstract setting in Section 2. Like [18], this derivation includes an arbitrary prior on the number of objects and (in Appendix A) false alarms. Its novel aspects include a dependence on systematic errors (such as bias or covariance inflation) and a correlated prior on object state. In Section 3, this abstract derivation is applied to cases comprising kinematic and non-kinematic data types, and the association probability formula is decomposed. The kinematic component is dealt with in Section 4. The key contribution of this section is its treatment of bias, which improves on previous work in three ways: (1) rather than removing "the bias" (i.e., some particular bias hypothesis), it performs the proper Bayesian operation of integrating over all bias hypotheses; (2) it holds for an arbitrary number of sensors; and (3) it is exact—a certain integral approximation typically made even in the non-bias case is circumvented here. Section 5 demonstrates how to handle non-kinematic data in fairly complex cases, such as when the feature distributions and detection probabilities vary with object type. It also shows how to deal with missing data, and provides a robust and general method for handling noisy features. Finally, Section 6 works through an example in detail and demonstrates that the meaningful, exact probabilities produced by XMAP can be used to verify the formulas and their implementation to high precision.

2. ASSOCIATION PROBABILITY DERIVATION

This section derives a general formula for association probability in an abstract space. Working in an abstract space allows us to handle arbitrary types of measurement data in a consistent manner, whether it be traditional kinematic data, or, say, the messy output of a feature extractor which combines real-valued data with object classification calls and status flags. Thus, the bulk of this paper may be viewed as applying the general Theorem 2.4 below to special cases. A technical detail to bear in mind with this abstract treatment is that integrating over an abstract space requires one to specify a *measure* over the space. There will be no need, in this paper, to use measures other than Lebesgue measure for continuous data and counting measure for discrete data (which converts integrals to sums), so we will assume that the measure is clear from context.

We begin with a simple result for the probability density of getting a specific array of measurements on a single sensor s given the states of the objects that produced them. This result depends on the measurement likelihood function $L^s(z \mid x, \beta)$, which specifies the probability density of the measurement z arising given that the object which produced it was detected and was in state x, and that a systematic error β is acting on all measurements on sensor s. It depends also on the detection probability $P_{\mathrm{D}}^s(x)$ for an object in state x, and we use the notation $Q_{\mathrm{D}}^s(x) = 1 - P_{\mathrm{D}}^s(x)$ to denote the non-detection probability. This systematic error β may represent any measurement error process that acts on all measurements on a sensor at once, such as a translational bias or covariance inflation.

Let $\mathbf{z}^s = (z_i^s)_{i=1}^{n^s}$ denote an array of the n^s measurements on sensor s at some fixed time, and $\mathbf{x} = (x_j)_{j=1}^{n}$ denote the array of states of the n objects in scene. We let $J = \{1, 2, \ldots, n\}$ denote the set of all objects, and J_{D}^s denote the subset of objects detected on sensor s. We use the mapping $a^s : J_{\mathrm{D}}^s \to \{1, 2, \ldots, n^s\}$ to specify which object produced which measurement. We assume there are no false alarms (the false-alarm case is addressed in Appendix A), that there are no split or merged measurements, and that all permutations of measurement labels are equally likely. With these assumptions, we obtain the following preliminary result.

LEMMA 2.1 *The probability density of the measurement array* $\mathbf{z}^s$ *arising according to the mapping* a^s *given the object state array* $\mathbf{x}$ *and the systematic error* β^s *is*

$$\Pr(\mathbf{z}^s, a^s \mid \mathbf{x}, \beta^s) = \frac{1}{n^s!} \prod_{j \in J_{\mathrm{D}}^s} P_{\mathrm{D}}^s(x_j) L^s(z_{a^s(j)}^s \mid x_j, \beta^s) \times \prod_{j \in J \setminus J_{\mathrm{D}}^s} Q_{\mathrm{D}}^s(x_j). \tag{2.1}$$

PROOF Given the object state array $\mathbf{x}$, the probability $\Pr(J_{\mathrm{D}}^s \mid \mathbf{x})$ of the subset of detected objects being precisely J_{D}^s is the product of $P_{\mathrm{D}}^s(x_j)$ over $j \in J_{\mathrm{D}}^s$ times the product of $Q_{\mathrm{D}}^s(x_j)$ over $j \in J \setminus J_{\mathrm{D}}^s$. Given J_{D}^s (and $\mathbf{x}$), each of the possible $n^s!$ mappings a^s are equally likely, so $\Pr(a^s, J_{\mathrm{D}}^s \mid \mathbf{x})$, which is identical to $\Pr(a^s \mid \mathbf{x})$ because a^s determines J_{D}^s, can be expressed $\Pr(a^s \mid \mathbf{x}) = \Pr(a^s \mid J_{\mathrm{D}}^s, \mathbf{x}) \Pr(J_{\mathrm{D}}^s \mid \mathbf{x}) = \Pr(J_{\mathrm{D}}^s \mid \mathbf{x}) / n^s!$. The probability density of the measurement array $\mathbf{z}^s$ given a^s, $\mathbf{x}$, and β^s is the product of the individual likelihood functions $L^s(z_i^s \mid x_j, \beta^s)$, where $i = a^s(j)$, over all $j \in J_{\mathrm{D}}^s$. Equation (2.1) now follows from $\Pr(\mathbf{z}^s, a^s \mid \mathbf{x}, \beta^s) = \Pr(\mathbf{z}^s \mid a^s, \mathbf{x}, \beta^s) \Pr(a^s \mid \mathbf{x})$.

It is straightforward to generalize Lemma 2.1 from a single sensor to a set of sensors S. We let $\mathbf{z} = (\mathbf{z}^s)_{s \in S}$ denote the array of all individual measurement arrays $\mathbf{z}^s$, and $\boldsymbol{\beta} = (\beta^s)_{s \in S}$ denote the array of all systematic errors. We assume that the measurement process is independent between sensors, bearing in mind that this limits the applicability to tracking unless the inter-sensor dependence can be compensated for. With multiple sensors it is convenient to express the information contained in the mappings $\{a^s\}_{s \in S}$ in terms of

$$a(j) = \{(s, i) : a^s(j) = i\} \qquad \text{and} \qquad \bar{a}(j) = \{s : j \in J_{\mathrm{D}}^s\}. \tag{2.2}$$

The function $\bar{a}$ gives the set of sensors which detect each object j, whereas a gives the measurement indices of each detection too. With this notation, we have the following corollary of Lemma 2.1.

COROLLARY 2.2 *The probability density of the measurement arrays* $\mathbf{z}$ *arising according to the function* a *given the object state array* $\mathbf{x}$ *and the systematic errors* $\boldsymbol{\beta}$ *is*

$$\Pr(\mathbf{z}, a \mid \mathbf{x}, \boldsymbol{\beta}) = \left(\prod_{s \in S} \frac{1}{n^s!} \right) \prod_{j \in J} \times \left(\prod_{(s,i) \in a(j)} P_{\mathrm{D}}^s(x_j) L^s(z_i^s \mid x_j, \beta^s) \prod_{s \notin \bar{a}(j)} Q_{\mathrm{D}}^s(x_j) \right). \tag{2.3}$$

PROOF The assumption of independence between the measurement processes implies that $\Pr(\mathbf{z}, a \mid \mathbf{x}, \boldsymbol{\beta})$ may be obtained as the product of (2.1) over $s \in S$. The result (2.3) is merely a rearrangement of this.

We now seek to eliminate the dependence on the state array $\mathbf{x}$. To do so, we need an expression for the prior distribution of the state. The expression introduced below is novel, and requires some discussion of (a) the shortcomings of the usual approach, (b) how the problem should be solved in principle, and (c) the compromise used here. Although the derivation in this section applies to an arbitrary state space, the problematic case is the traditional, kinematic one, so we shall think of $\mathbf{x}$ as a kinematic quantity for this discussion.

The usual assumption about the prior distribution of $\mathbf{x}$ when bias is absent is that its components x_j are i.i.d., each distributed according to some known distribution p^0. The distribution $p^0(x)$ of a single object state x is then taken to be uniform over some finite region of space X^0 [17]. An appropriate volume V for X^0 may be estimated from the data, and $p^0(x)$ modeled as $I_{X^0}(x)/V$, where $I_{X^0}(x)$ is the indicator function for the region X^0. The precise location of the region X^0 does not matter because $p^0(x)$ is later approximated by the constant $1/V$ in the integrals where it appears. This approximation is valid because the measurement errors are typically much smaller than X^0, and this formulation, which leads to the *adaptive threshold* works well in practice [26].

Bias errors may be larger than X^0, however, so the approximation $p^0(x) \approx 1/V$ fails in the bias case. Maintaining $p^0(x)$ as $I_{X^0}(x)/V$ yields intractable integrals, so it is natural to consider a Gaussian model for $p^0(x)$. In this case, however, the precise location of the Gaussian's peak in state space must be estimated from the measurements, while correcting for the (unknown) biases, and the resulting formulation becomes messy and *ad hoc*. Indeed, it must be *ad hoc* because it involves estimating the location of the Gaussian's peak *from the data*—a clear violation of Bayesian methodology.

One is led to such violations because of a faulty initial assumption: that the prior distribution of $\mathbf{x}$ is well modeled as the product of $p^0(x_j)$ for some *known* distribution p^0. The characteristic size V and nominal *center* ξ of this prior distribution are usually both unknown. However, it is reasonable to model the states x_j as being conditionally i.i.d. given V and ξ, and then to specify priors on V and ξ. This leads to additional integrals over V and ξ. However, one may argue, for example, that the weight in the integrand of the integral over V is concentrated in the region of V-space that is reinforced by the measurement data, so evaluating the integrand at a single point V^* (estimated from the data) amounts to a reasonable approximation of the integral. This provides a Bayesian justification for an otherwise *ad hoc* procedure. However, in the bias case there is no single value of ξ where the integrand is concentrated: this center location depends on the unknown biases β^s. Rather than replace ξ by this function of the biases, it is fairly easy to retain it and perform the integrals exactly. Doing the same for V, however, (which, in the Gaussian case is actually an entire covariance matrix) is too difficult, so we will estimate it from the data, relying on the argument above for justification.

We therefore assume the following prior distribution on the state array $\mathbf{x}$ (which we now resume treating as abstract rather than kinematic):

$$p^0(\mathbf{x} \mid n, \xi) = \prod_{j \in J} p^0(x_j \mid \xi). \tag{2.4}$$

Multiplying (2.3) by $p^0(\mathbf{x} \mid n, \xi)$ and integrating over $\mathbf{x}$ results in a product of integrals over x_j for each object $j \in J$. These have the form

$$P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi) = \int p^0(x \mid \xi) \prod_{(s,i) \in \alpha} P_{\mathrm{D}}^s(x) L^s(z_i^s \mid x, \beta^s) \times \prod_{s \notin \bar{\alpha}} Q_{\mathrm{D}}^s(x) dx \tag{2.5}$$

for $\alpha = a(j)$. For the special case of α containing only the single measurement (s,i) we will use the notation $P^s(z_i^s \mid \boldsymbol{\beta}, \xi)$ in lieu of $P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi)$. Let J_{D} denote the subset of objects in J detected on some sensor, and $n_{\mathrm{D}} = |J_{\mathrm{D}}|$ be the number of detected objects. For $j \notin J_{\mathrm{D}}$, (2.5) takes a particularly simple form:

$$q = \int p^0(x \mid \xi) \prod_{s \in S} Q_{\mathrm{D}}^s(x) dx. \tag{2.6}$$

Here we are imposing the condition that q is independent of ξ. This will happen automatically later when we stipulate in (3.18) that $P_{\mathrm{D}}^s(x)$ be independent of the kinematic component of x. This stipulation is not realistic: $P_{\mathrm{D}}^s(x)$ can vary greatly with aspect angle and range. However, the case in which $P_{\mathrm{D}}^s(x)$ has kinematic dependence makes the calculations in Section 4 too complicated, though it would be a suitable topic for future work.

COROLLARY 2.3 *The probability density of the measurement arrays* $\mathbf{z}$ *arising according to the function* a *given*

the number of objects n, the systematic errors $\boldsymbol{\beta}$, and the center ξ of the prior region of state space is

$$\Pr(\mathbf{z}, a \mid n, \boldsymbol{\beta}, \xi) = \frac{q^{n-n_{\mathrm{D}}}}{\prod_{s\in S} n^s!} \prod_{j\in J_{\mathrm{D}}} P^{a(j)}(\mathbf{z} \mid \boldsymbol{\beta}, \xi). \quad (2.7)$$

PROOF This is obtained by multiplying (2.3) by (2.4) and integrating over $\mathbf{x}$.

The function a defined in (2.2) maps each object to the set of measurements it produces. An association $[a]$ is defined to be the collection of these sets, $[a] = \{a(j) : j \in J_{\mathrm{D}}\}$. There are exactly $n!/(n-n_{\mathrm{D}})!$ functions a' for which $[a'] = [a]$, all of which are equally probable, so the association probability $\Pr(\mathbf{z}, [a] \mid n, \boldsymbol{\beta}, \xi)$ is $n!/(n-n_{\mathrm{D}})!$ times $\Pr(\mathbf{z}, a \mid n, \boldsymbol{\beta}, \xi)$. We may now eliminate the dependence on the total number of objects n. The prior probability for the number of objects being n is denoted $\rho^0(n)$, so

$$\Pr(\mathbf{z}, [a] \mid \boldsymbol{\beta}, \xi) = \frac{\gamma^0(n_{\mathrm{D}})}{\prod_{s\in S} n^s!} \prod_{\alpha\in[a]} P^{\alpha}(\mathbf{z} \mid \boldsymbol{\beta}, \xi), \quad (2.8)$$

where

$$\gamma^0(n_{\mathrm{D}}) = \sum_{n=n_{\mathrm{D}}}^{\infty} \rho^0(n) \frac{n!}{(n-n_{\mathrm{D}})!} q^{n-n_{\mathrm{D}}}. \quad (2.9)$$

We denote the prior on the systematic errors $P^0(\boldsymbol{\beta})$, and the prior on the center, $P^0_{\Xi}(\xi)$. When integrating (2.8), the key quantity to compute is

$$F(\mathbf{z}, [a]) = \int\int \prod_{\alpha\in[a]} P^{\alpha}(\mathbf{z} \mid \boldsymbol{\beta}, \xi) P^0(\boldsymbol{\beta}) P^0_{\Xi}(\xi) d\boldsymbol{\beta}\, d\xi. \quad (2.10)$$

Finally, we introduce the following ratios of γ^0 and F to their values for the null association a_0, which assigns each of the n_T measurements in $\mathbf{z}$ to a distinct object:

$$g([a]) = \frac{\gamma^0(n_{\mathrm{D}})}{\gamma^0(n_T)} \qquad \text{and} \qquad G(\mathbf{z}, [a]) = \frac{F(\mathbf{z}, [a])}{F(\mathbf{z}, [a_0])}. \quad (2.11)$$

THEOREM 2.4 *The probability of the association* $[a]$ *given the measurements* $\mathbf{z}$ *is*

$$\Pr([a] \mid \mathbf{z}) = \Pr([a_0] \mid \mathbf{z}) g([a]) G(\mathbf{z}, [a]). \quad (2.12)$$

PROOF From (2.8) we observe that joint probability density $\Pr(\mathbf{z}, [a])$ is $F(\mathbf{z}, [a])$ times $\gamma^0(n_{\mathrm{D}})$ divided by the product of the $n^s!$. The conditional probability $\Pr([a] \mid \mathbf{z})$ of an association given the measurement data is $\Pr(\mathbf{z}, [a])/\Pr(\mathbf{z})$. Dividing $\Pr([a] \mid \mathbf{z})$ by the normalization constant $\Pr([a_0] \mid \mathbf{z})$ yields (2.12).

The key to computing the association probability is evaluating $F(\mathbf{z}, [a])$ (or its normalization $G(\mathbf{z}, [a])$). This is the topic of Sections 3–5. The combinatorial factor $g([a])$ encapsulates the effect of the prior distribution of the number of objects. Formulas for it are given in Appendix B. Although the derivation in this section assumed there are no false alarms, Appendix A demonstrates that the effect of false alarms may be included by modifying the factor $g([a])$—no change to the definition of $G(\mathbf{z}, [a])$ is necessary.

3. SIMPLIFICATION FOR SPECIAL CASES

Theorem 2.4 gives a general formula for association probability, but requires the evaluation of the integrals in (2.5) and (2.10). With complicated data types, evaluating these integrals is not as simple as it may appear. Therefore we demonstrate how the problem simplifies in various special cases. Section 3.1 gives a formula much simpler than (2.12) which holds when there is no dependence on $\boldsymbol{\beta}$ or ξ. Section 3.1.1 specializes this further to the two-sensor, kinematic case, connecting the general XMAP formulation presented here to the original MAP formulation of Mori and Chong [16, 17]. Finally, Section 3.2 demonstrates how to decompose the problem into kinematic and non-kinematic components when both data types are present. The kinematic component is then evaluated explicitly in Section 4, and the non-kinematic component in Section 5.

3.1. Simplification in the Absence of Systematic Error

When the systematic errors $\boldsymbol{\beta}$ and the center ξ are known, we may assume each to be zero (by suitably transforming the data $\mathbf{z}$): i.e., $P^0(\boldsymbol{\beta}) = \delta(\boldsymbol{\beta})$, and $P^0_{\Xi}(\xi) = \delta(\xi)$. Letting $P^{\alpha}(\mathbf{z})$ and $P^s(z^s_i)$ denote $P^{\alpha}(\mathbf{z} \mid \mathbf{0}, 0)$ and $P^s(z^s_i \mid 0, 0)$, respectively, we define

$$R^{\alpha}(\mathbf{z}) = \frac{P^{\alpha}(\mathbf{z})}{\prod_{(s,i)\in\alpha} P^s(z^s_i)}, \quad (3.1)$$

which is the ratio of the probability density of the measurements in α arising from a single object to the probability density that each arises from a different object (aside from a factor which accounts for the different number of detected objects in the two cases—this is embedded in $g([a])$).

THEOREM 3.1 *When the systematic errors* $\boldsymbol{\beta}$ *and the center* ξ *are known, the probability of the association* $[a]$ *given the measurements* $\mathbf{z}$ *is*

$$\Pr([a] \mid \mathbf{z}) = \Pr([a_0] \mid \mathbf{z}) g([a]) \prod_{\alpha\in[a]^+} R^{\alpha}(\mathbf{z}), \quad (3.2)$$

where $[a]^+$ *denotes the subset of those* $\alpha \in [a]$ *with at least two measurements.*

PROOF Because $R^{\alpha}(\mathbf{z}) = 1$ when $|\alpha| = 1$, the product over $\alpha \in [a]^+$ in (3.2) may be extended to $\alpha \in [a]$. This product equals $F(\mathbf{z}, [a])/F(\mathbf{z}, [a_0])$ when we set $P^0(\boldsymbol{\beta}) = \delta(\boldsymbol{\beta})$ and $P^0_{\Xi}(\xi) = \delta(\xi)$ in (2.10).

3.1.1. Two-sensor kinematic case

To recover the original MAP result [16, 17], we begin with Theorem 3.1 and make four further simplify-

ing assumptions: first, that $\rho^0(n)$ is Poisson distributed; second, that there are only two sensors; third, that the detection probabilities on each sensor are constant; and fourth, that the data is purely kinematic, with Gaussian measurement error distributions. The two-sensor kinematic case without bias is important because it admits a computationally efficient solution. One first constructs a cost matrix whose entries c_{ij} are the Mahalanobis distances between measurement i on sensor 1 and measurement j on sensor 2. Then a *cost threshold* is subtracted from each c_{ij}. Finally, one finds the association with minimal total cost using, for example, the JVC algorithm [8, 13], and, if desired, iterates this process using Murty's algorithm [21] to get the k best associations. (When there are more than two sensors, however, finding the association with least cost is known to be NP-hard. Approximate methods have been employed based on Lagrangian relaxation [22] or on stitching together solutions to pairwise problems. These approaches are compared in [2].)

The effect of letting $\rho^0(n)$ be Poisson distributed is discussed in Appendix B. It allows us to replace (3.2) with the purely multiplicative form (B.4). Specializing to two sensors simplifies matters further because in this case every $\alpha \in [a]^+$ has the form $\alpha = \{(1,i),(2,j)\}$. We may re-write this more compactly as $\alpha = (i,j)$. Thus the two-sensor version of (B.4) may be written

$$\Pr([a] \mid \mathbf{z}) = \Pr([a_0] \mid \mathbf{z}) \prod_{(i,j)\in[a]^+} \tilde{R}_{ij}(\mathbf{z}), \tag{3.3}$$

where

$$\tilde{R}_{ij}(\mathbf{z}) = \frac{\tilde{P}^{12}(z_i^1, z_j^2)}{\tilde{P}^1(z_i^1)\tilde{P}^2(z_j^2)}. \tag{3.4}$$

The functions $\tilde{P}^s$ and $\tilde{P}^{12}$ are special cases of (2.5), with β and ξ eliminated, P_{D}^s set to a constant, and factors of ν in (B.5) included:

$$\tilde{P}^s(z_i^s) = \nu P_{\mathrm{D}}^s Q_{\mathrm{D}}^{3-s} \int p^0(x) L^s(z_i^s \mid x) dx \qquad \text{and} \tag{3.5}$$

$$\tilde{P}^{12}(z_i^1, z_j^2) = \nu P_{\mathrm{D}}^1 P_{\mathrm{D}}^2 \int p^0(x) L^1(z_i^1 \mid x) L^2(z_j^2 \mid x) dx. \tag{3.6}$$

(To incorporate false alarms, replace the factor $\nu P_{\mathrm{D}}^s Q_{\mathrm{D}}^{3-s}$ in (3.5) with $\nu P_{\mathrm{D}}^s Q_{\mathrm{D}}^{3-s} + \nu_{\mathrm{FA}}^s$, where ν_{FA}^s is the expected number of false alarms on sensor s: see (B.18).)

The integrals in (3.5) and (3.6) are simple to evaluate when L^s has a Gaussian distribution. We use the notation

$$\mathcal{N}(x;\mu,V) = \frac{1}{\sqrt{|2\pi V|}} \exp\left(-\frac{1}{2}(x-\mu)^T V^{-1}(x-\mu)\right) \tag{3.7}$$

for a Gaussian in x with mean μ and covariance matrix V. Specializing to the standard kinematic case, we let the value z_i^s of measurement i on sensor s have the form $z_i^s = (y_i^s, V_i^s)$, where y_i^s the state estimate, and V_i^s is the error estimate on y_i^s. We then stipulate that

$$L^s(z_i^s \mid x) = \mathcal{N}(y_i^s; x, V_i^s) P_V(V_i^s). \tag{3.8}$$

Essentially this means that y_i^s has a Gaussian distribution centered at the true state x, with covariance matrix given by V_i^s. There is an additional complication, however: the estimated covariance matrix V_i^s is part of the data, so its distribution must be modeled as well. The simplest assumption is that it is independent of x (and of y_i^s). In this case the precise form of the distribution P_V does not matter: it drops out of the calculation. (More sophisticated treatments are certainly possible: for example, there may be systematic over- or under-reporting of covariance, or the size of the covariance matrix itself may yield object-type information, in which case P_V would be modeled to have a dependence on the object type component of the state.)

Following [7], we let the prior distribution on an object's state x be constant over some region X^0:

$$p^0(x) = \frac{1}{\mathrm{Vol}(X^0)} I_{X^0}(x), \tag{3.9}$$

where $I_{X^0}(x)$ is equal to 1 for $x \in X^0$, and 0 otherwise. Given the above assumptions, (3.4) may be written

$$\tilde{R}_{ij}(\mathbf{z}) = \mathcal{N}(y_i^1; y_j^2, V_i^1 + V_j^2) \frac{\mathrm{Vol}(X^0)}{\nu q} \times \frac{\int_{X^0} \mathcal{N}(x;\mu_{ij},W_{ij}) dx}{\int_{X^0} \mathcal{N}(x;y_i^1,V_i^1) dx \int_{X^0} \mathcal{N}(x;y_j^2,V_j^2) dx}, \tag{3.10}$$

where $q = Q_{\mathrm{D}}^1 Q_{\mathrm{D}}^2$, and μ_{ij} and W_{ij} are given by

$$\begin{aligned} \mu_{ij} &= y_i^1 + V_i^1 (V_i^1 + V_j^2)^{-1} (y_j^2 - y_i^1) \qquad \text{and} \\ W_{ij} &= V_i^1 (V_i^1 + V_j^2)^{-1} V_j^2. \end{aligned} \tag{3.11}$$

Assuming the Gaussians in the integrals in (3.10) have most of their weight within X^0, each integral is approximately 1. This yields the following *cost* of associating i and j:

$$\begin{aligned} c_{ij} &= -2 \log \tilde{R}_{ij}(\mathbf{z}) \\ &= (y_i^1 - y_j^2)^T (V_i^1 + V_j^2)^{-1} (y_i^1 - y_j^2) - A_{ij}, \end{aligned} \tag{3.12}$$

where

$$A_{ij} = 2 \log \mathrm{Vol}(X^0) - \log |2\pi (V_i^1 + V_j^2)| - 2 \log(\nu q). \tag{3.13}$$

The cost c_{ij} is thus seen to be the Mahalanobis distance $(y_i^1 - y_j^2)^T (V_i^1 + V_j^2)^{-1} (y_i^1 - y_j^2)$ between the measurement pair (i,j) minus Mori and Chong's adaptive threshold A_{ij} [16, 17, 18]. Traditionally, a variety of methods had been used to set this threshold [4, 5]. Although the main historical significance of Mori and Chong's work is the introduction of a rigorous Bayesian

approach to association, the more immediate impact was the introduction of a threshold A_{ij} which has been shown to be superior to the previous, fixed thresholds [26].

There is a minor flaw with (3.13), however: it fails when some V_i^s are large relative to the region X^0 because this violates the assumption that allowed the integrals in (3.10) to be approximated as 1. Such measurements must be preprocessed out as unassociatable when using (3.13). Additional, non-kinematic data may render such measurements associatable, however, so it is preferable to modify (3.13) to be robust to any input V_i^s. A complicated method for doing this is given in [9], but here a much simpler method is given. When the covariance of a Gaussian being integrated is large compared to X^0, the integral may be approximated as the Gaussian's peak value times $\mathrm{Vol}(X^0)$. This leads to the following robust modification of (3.13), which has the heuristic interpretation of limiting the uncertainty of an object's location to X^0 even if $V_i^s \to \infty$:

$$A_{ij} = \max(2\log \mathrm{Vol}(X^0) - \log|2\pi(V_i^1 + V_j^2)|, 0) - 2\log(\nu q). \tag{3.14}$$

To use the adaptive threshold (3.14) one needs values for q, ν, and $\mathrm{Vol}(X^0)$. Section 2 notes that when such parameters are unknown, the proper Bayesian procedure is to give them a prior distribution and integrate them out of the problem. In practice, however, setting values of P_{D}^s that are even approximately correct produces better results than those obtained using the traditional, fixed threshold [26]. These values are used to compute $q = (1 - P_{\mathrm{D}}^1)(1 - P_{\mathrm{D}}^2)$ and, using (B.6), $\nu = (n^1 + n^2)/(P_{\mathrm{D}}^1 + P_{\mathrm{D}}^2)$.

It remains to estimate $\mathrm{Vol}(X^0)$. Following [9], we do this by first estimating the covariance of the location data y_i^s for all measurements. In the absence of sensor bias, the unbiased covariance estimator $\hat{V}$ may be computed as follows. Collect all n_T measurement positions y_i^s on all sensors into a single array with elements y_i for $i = 1, 2, \ldots, n_T$. Then

$$\hat{V} = \frac{1}{n_T - 1}\sum_{i=1}^{n_T}(y_i - \hat{y})(y_i - \hat{y})^T \qquad \text{where} \quad \hat{y} = \frac{1}{n_T}\sum_{i=1}^{n_T} y_i. \tag{3.15}$$

In Section 4 we will consider the case with r sensors and bias. In this case, $\hat{V}$ is given by

$$\hat{V} = \frac{1}{n_T - r}\sum_{s\in S}\sum_{i=1}^{n^s}(y_i^s - \hat{y}^s)(y_i^s - \hat{y}^s)^T \qquad \text{where} \quad \hat{y}^s = \frac{1}{n^s}\sum_{i=1}^{n^s} y_i^s. \tag{3.16}$$

To compute the volume of X^0 from $\hat{V}$, we assume that X^0 is a Cartesian product of ellipsoidal regions with covariance matrix $\hat{V}$. If X^0 is a product of m-dimensional ellipsoids (e.g., x could be 6-dimensional, with $m = 3$ being the physical dimension of position- and of velocity-space), then

$$\mathrm{Vol}(X^0) = \sqrt{|2\pi\theta\hat{V}|} \qquad \text{where} \quad \theta = (1 + m/2)(m/2)!^{-2/m}. \tag{3.17}$$

The values of θ for $m = 1$, 2, and 3 are $6/\pi \approx 1.91$, 2, and $(5/3)\sqrt[3]{6/\pi} \approx 2.07$, respectively. Because these values are so close, setting $\theta = 2$ for all problems is an acceptable approximation.

This volume estimate is not ideal. It is sensitive to outliers and to measurements being close to co-planar. Fortunately, its effect is limited to the threshold—i.e., the decision of whether to associate two measurements at all. Some authors dispense with the volume estimation entirely, using a diffuse spatial prior [14, 15], which is perfectly valid, but limits the power of the resulting method to hypothesis tests between associations representing the same number of detected objects.

3.2. Splitting into Components

When $\mathbf{z}$ comprises various data types, with some degree of independence in how each type is generated, (2.12) can be split into components for each data type. Here we will make a major simplification by splitting the kinematic data from any non-kinematic data types present. The kinematic data retain the complications due to β and ξ, and Section 4 demonstrates how to handle this. The non-kinematic data is modeled to be without the complications due to β and ξ: therefore the non-kinematic component of association probability simplifies into a product over $\alpha \in [a]^+$, as in Theorem 3.1. Examples of how to model various non-kinematic data types are given in Section 5.

To split the problem into components, we split both the state x and each measurement z_i^s into a kinematic component (K) and another component (J, for all non-kinematic variables *jointly*): let $x = (x^K, x^J)$ and $z_i^s = (z_i^{Ks}, z_i^{Js})$. We make the following assumptions about how the prior distribution, detection probability, and measurement likelihood functions split:

$$P_{\mathrm{D}}^s(x) = P_{\mathrm{D}}^{Js}(x^J), \tag{3.18}$$

$$p^0(x \mid \xi) = p^{K0}(x^K - \xi)p^{J0}(x^J) \qquad \text{and} \tag{3.19}$$

$$L^s(z_i^s \mid x, \beta^s) = L^{Ks}(z_i^{Ks} \mid x^K, \beta^s)L^{Js}(z_i^{Js} \mid x^J). \tag{3.20}$$

Equation (3.18) stipulates that the detection probability is independent of the kinematic state. As discussed in Section 2, this is an unfortunate but necessary oversimplification. It is allowed to depend on non-kinematic variables, however: for example, it is plausible that one could have a reasonable model of detection probability

as a function of object type (cf. Section 5.1). Equations (3.19) and (3.20) stipulate that the complications due to the center ξ of the object region and the systematic error β^s are solely kinematic phenomena. The assumptions (3.18)–(3.20) permit a relatively simple treatment of the non-kinematic variables, while addressing the effects of kinematic bias.

The definition of q in (2.6) may be simplified to

$$q = \int \prod_{s \in S} Q_{\mathrm{D}}^{Js}(x^J) dx^J, \tag{3.21}$$

because for any ξ the integral of $p^{K0}(x^K - \xi)$ over all x^K is 1. Note that q is independent of ξ, as required in the text following (2.6). The key probability density $P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi)$ may be split as follows:

$$P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi) = P^{K\alpha}(\mathbf{z}^K \mid \boldsymbol{\beta}, \xi) P^{J\alpha}(\mathbf{z}^J), \tag{3.22}$$

using (2.5) to give us these formulas for the kinematic and non-kinematic components of P^α:

$$P^{K\alpha}(\mathbf{z}^K \mid \boldsymbol{\beta}, \xi) = \int p^{K0}(x^K - \xi) \prod_{(s,i) \in \alpha} L^{Ks}(z_i^{Ks} \mid x^K, \beta^s) dx^K \quad \text{and} \tag{3.23}$$

$$P^{J\alpha}(\mathbf{z}^J) = \int p^{J0}(x^J) \prod_{(s,i) \in \alpha} P_{\mathrm{D}}^{Js}(x^J) L^{Js}(z_i^{Js} \mid x^J) \times \prod_{s \notin \bar{\alpha}} Q_{\mathrm{D}}^{Js}(x^J) dx^J. \tag{3.24}$$

THEOREM 3.2 *Given the assumptions* (3.18)–(3.20)*, the probability of the association* $[a]$ *given the measurements* $\mathbf{z}$ *is*

$$\Pr([a] \mid \mathbf{z}) = \Pr([a_0] \mid \mathbf{z}) g([a]) G^K(\mathbf{z}^K, [a]) G^J(\mathbf{z}^J, [a]), \tag{3.25}$$

with

$$G^K(\mathbf{z}^K, [a]) = \frac{F^K(\mathbf{z}^K, [a])}{F^K(\mathbf{z}^K, [a_0])}, \tag{3.26}$$

where

$$F^K(\mathbf{z}^K, [a]) = \int \int \prod_{\alpha \in [a]} P^{K\alpha}(\mathbf{z}^K \mid \boldsymbol{\beta}, \xi) P^0(\boldsymbol{\beta}) P_\Xi^0(\xi) d\boldsymbol{\beta}\, d\xi, \tag{3.27}$$

and

$$G^J(\mathbf{z}^J, [a]) = \prod_{\alpha \in [a]^+} R^{J\alpha}(\mathbf{z}^J), \tag{3.28}$$

where

$$R^{J\alpha}(\mathbf{z}^J) = \frac{P^{J\alpha}(\mathbf{z}^J)}{\prod_{(s,i) \in \alpha} P^{Js}(z_i^{Js})}. \tag{3.29}$$

PROOF This result follows directly from the definitions of the quantities involved.

4. THE KINEMATIC COMPONENT

In this section we derive an exact formula for the kinematic component $G^K(\mathbf{z}^K, [a])$ of the association probability in the case of an arbitrary number of sensors with bias effects included. We will drop the superscript K throughout this section. Equation (3.26) shows that evaluating $G(\mathbf{z}, [a])$ involves integrating over the sensor biases $\boldsymbol{\beta}$. This differs from the more typical approaches which identify and remove a bias hypothesis (either a distinct hypothesis for each association, or, more crudely, one hypothesis for all associations). Such approaches can fail even in quite simple scenarios, such as the one discussed in [10].

To obtain a formula for the kinematic component $G(\mathbf{z}, [a])$ of the association probability, we must evaluate the integrals in (3.23) and (3.27). To do so, we need appropriate models for the quantities which appear in them. For the bias prior we assume

$$P^0(\boldsymbol{\beta}) = \prod_{s \in S} P^{0s}(\beta^s) = \prod_{s \in S} \mathcal{N}(\beta^s; \beta_0^s, B^s). \tag{4.1}$$

Here β_0^s is the mean bias on sensor s, and B^s is the bias covariance matrix. In practice, one would typically set β_0^s to zero because one could simply add it to each measurement on sensor s in a pre-processing step. The bias covariance matrix B^s should be part of the performance specifications for sensor s. If it is not, however, one may set B^s to be diffuse. We let the prior on ξ be diffuse:

$$P_\Xi^0(\xi) = \mathcal{N}(\xi; \xi_0, V_\Xi) \qquad \text{where} \quad V_\Xi \to \infty. \tag{4.2}$$

The irrelevant value of ξ_0 will be retained until the step that takes $V_\Xi \to \infty$ eliminates it.

We let the measurement likelihood function be Gaussian, writing the measurement as $z_i^s = (y_i^s, V_i^s)$, as in Section 3.1.1. Generalizing (3.8) to include bias, we have

$$L^s(z_i^s \mid x, \beta^s) = \mathcal{N}(y_i^s; x - \beta^s, V_i^s). \tag{4.3}$$

Here the factor $P_V(V_i^s)$ which appeared in (3.8) has been set to 1 (because the value does not matter, and it would, in fact, be 1 in a suitably chosen measure space). Finally, we assume that the prior distribution on x is Gaussian with mean ξ and known variance V_0,

$$p^0(x - \xi) = \mathcal{N}(x; \xi, V_0), \tag{4.4}$$

for the reasons discussed before Equation (2.4). Equation (3.16) may be used to produce a value for V_0 in practice.

LEMMA 4.1

$$P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi) = \sqrt{|2\pi W_\alpha|}\, \mathcal{N}(\xi, \mu_\alpha, V_0) \prod_{(s,i) \in \alpha} \mathcal{N}(y_i^s + \beta^s; \mu_\alpha, V_i^s), \tag{4.5}$$

where

$$W_\alpha = \left(V_0^{-1} + \sum_{(s,i) \in \alpha} (V_i^s)^{-1} \right)^{-1} \qquad \text{and}$$
$$\mu_\alpha = W_\alpha \left(V_0^{-1} \xi + \sum_{(s,i) \in \alpha} (V_i^s)^{-1} (y_i^s + \beta^s) \right). \tag{4.6}$$

PROOF With (4.1)–(4.4), the integrand in (3.23) becomes a product of Gaussians, which may be integrated using the standard formula (D.3).

Substituting (4.5) into (3.27) produces the following key integral to evaluate:

$$F(\mathbf{z},[a]) = \left(\prod_{\alpha\in[a]}\sqrt{|2\pi W_\alpha|}\right)\int\int\cdots \times\int\prod_{\alpha\in[a]}\left(\mathcal{N}(\xi,\mu_\alpha,V_0)\prod_{(s,i)\in\alpha}\mathcal{N}(y_i^s+\beta^s;\mu_\alpha,V_i^s)\right) \times\prod_{s\in S}\mathcal{N}(\beta^s;\beta_0^s,B^s)\mathcal{N}(\xi;\xi_0,V_\Xi)d\beta^1 d\beta^2\cdots d\beta^r d\xi. \tag{4.7}$$

Evaluating (4.7) yields the kinematic component $G(\mathbf{z},[a])$ of the association probability $\Pr([a]\,|\,\mathbf{z})$ in (3.25). Two formulas for $G(\mathbf{z},[a])$ will be given. First, Theorem 4.2 provides a formula based on the direct evaluation of (4.7). Theorem 4.3 then gives a more computationally efficient formula achieved by applying certain transformations to the first result.

THEOREM 4.2

$$G(\mathbf{z},[a]) = C\exp\left(-\frac{1}{2}\left(\kappa([a])+\sum_{\alpha\in[a]}\kappa_\alpha\right)\right), \tag{4.8}$$

where C is chosen so that $G(\mathbf{z},[a_0])=1$, and the costs $\kappa([a])$ and κ_α are defined by

$$\kappa([a]) = \log|\mathbf{U}| - \mathbf{b}^T\mathbf{U}^{-1}\mathbf{b}, \qquad \text{and} \tag{4.9}$$

$$\kappa_\alpha = \log|V_0W_\alpha^{-1}| - m_\alpha^T W_\alpha^{-1} m_\alpha. \tag{4.10}$$

The matrix $\mathbf{U}$ *and vector* $\mathbf{b}$ *have the following block structure:*

$$\mathbf{U} = \begin{pmatrix} U_{1,1} & U_{1,2} & \cdots & U_{1,r} & U_{1,\emptyset} \\ U_{2,1} & U_{2,2} & \cdots & U_{2,r} & U_{2,\emptyset} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ U_{r,1} & U_{r,2} & \cdots & U_{r,r} & U_{r,\emptyset} \\ U_{\emptyset,1} & U_{\emptyset,2} & \cdots & U_{\emptyset,r} & U_{\emptyset,\emptyset} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \\ b_\emptyset \end{pmatrix}. \tag{4.11}$$

The entries in $\mathbf{U}$ *are matrices defined as follows. For* $s_1, s_2 \in S$,

$$U_{s_1,s_2} = \delta_{s_1 s_2}U^{s_1} - \sum_{\alpha\in[a]} I_\alpha^{s_1} I_\alpha^{s_2} (V_{\iota_\alpha^{s_1}}^{s_1})^{-1} W_\alpha (V_{\iota_\alpha^{s_2}}^{s_2})^{-1}, \tag{4.12}$$

$$U_{s_1,\emptyset} = -\sum_{\alpha\in[a]} I_\alpha^{s_1}(V_{\iota_\alpha^{s_1}}^{s_1})^{-1} W_\alpha V_0^{-1}, \tag{4.13}$$

$$U_{\emptyset,s_2} = -\sum_{\alpha\in[a]} I_\alpha^{s_2} V_0^{-1} W_\alpha (V_{\iota_\alpha^{s_2}}^{s_2})^{-1},$$

$$U_{\emptyset,\emptyset} = n_D V_0^{-1} - \sum_{\alpha\in[a]} V_0^{-1} W_\alpha V_0^{-1}, \tag{4.14}$$

where $I_\alpha^s = 1$ if $s\in\bar{\alpha}$ (and ι_α^s denotes the (unique) i for which $(s,i)\in\alpha$ in this case), and $I_\alpha^s = 0$ when $s\notin\bar{\alpha}$. Similarly, for $s\in S$, the entries in $\mathbf{b}$ *are vectors defined by*

$$b_s = (B^s)^{-1}\beta_0^s - \sum_{\alpha\in[a]} I_\alpha^s (V_{\iota_\alpha^s}^s)^{-1}(y_{\iota_\alpha^s}^s - m_\alpha) \qquad \text{and}$$

$$b_\emptyset = V_0^{-1}\sum_{\alpha\in[a]} m_\alpha. \tag{4.16}$$

The components of $\mathbf{U}$ *and* $\mathbf{b}$ *depend on the following quantities:*

$$W_\alpha = \left(V_0^{-1} + \sum_{(s,i)\in\alpha}(V_i^s)^{-1}\right)^{-1}, \tag{4.17}$$

$$m_\alpha = W_\alpha \sum_{(s,i)\in\alpha}(V_i^s)^{-1} y_i^s \qquad \text{and}$$

$$U^s = (B^s)^{-1} + \sum_{i=1}^{n^s}(V_i^s)^{-1}. \tag{4.18}$$

PROOF Equation (4.7) is more complicated than it appears: the notation μ_α, defined in (4.6), conceals a dependence on each of the integration variables in many of the Gaussian factors. Nevertheless, (4.7) is just an integral of products of Gaussians, where the arguments of the Gaussians are linear combinations of the integration variables. Equation (D.10) provides a formula for integrals of this form. To use this formula, we first must explicitly cast (4.7) in the form of (D.4). The Gaussians in (4.7) are of four types: type 1 is $\mathcal{N}(\xi;\mu_\alpha,V_0)$, indexed by $\alpha\in[a]$; type 2 is $\mathcal{N}(y_i^s+\beta^s;\mu_\alpha,V_i^s)$, indexed by $(s,i)\in\alpha$ and $\alpha\in[a]$; type 3 is $\mathcal{N}(\beta^s;\beta_0^s,B^s)$, indexed by $s\in S$, and type 4 is the single Gaussian $\mathcal{N}(\xi;\xi_0,V_\Xi)$. The integration variables in (4.7) are of two types: type a is β^{s_1}, indexed by $s_1\in S$; and type b is the single variable ξ. Thus, there are eight cases for the quantity A_{ij} in (D.4). In case 1a, for example, A_{α,s_1} is the coefficient of β^{s_1} in the exponent of $\mathcal{N}(\xi;\mu_\alpha,V_0)$, i.e., $A_{\alpha,s_1} = -I_\alpha^{s_1} W_\alpha (V_{\iota_\alpha^{s_1}}^{s_1})^{-1}$. The quantity m_i in (D.4) is simpler: there are only four cases to consider. For example, in case 2, $m_{(s,i,\alpha)}$ is the constant term in the exponent of $\mathcal{N}(y_i^s+\beta^s;\mu_\alpha,V_i^s)$, i.e., $m_{(s,i,\alpha)} = m_\alpha - y_i^s$. After A_{ij} and m_i have been expressed explicitly for each case, one may use (D.5)–(D.7) to obtain formulas for for $\mathbf{U}$, $\mathbf{b}$, and c, letting $V_\Xi\to\infty$ in each. The results for $\mathbf{U}$ and $\mathbf{b}$ are given in (4.11)–(4.16). The expression for c simplifies to

$$c = -\sum_{\alpha\in[a]} m_\alpha^T W_\alpha^{-1} m_\alpha + \sum_{s\in S}\sum_{i=1}^{n^s}(y_i^s)^T(V_i^s)^{-1}y_i^s + \sum_{s\in S}(\beta_0^s)^T(B^s)^{-1}\beta_0^s. \tag{4.19}$$

Invoking (D.10) to evaluate (4.7) now yields the result (4.8).

The need to invert the $(r+1)\times(r+1)$ block matrix $\mathbf{U}$ may seem rather excessive, and indeed it is. The next theorem reduces this requirement to one of inverting an $(r-1)\times(r-1)$ block matrix. Thus in the two-sensor case there is no need to form oversize matrices at all. Another problem with inverting $\mathbf{U}$ is that it is singular in the important case of all the bias prior covariances B^s being diffuse. Although this can be dealt with easily as a special case (by deleting the final row and column blocks of $\mathbf{U}$ and the final block of $\mathbf{b}$), the formula is ill-conditioned for large B^s. The next theorem provides a well conditioned formula.

THEOREM 4.3 *If the measurements* $\mathbf{z}$ *have been preprocessed so that* $\beta_0^s = 0$ *for all* $s \in S$*, then the definition of* $\kappa([a])$ *in* (4.9) *may be replaced with*

$$\kappa([a]) = \log(|\mathbf{W}_-^*|\,|H|) - (\tilde{\mathbf{b}}_-^{*T}\mathbf{W}_-^{*-1}\tilde{\mathbf{b}}_-^* + h^T H^{-1} h), \tag{4.20}$$

where the quantities involved are defined as follows. First,

$$\mathbf{W}_-^* = \tilde{\mathbf{U}}_{0-}^* + \mathbf{D}_-^*, \tag{4.21}$$

where $\mathbf{D}_-^*$ *encapsulates the dependence on the bias prior information:*

$$\begin{aligned} \mathbf{D}_-^* &= (D_{ss'}^*)_{s,s'=1}^{r-1} \qquad \text{with} \\ D_{ss'}^* &= \delta_{ss'}(B^s)^{-1} - (B^s)^{-1} D^{-1} (B^{s'})^{-1}, \\ D &= \sum_{s\in S} (B^s)^{-1}. \end{aligned} \tag{4.22}$$

The other quantities are

$$\begin{aligned} \tilde{\mathbf{U}}_{0-}^* &= (F_{ss'} - G_s H^{-1} G_{s'}^T)_{s,s'=1}^{r-1} \qquad \text{and} \\ \tilde{\mathbf{b}}_-^* &= (g_s - G_s H^{-1} h)_{s=1}^{r-1}, \end{aligned} \tag{4.23}$$

which are defined in terms of

$$F_{ss'} = \delta_{ss'} \sum_{i=1}^{n^s} (V_i^s)^{-1} - \sum_{\alpha\in[a]} I_\alpha^s I_\alpha^{s'} (V_{\iota_\alpha^s}^s)^{-1} W_\alpha (V_{\iota_\alpha^{s'}}^{s'})^{-1}, \tag{4.24}$$

$$G_s = -\sum_{\alpha\in[a]} I_\alpha^s (V_{\iota_\alpha^s}^s)^{-1} W_\alpha, \quad g_s = \sum_{\alpha\in[a]} I_\alpha^s (V_{\iota_\alpha^s}^s)^{-1} (m_\alpha - y_{\iota_\alpha^s}^s), \tag{4.25}$$

$$H = n_D V_0 - \sum_{\alpha\in[a]} W_\alpha \qquad \text{and} \qquad h = \sum_{\alpha\in[a]} m_\alpha. \tag{4.26}$$

PROOF Equation (4.20) is based on two reductions. The first eliminates the $\emptyset$ (or ξ) component from the block matrix $\mathbf{U}$. The second transforms the coordinate system of the r absolute biases to that of the $r-1$ relative biases and the sum of the biases, then eliminates the sum-of-biases component as well.

We begin the simplification with the following block decompositions of $\mathbf{U}$ and $\mathbf{b}$ into their (r-dimensional) $\boldsymbol{\beta}$ part and their (1-dimensional) ξ part:

$$\mathbf{U} = \begin{pmatrix} \tilde{\mathbf{U}}_0 + \mathbf{B}^{-1} & \tilde{\mathbf{U}}_\emptyset \\ \tilde{\mathbf{U}}_\emptyset^T & U_{\emptyset,\emptyset} \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} \tilde{\mathbf{b}} \\ b_\emptyset \end{pmatrix}, \tag{4.27}$$

where $\mathbf{B}$ denotes the $r\times r$ block diagonal matrix of bias covariance matrices B^s. From (4.12)–(4.16) we find that $\tilde{\mathbf{U}}_\emptyset = -\tilde{\mathbf{U}}_0\mathbf{1}$, $U_{\emptyset,\emptyset} = \mathbf{1}^T\tilde{\mathbf{U}}_0\mathbf{1}$, and $b_\emptyset = -\mathbf{1}^T\tilde{\mathbf{b}}$, where $\mathbf{1}$ denotes the $r\times 1$ block matrix of identity matrices. We may now eliminate the ξ component by applying (D.19) with the $\emptyset$ component in the role of the integrated variable I:

$$\mathbf{b}^T\mathbf{U}^{-1}\mathbf{b} = b_\emptyset^T U_{\emptyset,\emptyset}^{-1} b_\emptyset + \tilde{\mathbf{b}}^{*T}\tilde{\mathbf{U}}^{*-1}\tilde{\mathbf{b}}^* \qquad \text{and} \qquad |\mathbf{U}| = |U_{\emptyset,\emptyset}|\,|\tilde{\mathbf{U}}^*|. \tag{4.28}$$

In these equations, $\tilde{\mathbf{U}}^*$ and $\tilde{\mathbf{b}}^*$ are given by (D.14) and (D.15). To keep the bias prior information separated out, we write $\tilde{\mathbf{U}}^* = \tilde{\mathbf{U}}_0^* + \mathbf{B}^{-1}$, with

$$\tilde{\mathbf{U}}_0^* = \tilde{\mathbf{U}}_0 - \tilde{\mathbf{U}}_\emptyset U_{\emptyset,\emptyset}^{-1}\tilde{\mathbf{U}}_\emptyset^T \qquad \text{and} \qquad \tilde{\mathbf{b}}^* = \tilde{\mathbf{b}} - \tilde{\mathbf{U}}_\emptyset U_{\emptyset,\emptyset}^{-1} b_\emptyset. \tag{4.29}$$

For the second reduction, we simplify the expressions $\tilde{\mathbf{b}}^{*T}\tilde{\mathbf{U}}^{*-1}\tilde{\mathbf{b}}^*$ and $|\tilde{\mathbf{U}}^*|$ in (4.28). To do this, we transform the problem from the coordinates of the bias vector $\boldsymbol{\beta}$ to those of a vector $\mathbf{C}\boldsymbol{\beta}$ comprising the $r-1$ relative biases $\beta^s - \beta^r$, and the sum of the biases $\beta^1 + \beta^2 + \cdots + \beta^r$. The $r\times r$ block matrix $\mathbf{C}$ which accomplishes this, and its inverse, are

$$\mathbf{C} = \begin{pmatrix} \mathbf{I}_- & -\mathbf{1}_- \\ \mathbf{1}_-^T & I \end{pmatrix} \qquad \text{and} \qquad \mathbf{C}^{-1} = \frac{1}{r}\begin{pmatrix} r\mathbf{I}_- - \mathbf{1}_-\mathbf{1}_-^T & \mathbf{1}_- \\ -\mathbf{1}_-^T & I \end{pmatrix}, \tag{4.30}$$

where $\mathbf{I}_-$ is the $(r-1)\times(r-1)$ block identity matrix, and $\mathbf{1}_-$ is the $(r-1)\times 1$ block matrix of identity matrices. To simplify $\tilde{\mathbf{b}}^{*T}\tilde{\mathbf{U}}^{*-1}\tilde{\mathbf{b}}^*$ and $|\tilde{\mathbf{U}}^*|$, we work instead with the transformed quantities $\mathbf{W} = \mathbf{C}^{-T}\tilde{\mathbf{U}}^*\mathbf{C}^{-1}$ and $\mathbf{d} = \mathbf{C}^{-T}\tilde{\mathbf{b}}^*$, breaking them into the same blocks as $\mathbf{C}$:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_- & \mathbf{W}_r \\ \mathbf{W}_r^T & W_{rr} \end{pmatrix}, \qquad \mathbf{d} = \begin{pmatrix} \mathbf{d}_- \\ 0 \end{pmatrix}. \tag{4.31}$$

After some manipulation, we find

$$\tilde{\mathbf{b}}^{*T}\tilde{\mathbf{U}}^{*-1}\tilde{\mathbf{b}}^* = \mathbf{d}^T\mathbf{W}^{-1}\mathbf{d} = \mathbf{d}_-^T\mathbf{W}_-^{*-1}\mathbf{d}_- \qquad \text{and} \tag{4.32}$$

$$|\tilde{\mathbf{U}}^*| = |\mathbf{C}|^2|\mathbf{W}| = |\mathbf{C}|^2|W_{rr}|\,|\mathbf{W}_-^*|, \tag{4.33}$$

where (D.19) gives $\mathbf{W}_-^* = \mathbf{W}_- - \mathbf{W}_r W_{rr}^{-1}\mathbf{W}_r^T$. The original $\mathbf{U}$ and $\mathbf{b}$ quantities may now be written in terms of much simpler quantities:

$$\mathbf{b}^T\mathbf{U}^{-1}\mathbf{b} = \mathbf{d}_-^T\mathbf{W}_-^{*-1}\mathbf{d}_- + b_\emptyset^T U_{\emptyset,\emptyset}^{-1} b_\emptyset \qquad \text{and} \tag{4.34}$$

$$|\mathbf{U}| = |\mathbf{W}_-^*|\,|U_{\emptyset,\emptyset}|. \tag{4.35}$$

The result follows directly. The value of $\kappa([a])$ in (4.20) differs from that in (4.9) by $\log|D| - 2\log|V_0|$, but this can be absorbed into the normalizing constant C.

The expression (4.22) simplifies in certain special cases. When all the bias covariances B^s are diffuse, $\mathbf{D}^*_- = 0$, and in the two-sensor case, $\mathbf{D}^*_- = (B^1 + B^2)^{-1}$.

5. NON-KINEMATIC COMPONENTS

Equation (3.29) gives a simple formula for $R^{J\alpha}(\mathbf{z}^J)$, the contribution to association probability due to all non-kinematic data. (We will henceforth drop the superscript J.) Whereas the results of Section 4 are theorems that can be used "off-the-shelf," the category of non-kinematic data is too diverse to allow results that are this explicit yet broadly applicable. The explicit evaluation of $R^\alpha(\mathbf{z})$ using (3.29) depends on the statistical characteristics of the data, so one must be prepared to derive an appropriate formula for $R^\alpha(\mathbf{z})$ for one's particular problem. In this section we will present examples for various illustrative cases. These cases cover a range of possible types of non-kinematic data. They may be used directly for certain applications, modified for others, or referred to for guidance in developing appropriate formulas for applications further afield. Section 5.1 discusses how to handle object classification information in conjunction with feature data which may depend on object type. Sections 5.2 and 5.3 provide general methods for handling noisy and missing data, respectively.

5.1. Object-Type-Dependent Features

We will use the term "object type" to refer to a specific kind of feature: a discrete one, representing a finite number of classes to which the object could belong. It is typically measured by a classifier. The possible classification calls c are often the same as the possible object types t, but for the purposes of data association there is no requirement that the two be related. The quality of a classifier is determined by its confusion matrix. We use $L^s(c \mid t)$ to denote the confusion matrix entries for sensor s: i.e., the probability that an object of type t will be classified as c by sensor s.

Now consider a joint feature whose state space is parameterized by a state $x = (t, y)$, where t is the object type and y is the state of some other feature. Similarly, we decompose a measurement z into its classification c and the measurement w provided by a feature extractor attempting to measure y: i.e., $z = (c, w)$.

Equation (3.29) may be re-written

$$R^\alpha(\mathbf{c},\mathbf{w}) = \frac{P^\alpha(\mathbf{c},\mathbf{w})}{\prod_{(s,i)\in\alpha} P^s(c_i^s, w_i^s)}, \tag{5.1}$$

where (3.24) is now

$$P^\alpha(\mathbf{c},\mathbf{w}) = \sum_t \int p^0(t,y) \prod_{(s,i)\in\alpha} P_{\mathrm{D}}^s(t,y) L^s(c_i^s, w_i^s \mid t,y) \times \prod_{s\notin\bar{\alpha}} Q_{\mathrm{D}}^s(t,y)\,dy. \tag{5.2}$$

and $P^s(c_i^s, w_i^s)$ denotes $P^\alpha(\mathbf{c},\mathbf{w})$ for the special case $\alpha = \{(s,i)\}$. To evaluate (5.2) we need to specify what assumptions we are making about p^0, L^s, and P_{D}^s. Section 5.1.1 describes the simplest case in which R^α splits into $\mathbf{c}$ and $\mathbf{w}$ components. Section 5.1.2 describes a more interesting case in which the distribution of a measurement w depends not only on y, but on the object type t as well.

5.1.1. Independent Case

Suppose the following independence assumptions for p^0 and L^s are applicable,

$$p^0(t,y) = p^0(t)p^0(y), \tag{5.3}$$

$$L^s(c_i^s, w_i^s \mid t,y) = L^s(c_i^s \mid t) L^s(w_i^s \mid y). \tag{5.4}$$

(We are being somewhat cavalier here in the overloading of the notation p^0 and L^s, but trust that the meaning is clear in context because of the symbols used in their arguments.) Also suppose that P_{D}^s depends only on the object type t:

$$P_{\mathrm{D}}^s(t,y) = P_{\mathrm{D}}^s(t). \tag{5.5}$$

(Generalizing this to, say, $P_{\mathrm{D}}^s(t,y) = P_{\mathrm{D}}^s(t)P_{\mathrm{D}}^s(y)$ would cause complications because $Q_{\mathrm{D}}^s(t,y)$ would not enjoy the same simple multiplicative form.) With these assumptions, we find

$$R^\alpha(\mathbf{c},\mathbf{w}) = R^{C\alpha}(\mathbf{c}) R^{W\alpha}(\mathbf{w}), \tag{5.6}$$

where

$$R^{C\alpha}(\mathbf{c}) = \frac{P^{C\alpha}(\mathbf{c})}{\prod_{(s,i)\in\alpha} P^{Cs}(c_i^s)} \quad \text{and} \qquad R^{W\alpha}(\mathbf{w}) = \frac{P^{W\alpha}(\mathbf{w})}{\prod_{(s,i)\in\alpha} P^{Ws}(w_i^s)}, \tag{5.7}$$

with

$$P^{C\alpha}(\mathbf{c}) = \sum_t p^0(t) \prod_{(s,i)\in\alpha} P_{\mathrm{D}}^s(t) L^s(c_i^s \mid t) \prod_{s\notin\bar{\alpha}} Q_{\mathrm{D}}^s(t) \quad \text{and} \tag{5.8}$$

$$P^{W\alpha}(\mathbf{w}) = \int p^0(y) \prod_{(s,i)\in\alpha} L^s(w_i^s \mid y)\,dy, \tag{5.9}$$

and with $P^{Cs}(c_i^s)$ and $P^{Ws}(w_i^s)$ being the usual shorthand for the $\alpha = \{(s,i)\}$ case. Also note that the object-type–dependent detection probability leads to the following form for the non-detection parameter q:

$$q = \sum_t p^0(t) \prod_{s\in S} Q_{\mathrm{D}}^s(t). \tag{5.10}$$

In the absence of object-type data $\mathbf{c}$, $R^{C\alpha}(\mathbf{c})$ simplifies because $L^s(c_i^s \mid t)$ may be replaced by 1 in (5.8). When there is, furthermore, no object-type component of state t, $R^{C\alpha}(\mathbf{c})$ simplifies further:

$$R^{C\alpha} = \frac{\prod_{s\in\bar{\alpha}} P_{\mathrm{D}}^s \prod_{s\notin\bar{\alpha}} Q_{\mathrm{D}}^s}{\prod_{s\in\bar{\alpha}} \left(P_{\mathrm{D}}^s \prod_{s'\neq s} Q_{\mathrm{D}}^{s'}\right)} = q^{1-|\alpha|}, \qquad (5.11)$$

where q is simply the product of the Q_{D}^s over all $s \in S$, rather than (5.10).

The form (5.6) is quite convenient, as it allows one to separate the computations dealing with the object-type C data from that of the other feature W. However, it is often the case that objects of different types will have substantially different feature statistics. For example, objects of one type t_1 may have a certain prior distribution $p^0(y \mid t_1)$ on feature values as well has a certain measurement likelihood function $L^s(w \mid y, t_1)$, while objects of another type t_2 may not even have the feature in question. In cases like this, we need a more accommodating approach.

5.1.2. Dependent Case

Instead of making assumptions (5.3) and (5.4), we write this general decomposition of p^0 and L^s, which makes no assumptions—just manipulations of conditional probabilities:

$$p^0(t,y) = p^0(t)p^0(y \mid t), \qquad (5.12)$$

$$L^s(c_i^s, w_i^s \mid t, y) = L^s(c_i^s \mid t, y)L^s(w_i^s \mid c_i^s, t, y). \qquad (5.13)$$

We now make two mild assumptions about (5.13) in place of the radical assumption made in (5.4). First, we assume that $L^s(c_i^s \mid t, y)$ does not depend on y. Although the classifier behavior may in fact depend on the feature state y, we assume that this is a minor effect. This is convenient because although one expects a confusion matrix $L^s(c \mid t)$ to be provided with a classifier for a given sensor configuration, one is unlikely to be provided with the dependence of the classifier on y. Second, we assume that $L^s(w_i^s \mid c_i^s, t, y)$ does not depend on c_i^s. In other words, we assume that the true object type suffices to determine how the measurement w_i^s depends on the feature state y, and that the called type yields little additional information. With these assumptions, (5.13) simplifies to

$$L^s(c_i^s, w_i^s \mid t, y) = L^s(c_i^s \mid t)L^s(w_i^s \mid t, y). \qquad (5.14)$$

Finally, as in (5.5) of Section 5.1.1 we assume that the detection probability depends only on the object type: $P_{\mathrm{D}}^s(t,y) = P_{\mathrm{D}}^s(t)$.

Using (5.12), (5.14), and (5.5) we may simplify (5.2) to a relatively simple sum over t,

$$P^\alpha(\mathbf{c}, \mathbf{w}) = \sum_t p^0(t) \prod_{(s,i)\in\alpha} P_{\mathrm{D}}^s(t) L^s(c_i^s \mid t) \prod_{s\notin\bar{\alpha}} Q_{\mathrm{D}}^s(t) P^\alpha(\mathbf{w} \mid t), \qquad (5.15)$$

where

$$P^\alpha(\mathbf{w} \mid t) = \int p^0(y \mid t) \prod_{(s,i)\in\alpha} L^s(w_i^s \mid t, y)dy. \qquad (5.16)$$

The formula (5.10) for q also holds in this case. Thus this object-type–dependent case requires one to evaluate an analog of (5.9) for each object type t—namely (5.16). The result of this is coupled with the classification data, via (5.15), resulting in a calculation which is somewhat more complicated than (5.8), but is nevertheless rather straightforward, especially considering its much greater generality and realism.

It remains to evaluate the integral in (5.16). In doing so, we will first consider two general phenomena that affect its evaluation. Section 5.2 will address the issue that feature extraction is often a noisy procedure, and propose a robust, general principle for coping with this. Finally, Section 5.3 will describe how to cope with the fact that features might be missing and/or assessed to be missing.

5.2. Non-Informative Noise

Let us consider the object type t in (5.16) fixed in this section, and suppress it in the notation. This is equivalent to considering (5.9) instead (suppressing the W from the notation). Either interpretation yields

$$P^\alpha(\mathbf{w}) = \int p^0(y) \prod_{(s,i)\in\alpha} L^s(w_i^s \mid y)dy. \qquad (5.17)$$

A typical feature model one might use for $L^s(w \mid y)$ is a Gaussian. Such a model carries the risk of returning incredibly tiny assessments of probability density for a measurement w arising (e.g., 10^{-100}, 10^{-1000}, or smaller) when it doesn't match y well. In a realistic situation, the probability density of a measurement w could never be that small because there is always the possibility of some glitch in the feature extraction routine. By allowing such tiny probability densities to occur in the model, one runs the risk of the feature extractor completely preventing a pair of tracks being associated even when the kinematic information is extremely favorable to association. Because one of the chief fears in incorporating a feature extractor into an association algorithm is that it might ruin kinematic-only performance that is already fairly good, it is prudent to account for the possibility of noise in the feature measurement model. (Note: one could make the same argument to point out that the rapid decay of Gaussians allows anomalous kinematic data to override perfect feature matches, so one might include a noise term in the kinematic association terms as well. This would be a point worth considering when feature extraction technology reaches the maturity of kinematic tracking.)

We regard a function as representing pure noise when the distribution of measurements w it yields is independent of the actual feature state y. One option for a noise model is a uniform distribution of w over a certain range. Although this seems simple, it introduces an additional parameter (the width of the distribution) and complicates the required integrals, while not necessarily being a good model of noise. Instead, we promote the use of a *non-informative* model for noise. This model sets the distribution of measurements due to noise equal to the overall distribution of valid measurements. It is called non-informative because a measurement w provides no information as to whether it arose from a valid measurement of some object or merely from noise. Were one to know how noise differs statistically from valid measurements, one could use this information to flag certain measurements w as more likely to have arisen from noise than others, and perhaps squeeze even more performance out of an association algorithm, but at the risk of algorithm robustness should the noise behave differently than expected. In contrast, the non-informative assumption provides a conservative, robust baseline model for noise.

We use the non-informative assumption as follows. Rather than developing a model for the measurement likelihood function $L^s(w \mid y)$ directly, we develop a model $L^s_\mu(w \mid y)$ for how we expect the measurement w to be distributed given that (a) the true feature value is y, and (b) the measurement is actually behaving according to the model. We let a^s_μ denote the probability that the feature is obeying the model and express the overall measurement likelihood function $L^s(w \mid y)$ as

$$L^s(w \mid y) = a^s_\mu L^s_\mu(w \mid y) + (1 - a^s_\mu) P^s_\nu(w), \quad (5.18)$$

where $P^s_\nu(w)$ is the distribution of w given that it arises from noise. Note that there is no dependence on the true state y in this case. We now define the following analog of (5.17) for L^s_μ:

$$P^\alpha_\mu(\mathbf{w}) = \int p^0(y) \prod_{(s,i)\in\alpha} L^s_\mu(w^s_i \mid y) dy. \quad (5.19)$$

The special case $\alpha = \{(s,i)\}$ is particularly important here. Thinking of w as representing w^s_i here, we define

$$P^s_\mu(w) = \int p^0(y) L^s_\mu(w \mid y) dy. \quad (5.20)$$

Equation (5.20) expresses the overall distribution $P^s_\mu(w)$ of w values arising on sensor s when the features are "obeying the model." The non-informative noise assumption is that the distribution in the noise case $P^s_\nu(w)$ is identical to $P^s_\mu(w)$:

$$P^s_\nu(w) = P^s_\mu(w). \quad (5.21)$$

Using this, and substituting the expression in (5.18) for $L^s(w \mid y)$ into (5.17) yields

$$\begin{aligned} P^\alpha(\mathbf{w}) &= \sum_{\alpha'\subseteq\alpha} A^{\bar\alpha,\bar\alpha'}_\mu P^{\alpha'}_\mu(\mathbf{w}) \prod_{(s,i)\in\alpha\setminus\alpha'} P^s_\mu(w^s_i) \\ &= \sum_{\alpha'\subseteq\alpha \mid \alpha'\mid\geq 2} A^{\bar\alpha,\bar\alpha'}_\mu P^{\alpha'}_\mu(\mathbf{w}) \prod_{(s,i)\in\alpha\setminus\alpha'} P^s_\mu(w^s_i) \\ &\quad + C^{\bar\alpha}_\mu \prod_{(s,i)\in\alpha} P^s_\mu(w^s_i), \end{aligned} \quad (5.22)$$

where

$$A^{\bar\alpha,\bar\alpha'}_\mu = \prod_{s\in\bar\alpha'} a^s_\mu \prod_{s\in\bar\alpha\setminus\bar\alpha'} (1 - a^s_\mu) \qquad \text{and} \qquad C^{\bar\alpha}_\mu = \sum_{\bar\alpha'\subseteq\bar\alpha \mid \bar\alpha'\mid\leq 1} A^{\bar\alpha,\bar\alpha'}_\mu. \quad (5.23)$$

Equation (5.22) simplifies significantly in the two-sensor case. Letting $\alpha = \{(1,i),(2,j)\}$, it reduces to

$$P^\alpha(\mathbf{w}) = a^1_\mu a^2_\mu P^\alpha_\mu(\mathbf{w}) + (1 - a^1_\mu a^2_\mu) P^1_\mu(w^1_i) P^2_\mu(w^2_j). \quad (5.24)$$

The formula (5.22) (or (5.24)) applies both to the relatively simple case of Section 5.1.1, and the more complicated case of Section 5.1.2. In the latter case, it should be interpreted as a formula for $P^\alpha(\mathbf{w} \mid t)$, the dependence on object type t having been suppressed. In this case, one would use (5.19) to compute $P^\alpha_\mu(\mathbf{w} \mid t)$ from $p^0(y \mid t)$ and $L^s_\mu(w \mid t, y)$, and the parameters a^s_μ may (or may not) depend on t as well. The resulting formulas for $P^\alpha(\mathbf{w} \mid t)$ for each t would then be used in (5.15) in place of (5.16). In the two-sensor case, this amounts to a fairly trivial modification.

Section 5.1.1 treats the case where the feature is independent of object type. In this case, the result may be expressed in terms of a quantity $R^\alpha_\mu(\mathbf{w})$ which represents what we would use if we ignored noise modeling:

$$R^\alpha_\mu(\mathbf{w}) = \frac{P^\alpha_\mu(\mathbf{w})}{\prod_{(s,i)\in\alpha} P^s_\mu(w^s_i)}. \quad (5.25)$$

The inclusion of noise modeling yields the following forms for $R^\alpha(\mathbf{w})$,

$$\begin{aligned} R^\alpha(\mathbf{w}) &= \sum_{\alpha'\subseteq\alpha \mid \alpha'\mid\geq 2} A^{\bar\alpha,\bar\alpha'}_\mu R^{\alpha'}_\mu(\mathbf{w}) + C^{\bar\alpha}_\mu, \qquad \text{or} \\ R^\alpha(\mathbf{w}) &= a^1_\mu a^2_\mu R^\alpha_\mu(\mathbf{w}) + (1 - a^1_\mu a^2_\mu), \end{aligned} \quad (5.26)$$

in the general and two-sensor cases, respectively. The two-sensor formula has a pleasing interpretation as a convex combination of the model value $R^\alpha_\mu(\mathbf{w})$ and the neutral value 1.

To use the non-informative noise model, then, one takes the formula one has developed for $P^\alpha(\mathbf{w} \mid t)$ or $R^\alpha(\mathbf{w})$, renames it $P^\alpha_\mu(\mathbf{w} \mid t)$ or $R^\alpha_\mu(\mathbf{w})$, respectively, then uses the appropriate equation above ((5.22), (5.24), or (5.26)) to incorporate the possibility that real-world measurements may not obey one's model. This is a nice

form of modularity, which allows us to develop formulas without explicitly considering the effect of noise: it can be incorporated with any model one develops. Similarly, one may develop models for various object types t and combine them using (5.15). Next we develop a similarly modular capability for handling the possibility of missing features.

5.3. Missing Features

In this subsection we consider the complication that a feature may not be present on every object. If the feature extractor cannot handle this situation, but returns some meaningless value when the feature is absent, one can use the non-informative noise assumption of Section 5.2. However, when the feature extractor exhibits a distinctive behavior for objects that lack the feature, then it is advantageous to exploit this information in a way that the non-informative noise model does not. Here we assume that either the designer of the feature model has created a flag that indicates a belief that the feature is absent, or that some post-processing of the output produces such a flag. In either case, the performance would undoubtedly be imperfect, and may be characterized by a detection probability P_{d}^s (the probability of the feature being declared present on sensor s when it is indeed present) and a false alarm probability P_{fa}^s (the probability of the feature being declared present when it is in fact missing). We use the lower case d to distinguish the missing feature detection probability P_{d}^s here from the object detection probability P_{D}^s.

We extend the state space for the feature to account for the possibility of it being missing. If y denotes the feature state, then we distinguish between two types of values it can take: y_e, which denotes a value for a feature that exists, and $\emptyset$, which denotes that the feature is missing. The prior $p^0(y)$ on the feature state may be expressed as

$$p^0(y) = \begin{cases} P_{\mathrm{E}} p_{\mathrm{e}}^0(y_e) & \text{for} \quad y = y_e, \\ Q_{\mathrm{E}} & \text{for} \quad y = \emptyset, \end{cases} \tag{5.27}$$

where P_{E} is the prior probability of the feature existing, $Q_{\mathrm{E}} = 1 - P_{\mathrm{E}}$, and $p_{\mathrm{e}}^0(y_e)$ is the prior distribution on y_e given that the feature exists.

We let w_i^s denote the feature measurement on sensor s. Like the feature state, the measurement can take two types of values: a proper value w_{pi}^s, or a call from the feature extractor on sensor s that it is missing, $\emptyset^s$. To express the measurement likelihood function $L^s(w_i^s \mid y)$ now requires four cases:

$$\begin{aligned} L^s(w_{pi}^s \mid y_e) &= P_{\mathrm{d}}^s L_{\mathrm{d}}^s(w_{pi}^s \mid y_e), \\ L^s(w_{pi}^s \mid \emptyset) &= P_{\mathrm{fa}}^s L_{\mathrm{fa}}^s(w_{pi}^s), \end{aligned} \tag{5.28}$$

$$\begin{aligned} L^s(\emptyset^s \mid y_e) &= Q_d^s, \\ L^s(\emptyset^s \mid \emptyset) &= Q_{\mathrm{fa}}^s, \end{aligned} \tag{5.29}$$

where $Q_{\mathrm{d}}^s = 1 - P_{\mathrm{d}}^s$, $Q_{\mathrm{fa}}^s = 1 - P_{\mathrm{fa}}^s$, $L_{\mathrm{d}}^s(w_{pi}^s \mid y_e)$ is the measurement likelihood function given that the feature exists and measurement is proper, and $L_{\mathrm{fa}}^s(w_{pi}^s)$ is the measurement likelihood function given that the feature does not exist but the measurement is proper. These formulas are somewhat different from those in [11], where it is assumed that a "false alarm" can occur even in the case where the feature exists.

We now evaluate the integral in (5.9), suppressing the superscript W:

$$P^\alpha(\mathbf{w}) = \int p^0(y) \prod_{(s,i)\in\alpha} L^s(w_i^s \mid y) dy. \tag{5.30}$$

Let α_p denote the subset of $(s,i) \in \alpha$ for which w_i^s is a proper measurement, and $\alpha_\emptyset$ denote the subset for which $w_i^s = \emptyset^s$. We make a number of definitions similar to those of Section 5.2 for non-informative noise. Analogously to (5.19) we define

$$P_{\mathrm{d}}^{\alpha_p}(\mathbf{w}) = \int p_{\mathrm{e}}^0(y_e) \prod_{(s,i)\in\alpha_p} L_{\mathrm{d}}^s(w_{pi}^s \mid y_e) dy_e \tag{5.31}$$

to be the version of $P^\alpha(\mathbf{w})$ one would use when not considering the possibility of missing features. Thinking of w as representing w_{pi}^s we write the special case $\alpha = \{(s,i)\}$ of (5.31) as

$$P_{\mathrm{d}}^s(w) = \int p_{\mathrm{e}}^0(y_e) L_{\mathrm{d}}^s(w \mid y_e) dy_e. \tag{5.32}$$

We employ the non-informative noise assumption to model $L_{\mathrm{fa}}^s(w_{pi}^s)$:

$$L_{\mathrm{fa}}^s(w_{pi}^s) = P_{\mathrm{d}}^s(w_{pi}^s). \tag{5.33}$$

With the above definitions, we may re-write (5.30) as

$$\begin{aligned} P^\alpha(\mathbf{w}) = P_{\mathrm{E}} &\left(\prod_{(s,i)\in\alpha_p} P_{\mathrm{d}}^s \prod_{(s,i)\in\alpha_\emptyset} Q_{\mathrm{d}}^s \right) P_{\mathrm{d}}^{\alpha_p}(\mathbf{w}) \\ &+ Q_{\mathrm{E}} \left(\prod_{(s,i)\in\alpha_p} P_{\mathrm{fa}}^s \prod_{(s,i)\in\alpha_\emptyset} Q_{\mathrm{fa}}^s \right) \prod_{(s,i)\in\alpha_p} P_{\mathrm{d}}^s(w_{pi}^s). \end{aligned} \tag{5.34}$$

This section has described how to incorporate object-type-dependence, noise, and missing features into the evaluation of the feature component of association probability. Ultimately, regardless of what effects are included, one must evaluate an integral of the form (5.30) where the prior $p^0(y)$ and the measurement likelihood function $L^s(w_i^s \mid y)$ are appropriate to some specific feature. A Gaussian model for $L^s(w_i^s \mid y)$ may be appropriate, in which case one can treat the feature in the same way as one-dimensional kinematic data. A more complex example is presented in Section 6, where an angle-valued measurement has a von Mises error distribution coupled with the possibility of being off by 180°. The example shows explicitly how to evaluate (5.30) in

this case, and then combine it with all three phenomena in this section to produce association probabilities.

6. EXAMPLE

6.1. Sample Calculation

This section provides an example of the computation of association probabilities in a complex scenario. It has three sensors with unknown biases, object-type classification data and object-type-dependent detection probabilities, and finally, noisy and possibly missing features whose distributions vary between object types. We perform this sample calculation for the following data set:

- Measurements:

$$s=1:\quad y_1^1=(1.5,3.7),\qquad y_2^1=(9.3,1.9),\qquad y_3^1=(13.2,-11.2),$$

$$s=2:\quad y_1^2=(-6.2,14.9),\qquad y_2^2=(-3.6,6.9),\qquad y_3^2=(3.0,4.5),$$

$$s=3:\quad y_1^3=(-16.8,-4.8),\qquad y_2^3=(-7.3,-11.1).$$

- Error covariances:

$$s=1:\quad V_1^1=\begin{pmatrix}1.58 & -0.12\\ -0.12 & 2.27\end{pmatrix},\quad V_2^1=\begin{pmatrix}8.0 & -2.0\\ -2.0 & 10.2\end{pmatrix},\quad V_3^1=\begin{pmatrix}8.7 & -5.8\\ -5.8 & 12.8\end{pmatrix},$$

$$s=2:\quad V_1^2=\begin{pmatrix}13.4 & -0.6\\ -0.6 & 4.0\end{pmatrix},\quad V_2^2=\begin{pmatrix}57. & -31.\\ -31. & 30.\end{pmatrix},\quad V_3^2=\begin{pmatrix}3.22 & -1.15\\ -1.15 & 1.76\end{pmatrix},$$

$$s=3:\quad V_1^3=\begin{pmatrix}1.32 & 0.52\\ 0.52 & 0.94\end{pmatrix},\quad V_2^3=\begin{pmatrix}13.6 & 10.7\\ 10.7 & 9.3\end{pmatrix}.$$

- Called types:

$$s=1:\quad c_1^1=\blacktriangle,\qquad c_2^1=\blacklozenge,\qquad c_3^1=\bullet,$$

$$s=2:\quad c_1^2=\blacklozenge,\qquad c_2^2=\blacklozenge,\qquad c_3^2=\blacksquare,$$

$$s=3:\quad c_1^3=\blacklozenge,\qquad c_2^3=\blacktriangleright.$$

- Feature measurements:

$$s=1:\quad w_1^1=\emptyset^1,\qquad w_2^1=5.5771,\qquad w_3^1=6.2067,$$

$$s=2:\quad w_1^2=4.9773,\qquad w_2^2=4.9101,\qquad w_3^2=6.2011,$$

$$s=3:\quad w_1^3=5.1253,\qquad w_2^3=3.0885.$$

The data set is visualized in Fig. 1. The measurements are labeled from left to right on each sensor. The kinematic error covariances are represented by 90% containment ellipses. The called object types for the classifiers

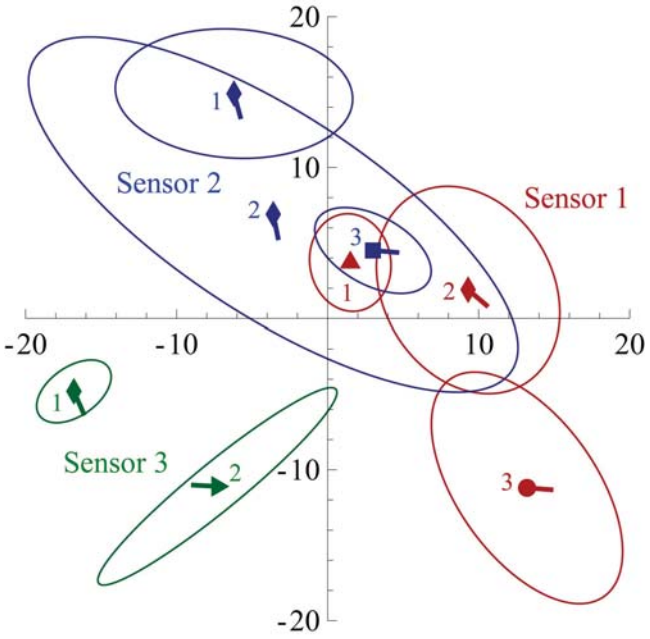

Fig. 1. Measurements and error covariances, with ellipses denoting 90% containment regions, symbols denoting object-type calls, and line segments denoting measured feature values (angles).

are the symbols used for plotting them. The feature measurements are angles ranging from 0 to 2π, represented by a small line segment, or the special value $\emptyset^s$ denoting a missing feature call by sensor s.

There are 778 possible associations for $\mathbf{n}=(3,3,2)$. To speak meaningfully of the probabilities of these associations given the data, one must make assumptions about the data generation process. Here, we make the following assumptions:

- Log-diffuse prior on number of objects (cf. Section B.1)
- Covariance of prior kinematic distribution:

$$V^0=\begin{pmatrix}8 & -9\\ -9 & 11\end{pmatrix}$$

- Diffuse bias priors for all sensors
- Four object types (■, ♦, ▲, and ▼) with statistics given in Tables I and II
- An object-type–dependent angular feature with statistics described below

The four ground-truth object types listed above may be thought of as the possible shapes of an object. The Bayesian methodology requires a prior distribution $p^0(t)$ on these types, which is given in Table I. Some object types may be easier to detect on a given sensor than others. Table I also gives the detection probability for each type on each sensor.

Although the prior probability is much larger for $t=\blacktriangledown$ than for other object types ($p^0(\blacktriangledown)=0.55$), (C.7) implies that the expected number $n_D(t)$ of each object type t detected on at least one sensor is roughly equal

TABLE I
Prior and Detection Probabilities for Object Types

	True type t ■	♦	▲	▼
$p^0(t)$	0.1	0.15	0.2	0.55

		True type t ■	♦	▲	▼
$P_D^s(t)$					
Sensor s	1	0.9	0.8	0.95	0.3
	2	0.8	0.99	0.7	0.9
	3	0.4	0.8	0.9	0.99

TABLE II
Confusion Matrices for Each Sensor

$L^1(c\|t)$		Called type c ■	♦	▲	▼	●
True type t	■	0.75	0.01	0.03	0.01	0.2
	♦	0.04	0.62	0.02	0.07	0.25
	▲	0.03	0.01	0.74	0.02	0.2
	▼	0.07	0.08	0.05	0.5	0.3

$L^2(c\|t)$		Called type c ■	♦	▲	▼
True type t	■	0.8	0.1	0.05	0.05
	♦	0.08	0.75	0.12	0.05
	▲	0.02	0.15	0.6	0.23
	▼	0.01	0.01	0.03	0.95

$L^3(c\|t)$		Called type c ■	♦	►
True type t	■	0.95	0.02	0.03
	♦	0.01	0.96	0.03
	▲	0.02	0.01	0.97
	▼	0.04	0.01	0.95

across types:

$$\mathbb{E}[n_D(\blacksquare) \mid \mathbf{n}] = 0.765169, \qquad \mathbb{E}[n_D(\blacktriangle) \mid \mathbf{n}] = 0.795727,$$

$$\mathbb{E}[n_D(\blacklozenge) \mid \mathbf{n}] = 0.763363, \qquad \mathbb{E}[n_D(\blacktriangledown) \mid \mathbf{n}] = 0.849989.$$

The measurement likelihood functions $L^s(c \mid t)$ for each sensor are given by the confusion matrices in Table II. In a real application, such values would be the result of training a classifier. Although the set of possible called types for a classifier is often identical to the set of ground truth types, only the sensor 2 classifier operates this way in the example. Sensor 1 has an additional called type ● representing "unknown," whereas Sensor 3 cannot distinguish between ▲ and ▼, so it issues a call ► which represents either one.

We now specify phenomenologically rich feature distributions for each of the object types above. First, we assume that the feature may be missing, and let $P_{\mathrm{E}}(t)$ be the prior probability that a proper, ground-truth feature value exists for an object of type t. The values of $P_{\mathrm{E}}(t)$ are given in Table III. Assuming a proper value y_e does exist, we assume it to be a uniformly distributed angular quantity: i.e., $p_{\mathrm{e}}^0(y_e \mid t) = 1/2\pi$.

The measurement likelihood function for object type t on sensor s is governed by five quantities: $a_\mu^s(t)$, $P_{\mathrm{d}}^s(t)$, $P_{\mathrm{fa}}^s(t)$, $p_{\mathrm{jump}}^s(t)$, $\sigma^s(t)$. These are given in Table III. As in Section 5.2, $a_\mu^s(t)$ represents the probability that the measurement w obeys the model described by the rest of the measurement likelihood function. The probabilities $P_{\mathrm{d}}^s(t)$ and $P_{\mathrm{fa}}^s(t)$ are the detection and false alarm probabilities for the decision problem of declaring a proper feature value or missing, as described in Section 5.3. Finally, we specify an underlying model for the non-noise, detection case based on the parameters $p_{\mathrm{jump}}^s(t)$ and $\sigma^s(t)$. The jump probability $p_{\mathrm{jump}}^s(t)$ is the probability that an additional π has been erroneously added to the feature measurement. The pseudo-standard–deviation $\sigma^s(t)$ is an analog to the standard deviation of a Gaussian in the case of a feature whose domain is the entire real line. The angular analog of a Gaussian is the Von Mises distribution [1]:

$$\mathcal{M}(\theta; \mu, \kappa) = \frac{e^{\kappa \cos(\theta - \mu)}}{2\pi I_0(\kappa)}, \tag{6.1}$$

where I_0 is the modified Bessel function of order 0. The κ in (6.1) is analogous to an inverse variance. Thus the distribution of a feature value w (before introducing the

TABLE III
Prior and Detection Probabilities for Object Types

	True type t			
	■	♦	▲	▼
$P_E(t)$	0.9	0.95	0.7	0.99

		True type t			
$a^s_\mu(t)$		■	♦	▲	▼
Sensor s	1	0.99	0.998	0.8	0.99
	2	0.8	0.999	0.9	$1-10^{-5}$
	3	0.99	0.99	0.95	$1-10^{-10}$

		True type t			
$P^s_d(t)$		■	♦	▲	▼
Sensor s	1	0.94	0.94	0.96	0.83
	2	0.99	0.99	0.98	0.97
	3	0.6	0.92	0.93	0.997

		True type t			
$P^s_{fa}(t)$		■	♦	▲	▼
Sensor s	1	0.02	0.04	0.03	0.002
	2	0.05	0.1	0.06	0.02
	3	0.001	0.01	0.02	0.3

		True type t			
$p^s_{\text{jump}}(t)$		■	♦	▲	▼
Sensor s	1	0.01	0.08	0	0.5
	2	0.02	0.15	0	0.5
	3	0.06	0.05	0.03	0.5

		True type t			
$\sigma^s(t)$		■	♦	▲	▼
Sensor s	1	0.1	0.23	0.05	0.02
	2	0.07	0.2	0.15	0.05
	3	0.05	0.3	0.08	0.03

additional effects of jump, missing features, and noise), is $\mathcal{M}(w; y, \sigma^s(t)^{-2})$.

The probability distribution is now fully specified, giving sufficient information to compute the probability of each possible association. In order to do so, we will use the following identity for the Von Mises distribution:

$$\frac{1}{2\pi}\int_0^{2\pi}\prod_{s=1}^{r}\mathcal{M}(\theta;\mu^s,\kappa^s)d\theta = I_0\left(\left|\sum_{s=1}^{r}\kappa^s\exp(i\mu^s)\right|\right)\Bigg/\prod_{s=1}^{r}2\pi I_0(\kappa^s). \tag{6.2}$$

We now compute the probability of one of the most plausible of the 778 possible associations, namely:

$$[a] = \{(1,1,1),(2,2,\emptyset),(3,3,2)\}$$

(which is shorthand for $[a] = \{\{(1,1),(2,1),(3,1)\}, \{(1,2),(2,2)\},\{(1,3),(2,3),(3,2)\}\}$. We begin by evaluating the quantities in (4.17):

$$W_{(1,1,1)} = \begin{pmatrix} 0.32753 & -0.08142 \\ -0.08142 & 0.33302 \end{pmatrix},$$

$$W_{(2,2,\emptyset)} = \begin{pmatrix} 2.78935 & -2.85439 \\ -2.85439 & 3.67734 \end{pmatrix},$$

$$W_{(3,3,2)} = \begin{pmatrix} 0.36425 & -0.08525 \\ -0.08525 & 0.33942 \end{pmatrix},$$

and, writing column vectors in rows for convenience,

$$m_{(1,1,1)} = (-4.87067, 3.66306),$$

$$m_{(2,2,\emptyset)} = (1.62393, -1.04784),$$

$$m_{(3,3,2)} = (3.09849, -1.43294).$$

These are then used to compute κ_α from (4.10) for each $\alpha \in [a]$:

$$\kappa_{(1,1,1)} = -87.4344,$$

$$\kappa_{(2,2,\emptyset)} = -0.2445,$$

$$\kappa_{(3,3,2)} = -23.8285.$$

The values of W_α and m_α above are also used to compute H, $\mathbf{W}^*_-$, h, and $\mathbf{b}^*_-$:

$$H = \begin{pmatrix} 20.519 & -23.979 \\ -23.979 & 28.650 \end{pmatrix},$$

$$\mathbf{W}^*_- = \begin{pmatrix} 0.56727 & 0.05332 & -0.18105 & -0.20300 \\ 0.05332 & 0.48860 & -0.10474 & -0.20623 \\ -0.18105 & -0.10474 & 0.34568 & 0.08847 \\ -0.20300 & -0.20623 & 0.08847 & 0.63862 \end{pmatrix},$$

$$h = (-0.14826, 1.18228),$$

$$\tilde{\mathbf{b}}^*_- = (-5.5339, -0.04816, -1.2648, -6.9713).$$

From these we have $|\mathbf{W}^*_-||H| = 0.473943$, and $\tilde{\mathbf{b}}^{*T}_- \cdot \mathbf{W}^{*-1}_- \tilde{\mathbf{b}}^*_- + h^T H^{-1} h = 247.8660$, so (4.20) yields $\kappa([a]) = -248.613$. Combining this in (4.8) with the values of κ_α we have

$$G^K(\mathbf{z}^K, [a]) = C \exp\left(-\frac{1}{2}\left(\kappa([a]) + \sum_{\alpha \in [a]} \kappa_\alpha\right)\right)$$
$$= 1.58147 \times 10^{78} C.$$

In addition to $G^K(\mathbf{z}^K, [a])$ there are two other contributions to the association probability in (3.25). We compute $g([a])$ from (B.10). This requires the probability q that an object is undetected on all sensors, which is given by (5.10):

$$q = \sum_t p^0(t) \prod_{s \in S} Q^s_{\mathrm{D}}(t) = 0.001945 \quad \text{(exactly)}.$$

Therefore $g([a]) = (2!/7!)0.998055^5 = 0.000392981$.

Finally, we compute the joint object-type–feature component $G^J(\mathbf{z}^J, [a])$, which is somewhat involved. Fortunately, (3.28) and (3.29) reduce this calculation to the computation of $P^{J\alpha}(\mathbf{z}^J)$ for each α, and there are only 47 possible sets α (compared to 778 associations). Because the feature distributions and detection probabilities are object-type–dependent, for any α, $P^{J\alpha}(\mathbf{z}^J)$ is given by $P^\alpha(\mathbf{c}, \mathbf{w})$ in (5.15) (the J being suppressed, and $\mathbf{z}^J$ expanded into $(\mathbf{c}, \mathbf{w})$). This, in turn, requires the computation of $P^\alpha(\mathbf{w} \mid t)$ in (5.16) for each object type t. We will compute $P^\alpha(\mathbf{w} \mid t)$ explicitly for one of the sets α in our example $[a]$, namely $\alpha = (3,3,2) = \{(1,3),(2,3),(3,2)\}$. To do so, we successively break down the computation in order to account for various phenomena—in each case expressing $P^\alpha(\mathbf{w} \mid t)$ in terms of some simpler version of $P^\alpha(\mathbf{w} \mid t)$ which does not include the phenomenon. Specifically, we do this for noise, then missing features, then the jump by π.

The first phenomenon to account for is noise, so we evaluate $P^\alpha(\mathbf{w})$ using (5.22), suppressing the dependence on t in the notation for brevity. This requires that we compute $P^{\alpha'}_\mu(\mathbf{w})$ for all eight subsets $\alpha' \subseteq \alpha$, where α' represents a hypothesis about which measurements in α arise from the model (as opposed to noise). To compute each $P^{\alpha'}_\mu(\mathbf{w})$, we turn to (5.34) to handle missing features. In this case, $\alpha_p = \alpha'$ for any $\alpha' \subseteq \alpha$ because w^s_i is proper (i.e., non-missing) for each $(s,i) \in \alpha$: $w^1_3 = 6.2067$, $w^2_3 = 6.2011$, and $w^3_2 = 3.0885$. This, in turn, requires us to evaluate $P^{\alpha'}_{\mathrm{d}}(\mathbf{w})$, accounting the the jump phenomenon. Unlike noise and missing features, we do not have a general equation to account for the possible jump by π, as it is a rather specific phenomenon. Therefore we adapt (5.17) to this specific case, providing a general formula for $P^\alpha(\mathbf{w})$ (which we then use to evaluate $P^{\alpha'}_{\mathrm{d}}(\mathbf{w})$):

$$P^\alpha(\mathbf{w}) = \int p^0(y) \prod_{(s,i) \in \alpha} (q^s_{\mathrm{jump}} L^s(w^s_i \mid y) + p^s_{\mathrm{jump}} L^s(w^s_i - \pi \mid y)) dy$$
$$= \sum_{\alpha' \subseteq \alpha} \prod_{(s,i) \in \alpha'} p^s_{\mathrm{jump}} \prod_{(s,i) \in \alpha \setminus \alpha'} q^s_{\mathrm{jump}} P^\alpha_M(\mathbf{w}^{\alpha'}). \tag{6.3}$$

Here α' represents a hypothesis about which measurements in α have experienced a jump by π, and $\mathbf{w}^{\alpha'}$ is the same as $\mathbf{w}$, but with π subtracted from each w^s_i for which $(s,i) \in \alpha'$. Finally, we must compute the $P^\alpha_M(\mathbf{w})$ in (6.3). This is given by (6.2).

$$P^\alpha_M(\mathbf{w}) = I_0\left(\left|\sum_{(s,i) \in \alpha} (\sigma^s)^{-2} \exp(i w^s_i)\right|\right) \Big/ \prod_{(s,i) \in \alpha} 2\pi I_0((\sigma^s)^{-2}). \tag{6.4}$$

Now, we may begin the numerical computation of the original $P^\alpha(\mathbf{w} \mid t)$ (including all phenomena) for $\alpha = (3,3,2) = \{(1,3),(2,3),(3,2)\}$.

$$P^\alpha_M(\mathbf{w} \mid \blacksquare) = \frac{I_0(|0.1^{-2} e^{6.2067 i} + 0.07^{-2} e^{6.2011 i} + 0.05^{-2} e^{3.0885 i}|)}{(2\pi)^3 I_0(0.1^{-2}) I_0(0.07^{-2}) I_0(0.05^{-2})}$$
$$= 8.87511 \times 10^{-264}.$$

Naturally, this value is tiny because $w^3_2 = 3.0885$ is many sigmas away from w^1_3 and w^2_3. However, when we evaluate (6.3), we sum over cases where π is subtracted from w^s_i for some subset α' of α. This will produce a large value for the subset $\alpha' = \{(3,2)\}$:

$$P^\alpha_M(\mathbf{w}^{\alpha'} \mid \blacksquare)$$
$$= \frac{I_0(|0.1^{-2} e^{6.2067 i} + 0.07^{-2} e^{6.2011 i} + 0.05^{-2} e^{(3.0885 - \pi) i}|)}{(2\pi)^3 I_0(0.1^{-2}) I_0(0.07^{-2}) I_0(0.05^{-2})}$$
$$= 2.55152.$$

Naturally, the same value is produced for $\alpha' = \{(1,3),(2,3)\}$, but all other $\alpha' \subseteq \alpha$ produce negligible values. Therefore (6.3) may be evaluated as

$$P^\alpha_{\mathrm{d}}(\mathbf{w} \mid \blacksquare) = 0.99 \times 0.98 \times 0.06 \times 2.55152$$
$$+ 0.01 \times 0.02 \times 0.94 \times 2.55152 + (\text{tiny})$$
$$= 0.149009.$$

Here we use the subscript d in anticipation of the next step: incorporating the effects of missing features via (5.34). This will require the value of $P^\alpha_{\mathrm{d}}(\mathbf{w} \mid \blacksquare)$ for one-element sets α—this value is always equal to the prior value $1/(2\pi)$ for this specific feature. Referring to Table III for the values of P_{E}, P^s_{d} and P^s_{fa}, we obtain the following probability density which accounts for the possibility of the object lacking the feature (and all three measurements in α arising as false alarms):

$$P^\alpha_\mu(\mathbf{w} \mid \blacksquare) = 0.9 \times 0.94 \times 0.99 \times 0.6 \times 0.149009$$
$$+ 0.1 \times 0.02 \times 0.05 \times 0.001 \times (2\pi)^{-3}$$
$$= 0.0748806.$$

Here we use the subscript μ in anticipation of the final step: incorporating noise. For this we need values of $P_\mu^{\alpha'}$ for all subsets $\alpha' \subseteq \alpha$. Repeating the above steps for each subset yields

$$P_\mu^{(3,3,2)}(\mathbf{w} \mid \blacksquare) = 0.0748806, \qquad P_\mu^{(\emptyset,\emptyset,\emptyset)}(\mathbf{w} \mid \blacksquare) = 1,$$
$$P_\mu^{(\emptyset,3,2)}(\mathbf{w} \mid \blacksquare) = 0.0289101, \qquad P_\mu^{(3,\emptyset,\emptyset)}(\mathbf{w} \mid \blacksquare) = 0.134963,$$
$$P_\mu^{(3,\emptyset,2)}(\mathbf{w} \mid \blacksquare) = 0.0193784, \qquad P_\mu^{(\emptyset,3,\emptyset)}(\mathbf{w} \mid \blacksquare) = 0.142603,$$
$$P_\mu^{(3,3,\emptyset)}(\mathbf{w} \mid \blacksquare) = 0.421703, \qquad P_\mu^{(\emptyset,\emptyset,2)}(\mathbf{w} \mid \blacksquare) = 0.0859596.$$

Therefore

$$\begin{aligned} P^\alpha(\mathbf{w} \mid \blacksquare) &= 0.99 \times 0.8 \times 0.99 \times 0.0748806 \\ &\quad + 0.8 \times 0.99 \times 0.01 \times 0.0289101 \times 0.134963 + \cdots \\ &\quad + 0.01 \times 0.2 \times 0.01 \times 1 \times 0.134963 \times 0.142603 \\ &\quad \times 0.0859596 = 0.0595787. \end{aligned}$$

Carrying out the above computation for each object type t, we get

$$P^\alpha(\mathbf{w} \mid \blacksquare) = 0.0595787, \qquad P^\alpha(\mathbf{w} \mid \blacktriangle) = 0.0221195,$$
$$P^\alpha(\mathbf{w} \mid \blacklozenge) = 0.00999947, \qquad P^\alpha(\mathbf{w} \mid \blacktriangledown) = 2.06241.$$

These are the quantities needed in (5.15) to compute $P^\alpha(\mathbf{c},\mathbf{w})$ for $\alpha = (3,3,2)$. Using the parameters in Tables I and II with the called types $c_3^1 = \bullet$, $c_3^2 = \blacksquare$, and $c_2^3 = \blacktriangleright$ we have

$$\begin{aligned} P^\alpha(\mathbf{c},\mathbf{w}) &= 0.1 \times 0.9 \times 0.8 \times 0.4 \times 0.2 \times 0.8 \times 0.03 \\ &\quad \times 0.0595787 + 0.15 \times 0.8 \times 0.99 \times 0.8 \times 0.25 \\ &\quad \times 0.08 \times 0.03 \times 0.00999947 + 0.2 \times 0.95 \times 0.7 \\ &\quad \times 0.9 \times 0.2 \times 0.02 \times 0.97 \times 0.0221195 + 0.55 \\ &\quad \times 0.3 \times 0.9 \times 0.99 \times 0.3 \times 0.01 \times 0.95 \times 2.06241 \\ &= 0.000883213. \end{aligned}$$

This value is the largest among three-measurement sets α, and the eighth-largest overall. The three largest values are $P^{(\emptyset,\emptyset,2)}(\mathbf{c},\mathbf{w}) = 0.00598235$, $P^{(3,\emptyset,2)}(\mathbf{c},\mathbf{w}) = 0.00281482$, and $P^{(3,3,\emptyset)}(\mathbf{c},\mathbf{w}) = 0.00241721$. After computing $P^\alpha(\mathbf{c},\mathbf{w})$ for all 47 sets α, we may use (3.29) to get $R^{J\alpha}(\mathbf{z}^J)$ for each α, and then multiply them in (3.28) to get $G^J(\mathbf{z}^J,[a])$. For $[a] = \{(1,1,1),(2,2,\emptyset),(3,3,2)\}$ this yields

$$\begin{aligned} G^J(\mathbf{z}^J,[a]) &= 107662. \times 4673.65 \times 548368. \\ &= 2.75925 \times 10^{14}. \end{aligned}$$

Therefore (3.25) reduces to

$$\begin{aligned} \Pr([a] \mid \mathbf{z}) &= 0.000392981 \times 1.58147 \times 10^{78} C \\ &\quad \times 2.75925 \times 10^{14} \Pr([a_0] \mid \mathbf{z}) \\ &= 1.71484 \times 10^{89} C_0, \end{aligned}$$

where $C_0 = C \Pr([a_0] \mid \mathbf{z})$.

TABLE IV
The Five Most Probable Associations

Association	Probability
$\{(1,1,\emptyset),(2,2,1),(3,3,2)\}$	0.452373
$\{(1,2,\emptyset),(2,1,1),(3,3,2)\}$	0.226248
$\{(1,1,1),(2,2,\emptyset),(3,3,2)\}$	0.137136
$\{(\emptyset,2,\emptyset),(1,\emptyset,\emptyset),(2,1,1),(3,3,2)\}$	0.0765394
$\{(\emptyset,1,1),(1,\emptyset,\emptyset),(2,2,\emptyset),(3,3,2)\}$	0.0375843

After performing the above computation for all associations, we may determine the value of the normalization coefficient C_0 to find that $\Pr([a] \mid \mathbf{z}) = 0.137136$, which makes it the third most probable association of the 778. The most probable associations are shown in Table IV.

6.2. Simulation

The example above indicates that calculating association probability can be rather involved when one is incorporating many phenomena. The presence of bias in the kinematic component of the data introduces a term $\kappa([a])$ in the kinematic cost that depends on the association $[a]$ as a whole, preventing integer programming algorithms from finding optimal associations easily. The costs for the non-kinematic component of the data may be decomposed into the costs of individual sets $\alpha \in [a]$. In this case, however, there is a different kind of complication. The measurement likelihood function for the feature must be designed for specific feature of interest. In the example above, the feature exhibits a number of complicating phenomena (dependence on object type, missing values, noise, and the possibility of jumping by 180°). There is a valid concern that errors may exist in the derivation, transcription, and/or coding of such complicated formulas.

However, the probabilities produced, such as those in Table IV, have a precise meaning and are calculated exactly. As such, they enable a powerful check. For example, if we were able to collect independent events with probabilities of exactly 23%, they must obey precisely the same statistical law as flipping a coin whose probability of heads is 23%. To test the probabilities in the example of Section 6.1, we make 100,000 runs in which the data is generated according to the procedure specified by the given parameters, then compute the association probabilities and note, in each run, which association was correct. To do this, we require a method for generating error covariance matrices V_i^s. This is done as follows. First, a matrix is drawn from the Wishart distribution $\mathcal{W}(V_{\text{gen}}^s/m_{\text{gen}}^s, m_{\text{gen}}^s)$, then the result is scaled by a factor e^ξ where $\xi \sim \mathcal{N}(0, \log \gamma_{\text{gen}}^s)$. This process is carried out independently for each V_i^s. The parameters used in this procedure are

- Wishart parameters for error covariances: $m_{\text{gen}}^1 = 10$, $m_{\text{gen}}^2 = 10$, $m_{\text{gen}}^3 = 10$

- Baseline error covariances:

$$V^1_{\text{gen}} = \begin{pmatrix} 5 & 1 \\ 1 & 7 \end{pmatrix},$$

$$V^2_{\text{gen}} = \begin{pmatrix} 6 & -3 \\ -3 & 4 \end{pmatrix},$$

$$V^3_{\text{gen}} = \begin{pmatrix} 7 & 5 \\ 5 & 4 \end{pmatrix}$$

- Scale factors for error covariances: $\gamma^1_{\text{gen}} = 2$, $\gamma^2_{\text{gen}} = 3$, $\gamma^3_{\text{gen}} = 4$

There are some subtleties to the simulation process due to the diffuse nature of the priors employed, and to the requirement of not leaving extraneous signatures in the data. There are three places where diffuse priors occur: in the number of objects n; in ξ, the center of the region of objects; and in the bias β^s of some sensors. Appendix C shows how to simulate data efficiently with a specified number of tracks n^s on each sensor, even when the prior ρ^0 is diffuse. To circumvent directly sampling from diffuse priors for ξ and β^s, we impose the convention that scenes from sensors with diffuse bias priors are pre-processed to have a centroid of zero, and that all the other scenes are pre-processed jointly to have a centroid of zero. This renders the choices of ξ and of β^s for those sensors s with diffuse priors irrelevant because they leave no trace in the data.

One hundred thousand runs were made for this scenario, each run producing probabilities for each of the 778 possible associations. These associations were grouped into bins: those with probability less than 0.0001, those with probability between 0.0001 and 0.001, etc. For example, there were 452,911 associations with probability between 0.001 and 0.01. Of these, 1453 were correct associations, i.e., 0.32081%. This is certainly consistent—it lies between 0.1% and 1%—but we will demonstrate a far more stringent validation.

One may consider the 77,800,000 association events as independent flips of biased coins with various probabilities p_i of coming up heads (meaning "correct association"). (Technically the events are not completely independent but only very nearly so: two events picked at random have a 1-in-a-100,000 chance of coming from the same run, and hence exhibiting dependence.) We may consider either the entire data set of flips, or any subsets we like provided they are chosen without reference to ground truth. The law of large numbers says that the number of heads in any such data set will asymptotically approach a Gaussian distribution whose mean equals the sum of p_i over the events i, and whose variance equals the sum of $p_i(1 - p_i)$. Therefore, letting "tot" denote the total number of associations in a data set and "#" denote the number of these which are correct, we have

$$\begin{aligned} \mathbb{E}(\#/\text{tot}) &= \frac{1}{\text{tot}} \sum_{i=1}^{\text{tot}} p_i \qquad \text{and} \\ \sigma(\#/\text{tot}) &= \frac{1}{\text{tot}} \sqrt{\sum_{i=1}^{\text{tot}} p_i(1 - p_i)}. \end{aligned} \tag{6.5}$$

We may now make a much more incisive observation about 0.32081% than the fact that it lies between 0.1% and 1%. We find that $\mathbb{E}(\#/\text{tot}) = 0.32728\%$, and $\sigma(\#/\text{tot}) = 0.00848\%$. Thus, not only is 0.32081% between 0.1% and 1%, and not only is it close to the expected value 0.32728%, but it is precisely 0.76 standard deviations below the expected value. Furthermore, the large number of trials ensures that the distribution is approximately Gaussian, so we can convert the score $\zeta = -0.76$ into the statement that the number of correct associations lies at the 22nd percentile: i.e., it is a perfectly typical event.

Table V gives similar results for all probability bins. The first two columns specify the range of computed association probabilities in the bin. The third column gives the total number of associations (out of 77.8 million) with computed probabilities in the range of the bin. Column four gives the number of associations in the probability bin which are correct (and therefore sums to 100,000), and column five gives the fraction which are correct (column four divided by column three). The sixth and seventh columns give the expected value and standard deviation of the value in column five based on (6.5). Column eight gives the number of standard deviations which column five is above or below its expected value (i.e., $\zeta = (\#/\text{tot} - \mathbb{E}(\#/\text{tot}))/\sigma(\#/\text{tot})$). Finally, column nine converts column eight into the corresponding percentile of a Gaussian distribution.

The results in Table V verify the association probability formulas given in this paper in the scenario of Section 6.1. The most anomalous result is that only 99.42% of the associations with probability in the range 99% to 99.9% are correct, compared to an expected value of 99.53%, but this is only a 2.2 sigma event: still quite typical. Similar results hold for events other than complete associations. An event of great interest, for example, is whether the association deemed most probable is correct. As shown in Table VI, this occurs 82.8% of the time.

It is notable that both small and large probabilities are reliable—there is no failure in any probability range. Similar results are presented in [10] for a simpler, but denser, scenario (two sensors, kinematic-only data). There, a comparison is made between the exact bias integration formula presented in Section 4 and a method, currently considered state-of-the-art, in which the single most likely bias is selected and removed for each association hypothesis. Although the latter method performed reasonably accurately (17.6%, compared to 23.1% for the exact method) the probabilities it produced were off

TABLE V
Truth Versus Correctly Computed Probabilities for Various Probability Ranges

$p_{\min}$	$p_{\max}$	tot	#	#/tot	$\mathbb{E}$(#/tot)	σ(#/tot)	ζ	%
0	0.0001	75,986,700	85	0.0000011	0.0000010	0.0000001	0.65	0.74
0.0001	0.001	1,043,478	350	0.0003354	0.0003407	0.0000181	−0.29	0.39
0.001	0.01	452,911	1,453	0.0032081	0.0032728	0.0000848	−0.76	0.22
0.01	0.02	70,545	991	0.0140478	0.0141537	0.0004446	−0.24	0.41
0.02	0.05	64,167	2,020	0.0314804	0.0318053	0.0006919	−0.47	0.32
0.05	0.1	34,618	2,383	0.068837	0.0708085	0.0013765	−1.43	0.08
0.1	0.2	27,279	3,907	0.143224	0.143045	0.002113	0.08	0.53
0.2	0.3	14,298	3,579	0.250315	0.245706	0.003592	1.28	0.90
0.3	0.4	9,497	3,249	0.342108	0.347989	0.004879	−1.21	0.11
0.4	0.5	7,774	3,522	0.453049	0.448765	0.005631	0.76	0.78
0.5	0.6	6,804	3,753	0.551587	0.549095	0.006022	0.41	0.66
0.6	0.7	6,557	4,234	0.645722	0.650153	0.005879	−0.75	0.23
0.7	0.8	7,618	5,747	0.754397	0.751855	0.004937	0.51	0.70
0.8	0.9	11,162	9,602	0.860240	0.855228	0.003319	1.51	0.93
0.9	0.99	35,018	33,669	0.961477	0.960444	0.001033	1.00	0.84
0.99	0.999	20,013	19,897	0.994204	0.995258	0.000485	−2.17	0.01
0.999	1	1,561	1,559	0.998719	0.999301	0.000669	−0.87	0.19

TABLE VI
MAP Performance for Correctly Computed Probabilities

tot	#	#/tot	$\mathbb{E}(\#/tot)$	$\sigma(\#/tot)$	ζ	%
100000	82805	0.828050	0.826473	0.000996	1.58	0.94

by up to several hundred standard deviations, and were particularly inaccurate for events at the high- and low-probability extremes.

7. CONCLUSION

This paper espouses a Bayesian methodology for developing association probability formulas to use as a basis for developing high-quality, robust association algorithms. Section 2 derives a general formula for association probability given data from multiple sensors with uncorrelated errors at some fixed time. Sections 4 and 5 provide specific formulas for biased kinematic data and various kinds of non-kinematic data, respectively. A hallmark of the Bayesian approach is that the probabilities produced can be rigorously verified by the method introduced in Section 6.2.

One may argue that although the probabilities produced by a Bayesian method may be quite accurate in the scenario for which they are designed, real data will surely fail to conform to the underlying model and input parameters, so the assumption that Bayesian algorithms are superior to simpler non-Bayesian ones is unjustified for real data. This is not actually an argument for or against Bayesian algorithms, however—it merely makes the valid point that the true measure of the quality of an algorithm or a formula is its performance on real data. An ideal testbed would subject algorithms to the same range of phenomena that domain experts believe to occur in real data, presenting such a variety of scenarios that over-fitting is impossible. If an algorithm is able to produce association probabilities that are rigorously verifiable (in the sense of Section 6.2) within such a testbed, then when it outputs a result such as "the probability that these two measurements arose from the same object is 75%," this has a clear and useful meaning: of all possible states of the world which could give rise to the data we have observed, the measurements arise from the same object in three out of four of them.

To achieve such a reliable algorithm it is necessary to develop an accurate simulation capability, then to develop an association probability formula which exploits the domain expertise captured in the simulation to achieve accurate probability calculations. Only though a methodology for developing formulas which produces verifiably meaningful probabilities in simpler cases can one hope to achieve the goal of producing meaningful probabilities in the more realistic and complex cases. This paper supplies a general approach for developing such formulas, as well as results for certain specific cases.

ACKNOWLEDGMENTS

The author thanks Larry Stone for critiquing earlier, less elegant versions of Section 2, Roy Streit for suggesting the proper method for evaluating (D.4), and the reviewers for their insightful recommendations.

APPENDIX A. INCORPORATING FALSE ALARMS

The derivation in Section 2 assumes that none of the measurements are false alarms. The incorporation of false alarms affects only the combinatorial factor $g([a])$ in (2.12), however. Therefore the simpler, non-false-alarm case is used in the body of this paper. All results may be extended to include false alarms by using the modified value of $g([a])$ presented in this appendix.

When false alarms are present, the size n_{D}^s of J_{D}^s (the set of objects detected on sensor s) may fall short

of n^s (the number of measurements on sensor s). Let I^s_{FA} denote the set of false alarm measurements on sensor s and $n^s_{\mathrm{FA}} = |I^s_{\mathrm{FA}}|$, so that $n^s_{\mathrm{D}} + n^s_{\mathrm{FA}} = n^s$. The prior probability that the number of false alarms on sensor s is exactly n^s_{FA} is denoted $\rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}})$: this is assumed to be a known property of sensor s. If a measurement arises as a false alarm, the distribution of its (spurious) value z^s_i given the systematic error β^s and center ξ is denoted $L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi)$. The analog of Lemma 2.1 in the false-alarm case is the following.

LEMMA A.1 *The probability density of the measurement array* $\mathbf{z}^s$ *arising according to the mapping* a^s *given the object state array* $\mathbf{x}$*, the systematic error* β^s*, and the center* ξ *of the prior region of state space is*

$$\begin{aligned}\Pr(\mathbf{z}^s, a^s \mid \mathbf{x}, \beta^s, \xi) &= n^s_{\mathrm{FA}}! \rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}}) \prod_{i \in I^s_{\mathrm{FA}}} L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi) \\ &\quad \times \frac{1}{n^s!} \prod_{j \in J^s_{\mathrm{D}}} P^s_{\mathrm{D}}(x_j) L^s(z^s_{a^s(j)} \mid x_j, \beta^s) \prod_{j \notin J^s_{\mathrm{D}}} Q^s_{\mathrm{D}}(x_j).\end{aligned} \tag{A.1}$$

PROOF Given the object state array $\mathbf{x}$, the probability $\Pr(n^s, J^s_{\mathrm{D}} \mid \mathbf{x})$ that the subset of detected objects is J^s_{D} and the total number of measurements n^s reflects an additional $n^s_{\mathrm{FA}} = n^s - n^s_{\mathrm{D}}$ false alarms is $\rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}})$ times the product of $P^s_{\mathrm{D}}(x_j)$ and $Q^s_{\mathrm{D}}(x_j)$ over detected and undetected objects j, respectively. Given J^s_{D} and n^s, there are $n^s!/n^s_{\mathrm{FA}}!$ equally likely mappings a^s, so $\Pr(a^s \mid \mathbf{x}) = \Pr(n^s, J^s_{\mathrm{D}} \mid \mathbf{x}) n^s_{\mathrm{FA}}!/n^s!$. The probability density of the measurement array $\mathbf{z}^s$ given a^s, $\mathbf{x}$, β^s, and ξ is the product of the individual likelihood functions $L^s(z^s_i \mid x_j, \beta^s)$ for the detections $i = a^s(j)$ times the product of $L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi)$ for the false alarms. Equation (A.1) now follows from $\Pr(\mathbf{z}^s, a^s \mid \mathbf{x}, \beta^s, \xi) = \Pr(\mathbf{z}^s \mid a^s, \mathbf{x}, \beta^s, \xi) \Pr(a^s \mid \mathbf{x})$.

Equation (A.1) is identical to (2.1) except that it has a dependence on ξ and an additional false-alarm–related pre-factor. This pre-factor remains intact throughout the derivation of association probability in Section 2, until we arrive at this analog of (2.8):

$$\begin{aligned}\Pr(\mathbf{z}, [a]' \mid \boldsymbol{\beta}, \xi) &= n^s_{\mathrm{FA}}! \rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}}) \prod_{i \in I^s_{\mathrm{FA}}} L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi) \\ &\quad \times \frac{\gamma^0(n_{\mathrm{D}})}{\prod_{s \in S} n^s!} \prod_{\alpha \in [a]'} P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi),\end{aligned} \tag{A.2}$$

where the notation $[a]'$ denotes an incomplete association—one for which not all measurements are included in some cluster $\alpha \in [a]'$ (the remaining measurements being false alarms). We may simplify (A.2) by prescribing a natural form for the false alarm likelihood function $L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi)$. To do this, we first define the probability p^s that an object is detected on sensor s and on no others (without regard to what value it produced there):

$$p^s = \int p^0(x \mid \xi) P^s_{\mathrm{D}}(x) \prod_{s' \neq s} Q^{s'}_{\mathrm{D}}(x) dx. \tag{A.3}$$

(Like q, p^s is independent of ξ when detection probability is independent of kinematic state.) We write $L^s_{\mathrm{D}}(z^s_i \mid \beta^s, \xi) = P^s(z^s_i \mid \beta^s, \xi)/p^s$ to denote the probability density of an object producing the value z^s_i on sensor s, given that it was detected by sensor s and no others, where $P^s(z^s_i \mid \beta^s, \xi)$ is given by (2.5) for the special case $\alpha = \{(s,i)\}$.

The likelihood ratio

$$\Lambda^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi) = \frac{L^s_{\mathrm{FA}}(z^s_i \mid \beta^s, \xi)}{L^s_{\mathrm{D}}(z^s_i \mid \beta^s, \xi)} \tag{A.4}$$

gives the relative likelihood of the value z^s_i to arise from a false alarm versus a single-sensor object detection. Because the integrals of both its numerator and denominator (over z^s_i) are 1, if $\Lambda^s_{\mathrm{FA}} > 1$ for some value of z^s_i (meaning z^s_i is more likely to be a false alarm), then it must be less than 1 somewhere else. If one has a very clear notion of the characteristics of measurements that have arisen due to false alarms, it is appropriate to model L^s_{FA} directly. Otherwise, it is prudent to model Λ^s_{FA} instead, perhaps assigning it values larger than 1 for wild values of z^s_i, and somewhat less than 1 for more credible values. The simplest, most robust model for Λ^s_{FA} is that it is identically 1. We call this the *non-informative noise* assumption (cf. Section 5.2), because it stipulates that we have no *a priori* knowledge of the false alarm behavior that allows us to distinguish between false alarms and detections, even when we know the true values of the systematic errors β^s and the center ξ of the set of objects. The non-informative noise assumption has the additional benefit of simplifying the association equations.

If we make the non-informative noise assumption, then the effect of false alarms can be encapsulated in a factor $\gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}})$, defined by

$$\gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}}) = n^s_{\mathrm{FA}}! \rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}}) (p^s)^{-n^s_{\mathrm{FA}}}. \tag{A.5}$$

Given this definition, Equation (A.2) may be expressed more simply.

LEMMA A.2 *Using the non-informative noise assumption for false alarms, the probability density of the measurement arrays* $\mathbf{z}$ *arising according to the incomplete association* $[a]'$ *given the systematic error* β^s*, and the center* ξ *of the prior region of state space is*

$$\Pr(\mathbf{z}, [a]' \mid \boldsymbol{\beta}, \xi) = \prod_{s \in S} \gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}}) \frac{\gamma^0(n_{\mathrm{D}})}{\prod_{s \in S} n^s!} \prod_{\alpha \in [a]} P^\alpha(\mathbf{z} \mid \boldsymbol{\beta}, \xi), \tag{A.6}$$

where $[a]$ *represents the completion of the incomplete association* $[a]'$*: i.e.,* $[a]'$ *with clusters* $\{(s,i)\}$ *adjoined for each false alarm measurement.*

PROOF Follows directly from (A.2)–(A.5).

The incomplete association $[a]'$ distinguishes between objects detected on only one sensor and false alarms. Its completion $[a]$ does not, but this can be quite useful in compressing the hypothesis space when one cares only about which measurements to associate. To obtain the probability of a completion $[a]$, we sum (A.6) over all $[a]'$ for which $[a]$ is the completion. This sum acts only on the combinatorial factors $\gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}})$ and $\gamma^0(n_{\mathrm{D}})$. To express this sum we need additional notation. Let n^+_{D} be the number of sets $\alpha \in [a]$ with at least two elements, and n^s_1 be the number of measurements on sensor s which occur by themselves in $[a]$. We may write n_{D} in terms of n^+_{D}, by noting that the number of measurements on sensor s which arise from objects detected once only is $n^s_1 - n^s_{\mathrm{FA}}$, so

$$n_{\mathrm{D}} = |[a]'| = n^+_{\mathrm{D}} + \sum_{s\in S}(n^s_1 - n^s_{\mathrm{FA}}). \tag{A.7}$$

Summing (A.6) over all possible associations $[a]'$ yielding the same $[a]$ amounts to summing over all possible number of false alarms on each sensor (i.e., $n^s_{\mathrm{FA}} = 0$ to n^s_1) and accounting for all ways to choose the n^s_{FA} false alarms from n^s_1 measurements. Thus we introduce the quantity

$$\gamma(n^+_{\mathrm{D}}, \mathbf{n}_1) = \sum_{n^1_{\mathrm{FA}}=0}^{n^1_1} \cdots \sum_{n^r_{\mathrm{FA}}=0}^{n^r_1} \gamma^0\left(n^+_{\mathrm{D}} + \sum_{s\in S}(n^s_1 - n^s_{\mathrm{FA}})\right) \times \prod_{s\in S}\left(\binom{n^s_1}{n^s_{\mathrm{FA}}}\gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}})\right), \tag{A.8}$$

where $\mathbf{n}_1 = (n^1_1, n^2_1, \ldots, n^r_1)$. The probability density $\Pr(\mathbf{z}, [a] \mid \beta, \xi)$ may now be given by a formula identical to (2.8), but with $\gamma^0(n_{\mathrm{D}})$ replaced by $\gamma(n^+_{\mathrm{D}}, \mathbf{n}_1)$. Thus we arrive at this analog of Theorem 2.4:

THEOREM A.3 *When false alarms may occur, the probability of the association* $[a]$ *given the measurements* $\mathbf{z}$ *is*

$$\Pr([a] \mid \mathbf{z}) = g([a])G(\mathbf{z}, [a])\Pr([a_0] \mid \mathbf{z}), \tag{A.9}$$

where

$$g([a]) = \frac{\gamma(n^+_{\mathrm{D}}, \mathbf{n}_1)}{\gamma(0, \mathbf{n})}, \tag{A.10}$$

and $\mathbf{n}$ *denotes* $(n^1, n^2, \ldots, n^r)$.

PROOF This follows from (A.6), and the definition (A.8).

The significance of Theorem A.3 is that the effect of false alarms is entirely encapsulated in the combinatorial factor $g([a])$, which is independent of the measurement data $\mathbf{z}$. Although (A.10) is considerably more complicated than (2.11), it can be evaluated explicitly and stored, if desired, for any priors ρ^0 and ρ^s_{FA}. More typically, one simply assumes these priors to be Poisson: Section B.3 gives a simple formula for $g([a])$ in this case.

APPENDIX B. PRIOR ON THE NUMBER OF OBJECTS

The discussion preceding Equation (2.4) addresses the violation of Bayesian methodology one makes when estimating prior information from the data. This violation occurs when one makes the popular assumption that the prior distribution of the number of objects $\rho^0(n)$ is Poisson with the mean number of objects estimated from the data. This issue is addressed here. Section B.1 computes the combinatorial factor $g([a])$ for several choices of prior $\rho^0(n)$. Section B.2 demonstrates why the Poisson prior can work well in practice despite its questionable validity. Finally, Section B.3 computes $g([a])$ explicitly for the false alarm case, assuming Poisson priors.

B.1. The Combinatorial Factor

Poisson prior

For the Poisson prior

$$\rho^0(n) = e^{-\nu}\frac{\nu^n}{n!}, \tag{B.1}$$

(2.9) simplifies to

$$\gamma^0(n_D) = e^{\nu(q-1)}\nu^{n_D}, \tag{B.2}$$

and therefore (2.11) reduces to

$$g([a]) = \nu^{n_D - n_T}. \tag{B.3}$$

For this prior, the association probability in the absence of systematic errors (3.2) takes the following, purely multiplicative form

$$\Pr([a] \mid \mathbf{z}) = \Pr([a_0] \mid \mathbf{z}) \prod_{\alpha\in[a]^+} \tilde{R}^\alpha(\mathbf{z}), \tag{B.4}$$

where

$$\tilde{R}^\alpha(\mathbf{z}) = \frac{\nu P^\alpha(\mathbf{z})}{\prod_{(s,i)\in\alpha}\nu P^s(z^s_i)}. \tag{B.5}$$

This form is required for the traditional, efficient solution of the two-sensor association problem, because it allows the cost of an association to be represented as a sum of costs over each $\alpha \in [a]^+$. In the two-sensor case, one can organize these costs into a cost matrix, and find the association with Maximal A Posteriori Probability (the MAP association) [7] using efficient integer programming techniques such as the JVC algorithm [13]. The Poisson prior is the only prior on n which yields this purely multiplicative form [17, 18, 19], which makes it quite convenient to use.

To use the Poisson prior, however, requires that one estimate ν from the data. For constant detection probabilities P^s_{D}, if one considers the probability of the number of tracks on each sensor given ν, then one gets the following Maximum Likelihood Estimator of ν:

$$\hat{\nu} = \frac{n_T}{\sum_{s\in S} P^s_D}. \tag{B.6}$$

This, then, is a reasonable value of ν to use with the Poisson prior in practice.

A useful class of priors

For any nonnegative integer k and any $\lambda \in [0,1)$, the following prior is well defined for $n \in \mathbb{Z}$:

$$\rho^0(n) = CI[n \geq k]\frac{(n-k)!}{n!}\lambda^n, \tag{B.7}$$

where the indicator function $I[n \geq k]$ is 1 when $n \geq k$ and 0 otherwise, and C is a normalizing constant. For this prior, (2.11) reduces to

$$g([a]) = \frac{(n_D-k)!}{(n_T-k)!}(1-q)^{n_T-n_D}, \tag{B.8}$$

in the limit $\lambda \to 1^-$, provided $n_D \geq k$. This prior has a useful interpretation in the cases $k = 0$ and $k = 1$.

Diffuse prior

The diffuse prior is obtained when $k = 0$. This prior weights all possible number of objects equally, and one might think this would be a reasonable, canonical choice for a prior in the absence of information. In this case, (B.8) is

$$g([a]) = \frac{n_D!}{n_T!}(1-q)^{n_T-n_D}. \tag{B.9}$$

This diffuse prior does not have the same claim to impartiality as the diffuse spatial prior, however. The spatial case has a compelling invariance with respect to Euclidean transformations which has no analog here.

Log-diffuse prior

If one had to guess an integer in the absence of any information, the guess 5 would be better than 74,936 because the latter is so specific. Rather than assuming a uniform distribution for n, we assume a uniform distribution for $\log n$, in accordance with Benford's law. This yields the $k = 1$ case of (B.8):

$$g([a]) = \frac{(n_D-1)!}{(n_T-1)!}(1-q)^{n_T-n_D}. \tag{B.10}$$

This is the form of $g([a])$ recommended for general use when the computational issues favoring the Poisson prior are not important.

B.2. Justification of the Poisson prior

If we use the Poisson prior, then $\log(g([a]))$ varies linearly with n_D (aside from a constant offset): $\log(g([a])) = \log(\nu)(n_D - n_T)$. This is the behavior that produces the purely multiplicative form (B.4) for $\Pr([a] \mid \mathbf{z})$. An alternative, then, to using the Poisson prior is to linearize $\log(g([a])$ about some point n_D^* for a prior one believes is appropriate.

The slope of $\log(g([a])$ for the Poisson prior is constant,

$$\frac{d}{dn_D}\log(g([a]) = \log(\nu), \tag{B.11}$$

whereas this slope for the general prior (B.7) is

$$\begin{aligned}\frac{d}{dn_D}\log(g([a]) &= \psi(n_D+1-k) - \log(1-q)\\ &\approx \log\left(\frac{n_D+1/2-k}{1-q}\right),\end{aligned} \tag{B.12}$$

where ψ denotes the digamma function. Therefore the Poisson prior may, in fact, be regarded as a linearization of the general prior (B.7) about the point

$$n_D^* = (1-q)\nu + k - 1/2. \tag{B.13}$$

This is a quite reasonable place to linearize. If ν is a good estimate of the number of objects (such as the MLE value ν given by (B.6)), then $(1-q)\nu$ is a good estimate of the number of detected objects n_D. The offset by $-1/2$ or $1/2$ in the cases $k = 0$ or $k = 1$, respectively, is negligible.

Therefore, although using the Poisson prior with some estimate of ν based on the data violates Bayesian methodology, it results in a formula that is fairly accurate because it is a linearization of a justifiable diffuse or log-diffuse prior on the number of objects. Because it is an artificially tight distribution, however, it will underestimate the probabilities of associations with a very large or a very small numbers of detected objects n_D.

B.3. False Alarm Case

In contrast to $\rho^0(n)$, it is quite natural to assume the number of false alarms on a sensor is Poisson distributed. We let

$$\rho^s_{\mathrm{FA}}(n^s_{\mathrm{FA}}) = e^{-\nu^s_{\mathrm{FA}}}\frac{(\nu^s_{\mathrm{FA}})^{n^s_{\mathrm{FA}}}}{n^s_{\mathrm{FA}}!}. \tag{B.14}$$

For this prior, (A.5) simplifies to

$$\gamma^0_{\mathrm{FA}}(n^s_{\mathrm{FA}}) = e^{-\nu^s_{\mathrm{FA}}}\left(\frac{\nu^s_{\mathrm{FA}}}{p^s}\right)^{n^s_{\mathrm{FA}}}. \tag{B.15}$$

Even with this Poisson assumption, the value of $g([a])$ does not simplify much unless we also assume that $\rho^0(n)$ is Poisson distributed. If we accept (B.1) and (B.14), however, then the complicated quantity $\gamma(n_D^+, \mathbf{n}_1)$ in (A.8) simplifies to

$$\begin{aligned}\gamma(n_D^+, \mathbf{n}_1) = \exp&\left(-\nu(1-q) - \sum_{s\in S}\nu^s_{\mathrm{FA}}\right)\\ &\times \nu^{n_D^+}\prod_{s\in S}\left(\nu + \frac{\nu^s_{\mathrm{FA}}}{p^s}\right)^{n_1^s}.\end{aligned} \tag{B.16}$$

Using this, we may write (A.10) as

$$g([a]) = \nu^{n_D^+}\prod_{s\in S}\left(\nu + \frac{\nu^s_{\mathrm{FA}}}{p^s}\right)^{n_1^s - n^s}. \tag{B.17}$$

As in the case without false alarms, the Poisson assumption results in the multiplicative form (B.4) for association probability, but with (B.5) replaced by

$$\tilde{R}^{\alpha}(\mathbf{z}) = \frac{\nu P^{\alpha}(\mathbf{z})}{\prod_{(s,i)\in\alpha}\left(\nu + \frac{\nu^s_{\mathrm{FA}}}{p^s}\right)P^s(z^s_i)}. \qquad \text{(B.18)}$$

APPENDIX C. SIMULATING A SPECIFIED NUMBER OF MEASUREMENTS

It would seem that the first step in the simulation of an association scenario would be to sample the total number of objects n from the prior $\rho^0(n)$. In the case of the diffuse and log-diffuse priors described in Appendix B, this is problematic: these priors have infinite mean. This difficulty may be overcome by fixing $\mathbf{n}$, the number of tracks observed on each sensor. Indeed, it is more convenient to gather statistics from runs with a constant value of $\mathbf{n}$ anyway, regardless of whether the prior ρ^0 forces one's hand. One could imagine using a rejection method to sample in this way: i.e., one could sample n from ρ^0, then generate tracks on each sensor in the usual way, but reject any scenario that does not end up with $\mathbf{n}$ tracks on the sensors. This is highly inefficient however. This appendix describes an efficient procedure which is statistically equivalent to the inefficient rejection method.

We begin with the probability of an object being detected on some subset $\bar{\alpha}$ of S and not on any other sensors:

$$P^{\bar{\alpha}} = \int p^0(x \mid \xi)\prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}(x)\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}(x)dx. \qquad \text{(C.1)}$$

It is assumed that this result is independent of ξ. In particular, we are interested in two cases. First, when $P^s_{\mathrm{D}}(x)$ is independent of x we have

$$P^{\bar{\alpha}} = \prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}. \qquad \text{(C.2)}$$

On the other hand, if P^s_{D} exhibits as dependence on object type, as in Section 5.1.1 or 5.1.2, then

$$P^{\bar{\alpha}} = \sum_t p^0(t)\prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}(t)\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}(t), \qquad \text{(C.3)}$$

which generalizes the formula for q given in (5.10).

We define $[\bar{a}]$ to be the *multiset* of values of $\bar{\alpha}$ for the $\alpha \in [a]$. That is, if $\alpha = \{(1,5),(3,2)\}$ (i.e., according to $[a]$ some object produced measurement 5 on sensor 1, measurement 2 on sensor 3, and no others), then $\bar{\alpha} = \{1,3\}$. If there are exactly four $\alpha \in [a]$ for which $\bar{\alpha} = \{1,3\}$, then we write $m(\{1,3\}) = 4$. This multiplicity function m is part of the multiset $[\bar{a}]$. After some manipulation, we find that

$$\Pr([\bar{a}]) = \gamma^0(n_{\mathrm{D}})\prod_{\bar{\alpha}\in[\bar{a}]}\frac{(P^{\bar{\alpha}})^{m(\bar{\alpha})}}{m(\bar{\alpha})!}. \qquad \text{(C.4)}$$

Each $[\bar{a}]$ determines the number of tracks $\mathbf{n}$ on the sensors, so we define $I_{\mathbf{n}}([\bar{a}])$ to be 1 if $[\bar{a}]$ is consistent with $\mathbf{n}$ and 0 otherwise. The conditional probability $\Pr([\bar{a}] \mid \mathbf{n})$ is given by

$$\Pr([\bar{a}] \mid \mathbf{n}) = \frac{I_{\mathbf{n}}([\bar{a}])\Pr([\bar{a}])}{\sum_{[\bar{a}]'} I_{\mathbf{n}}([\bar{a}]')\Pr([\bar{a}]')}. \qquad \text{(C.5)}$$

This formula may be used to randomly generate $[\bar{a}]$ given $\mathbf{n}$. Then, given $[\bar{a}]$, one may choose a value of $[a]$ from the equally likely possibilities.

Having obtained a random association $[a]$, one must then sample a state for each $\alpha \in [a]$, bearing in mind the statistical influence of the fact that the object is detected on the sensors $\bar{\alpha}$ and no others. In the case (C.2) of object-independent detection probabilities P^s_{D}, there is no effect on the state: it may be sampled in the usual fashion. In the case (C.3), where P^s_{D} depends on the object type t, the object type must be sampled more carefully, however. This requires accounting for its dependence on $\bar{\alpha}$. To sample the object type t for an object detected on sensors $s \in \bar{\alpha}$ but no others, one must draw from $\Pr(t \mid \bar{\alpha})$ rather than $p^0(t)$, where

$$\begin{aligned}\Pr(t \mid \bar{\alpha}) &= \frac{p^0(t)\prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}(t)\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}(t)}{P^{\bar{\alpha}}} \\ &= \frac{p^0(t)\prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}(t)\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}(t)}{\sum_{t'} p^0(t')\prod_{s\in\bar{\alpha}} P^s_{\mathrm{D}}(t')\prod_{s\notin\bar{\alpha}} Q^s_{\mathrm{D}}(t')}.\end{aligned} \qquad \text{(C.6)}$$

When detection probability is a function of object type t, the distribution of object types on detected objects may differ greatly from $p^0(t)$. When formulating an association scenario, it is helpful to know how many objects of each type to expect. The expected value of $n_D(t)$, the number of objects of type t detected, for a given value of $\mathbf{n}$ is

$$\mathbb{E}[n_D(t) \mid \mathbf{n}] = \sum_{[\bar{a}]}\Pr([\bar{a}] \mid \mathbf{n})\sum_{\bar{\alpha}\in[\bar{a}]}\Pr(t \mid \bar{\alpha}). \qquad \text{(C.7)}$$

When P^s_{D} is independent of t, $\Pr(t \mid \bar{\alpha}) = p^0(t)$, so $\mathbb{E}[n_D(t) \mid \mathbf{n}]$ reduces to $\mathbb{E}[n_D \mid \mathbf{n}]p^0(t)$.

APPENDIX D. A CORRELATED GAUSSIAN PRODUCT FORMULA

A standard formula for the product of k Gaussians is

$$\prod_{i=1}^{k}\mathcal{N}(x;m_i,V_i) = \sqrt{|2\pi W|}\,\mathcal{N}(x;\mu,W)\prod_{i=1}^{k}\mathcal{N}(m_i;\mu,V_i), \qquad \text{(D.1)}$$

where

$$W = \left(\sum_{i=1}^{k} V_i^{-1}\right)^{-1} \qquad \text{and} \qquad \mu = W\sum_{i=1}^{k} V_i^{-1}m_i, \qquad \text{(D.2)}$$

and therefore

$$\int \prod_{i=1}^{k} \mathcal{N}(x; m_i, V_i) dx = \sqrt{|2\pi W|} \prod_{i=1}^{k} \mathcal{N}(m_i; \mu, V_i). \tag{D.3}$$

Theorem 4.2 requires the evaluation of a product of k Gaussians which is much more complicated than (D.3). Let

$$v = \int \int \cdots \int \prod_{i=1}^{k} \mathcal{N}\left(\sum_{j=1}^{p} A_{ij} x_j; m_i, V_i \right) dx_1 dx_2 \cdots dx_p, \tag{D.4}$$

where each $x_j \in \mathbb{R}^n$, each $m_i \in \mathbb{R}^m$, each V_i is a symmetric positive definite $m \times m$ matrix, and each A_{ij} is an $m \times n$ matrix. In evaluating (D.4) the product over i becomes a sum over i within the argument of the Gaussian, which leads to the requirement for computing the following critical quantities:

$$U_{jj'} = \sum_{i=1}^{k} A_{ij}^T V_i^{-1} A_{ij'} \qquad \text{for} \quad 1 \le j, j' \le p, \tag{D.5}$$

$$b_j = \sum_{i=1}^{k} A_{ij}^T V_i^{-1} m_i \qquad \text{for } 1 \le j \le p \qquad \text{and} \tag{D.6}$$

$$c = \sum_{i=1}^{k} m_i^T V_i^{-1} m_i. \tag{D.7}$$

These may be used to express the integrand in (D.4) as a quadratic form over $\mathbb{R}^{np}$. We define

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1p} \\ U_{21} & U_{22} & \cdots & U_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ U_{p1} & U_{p2} & \cdots & U_{pp} \end{pmatrix} \qquad \text{and} \tag{D.8}$$

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix},$$

so $\mathbf{x}$ and $\mathbf{b}$ are vectors in $\mathbb{R}^{np}$, and $\mathbf{U}$ is an $np \times np$ symmetric positive semi-definite matrix. For a broad class of matrices A_{ij}, $\mathbf{U}$ is invertible and we may therefore define $\boldsymbol{\mu} = \mathbf{U}^{-1}\mathbf{b}$. The integrand of (D.4) may now be written

$$\prod_{i=1}^{k} \mathcal{N}\left(\sum_{j=1}^{p} A_{ij} x_j; m_i, V_i \right) = v \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{U}^{-1}), \tag{D.9}$$

provided $|\mathbf{U}| > 0$, where

$$v = \left(\exp(c - \mathbf{b}^T \mathbf{U}^{-1} \mathbf{b}) (2\pi)^{mk-np} |\mathbf{U}| \prod_{i=1}^{k} |V_i| \right)^{-1/2}. \tag{D.10}$$

D.1 Gaussian Transformations

Theorem 4.3 requires certain formulas for manipulating Gaussians. First, for any invertible $np \times np$ matrix $\mathbf{C}$, a simple algebraic manipulation shows that

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{U}^{-1}) = \|\mathbf{C}\| \mathcal{N}(\mathbf{C}\mathbf{x}; \mathbf{C}\boldsymbol{\mu}, (\mathbf{C}^{-T} \mathbf{U} \mathbf{C}^{-1})^{-1}), \tag{D.11}$$

where $\|\mathbf{C}\|$ denotes the absolute value of the determinant of $\mathbf{C}$.

We now give a formula for integrating (D.9) over x_{t+1} through x_p only for any $0 \le t \le p$. We let $\mathbf{x}_F = (x_1, x_2, \ldots, x_t)$ denote the free variables, and $\mathbf{x}_I = (x_{t+1}, x_{t+2}, \ldots, x_p)$ denote the variables of integration. We may then decompose $\mathbf{U}$ and $\mathbf{b}$ into corresponding blocks:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_F \\ \mathbf{x}_I \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} \mathbf{U}_{FF} & \mathbf{U}_{FI} \\ \mathbf{U}_{IF} & \mathbf{U}_{II} \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} \mathbf{b}_F \\ \mathbf{b}_I \end{pmatrix}, \tag{D.12}$$

where $\mathbf{U}_{FI}^T = \mathbf{U}_{IF}$. We define the following function of $\mathbf{x}_F$,

$$\boldsymbol{\mu}_I' = \mathbf{U}_{II}^{-1}(\mathbf{b}_I - \mathbf{U}_{IF}\mathbf{x}_F), \tag{D.13}$$

as well as the Schur complement [12] of $\mathbf{U}_{II}$ in $\mathbf{U}$ (denoted $\mathbf{U}_{FF}^*$), and related quantities:

$$\mathbf{U}_{FF}^* = \mathbf{U}_{FF} - \mathbf{U}_{FI}\mathbf{U}_{II}^{-1}\mathbf{U}_{IF}, \tag{D.14}$$

$$\mathbf{b}_F^* = \mathbf{b}_F - \mathbf{U}_{FI}\mathbf{U}_{II}^{-1}\mathbf{b}_I \qquad \text{and} \tag{D.15}$$

$$\boldsymbol{\mu}_F^* = \mathbf{U}_{FF}^{*-1}\mathbf{b}_F^*. \tag{D.16}$$

The Gaussian in (D.9) may now be written as

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{U}^{-1}) = \mathcal{N}(\mathbf{x}_F; \boldsymbol{\mu}_F^*, \mathbf{U}_{FF}^{*-1}) \mathcal{N}(\mathbf{x}_I; \boldsymbol{\mu}_I', \mathbf{U}_{II}^{-1}), \tag{D.17}$$

whence

$$\int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{U}^{-1}) d\mathbf{x}_I = \mathcal{N}(\mathbf{x}_F; \boldsymbol{\mu}_F^*, \mathbf{U}_{FF}^{*-1}). \tag{D.18}$$

The decomposition (D.12) gives an alternative formula for v in which the following quantities in (D.10) are expressed in an alternative fashion:

$$\begin{aligned} \mathbf{b}^T \mathbf{U}^{-1} \mathbf{b} &= \mathbf{b}_I^T \mathbf{U}_{II}^{-1} \mathbf{b}_I + \mathbf{b}_F^{*T} \mathbf{U}_{FF}^{*-1} \mathbf{b}_F^* \qquad \text{and} \\ |\mathbf{U}| &= |\mathbf{U}_{II}| \, |\mathbf{U}_{FF}^*|. \end{aligned} \tag{D.19}$$

These matrix identities may also be obtained directly through the block matrix inversion formulas [12].

REFERENCES

[1] M. Abramowitz and I. A. Stegun
Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables.
Dover, New York, 1964.

[2] J. Areta, Y. Bar-Shalom and M. Levedhal
A hierarchical benchmark association problem in missile defense.
Signal and Data Processing of Small Targets, 2005, 382–393.

[3] Y. Bar-Shalom and H. Chen
Multisensor track-to-track association for tracks with dependent errors.
43rd IEEE Conference on Decision and Control, 2004.

[4] Y. Bar-Shalom and T. E. Fortmann
Tracking and Data Association.
Academic Press, San Diego, 1988.

[5] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Artech, Boston, 1999.

[6] C-Y. Chong and S. Mori
Metrics for feature-aided track association.
9th International Conference on Information Fusion, July 2006.

[7] C. Y. Chong, S. Mori and K-C. Chang
Distributed multitarget multisensor tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Artech, Boston, 1990.

[8] O. E. Drummond, D. A. Castañon and M. S. Bellovin
Comparison of 2-D assignment algorithms for sparse, rectangular, floating point, cost matrices.
In Proceedings of the SDI Panels on Tracking, vol. 4, 1990.

[9] J. Ferry
XMAP: Track-to-track association with metric, feature, and target-type data.
9th International Conference on Information Fusion, July 2006.

[10] J. Ferry
Exact bias removal for the track-to-track association problem.
12th International Conference on Information Fusion, July 2009.

[11] J. P. Ferry and L. D. Stone
Track-to-track association with missing features.
In Proceedings MSS Sensor and Data Fusion, May 2005.

[12] Roger A. Horn and Charles R. Johnson
Matrix analysis.
Cambridge University Press, 1985.

[13] R. Jonker and A. Volgenant
A shortest augmenting path algorithm for dense and sparse linear assignment problems.
Computing, **38** (1987), 325–34.

[14] L. M. Kaplan and W. D. Blair
Assignment costs for multiple sensor track-to-track association.
7th International Conference on Information Fusion, June 2004.

[15] L. M. Kaplan, W. D. Blair and Y. Bar-Shalom
Simulation studies of multisensor track association and fusion methods.
In Proceedings of the 2006 IEEE Aerospace Conference, Mar. 2006.

[16] S. Mori and C-Y. Chong
Effects of unpaired objects and sensor biases on track-to-track association: problems and solutions.
In Proceedings MSS Sensor and Data Fusion, vol. 1, 2000, 137–151.

[17] S. Mori and C-Y. Chong
Track-to-track association metric—I.I.D.–Non-Poisson cases.
6th International Conference on Information Fusion, July 2003.

[18] S. Mori and C-Y. Chong
Data association hypothesis evaluation for i.i.d. but non-Poisson multiple target tracking.
APIE Defense & Security 2004 Symposium on Signal and Data Processing for Small Targets, 2004.

[19] S. Mori and C-Y. Chong
Evaluation of data association hypotheses: Non-Poisson I.I.D. cases.
7th International Conference on Information Fusion, 2004.

[20] S. Mori and C-Y. Chong
Hybrid multiple hypothesis data fusion for missile defense system object tracking and discrimination.
In Proceedings MSS Sensor and Data Fusion, May 2005.

[21] K. G. Murty
An algorithm for ranking all the assignments in order of increasing cost.
Operations Research, **16**, 3 (1968), 682–687.

[22] R. Popp, K. Pattipati and Y. Bar-Shalom
m-best *S*-D assignment algorithm with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (2001), 22–39.

[23] D. B. Reid
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, **AC-24**, 6 (1979), 843–854.

[24] R. A. Singer and A. J. Kanyuck
Computer control of multiple site correlation.
Automatica, **7** (1971), 455–463.

[25] R. W. Sittler
An optimal data association problem in surveillance theory.
IEEE Transactions on Military Electronics, (1964), 125–139.

[26] L. Stone, T. M. Tran and M. L. Williams
Improvement in track-to-track association from using an adaptive threshold.
12th International Conference on Information Fusion, July 2009.

[27] L. D. Stone, M. L. Williams and T. M. Tran
Track-to-track association and bias removal.
SPIE AeroSense International Conference, Apr. 2002.

Jim Ferry received his S.B. in mathematics from M.I.T., and his Sc.M. and Ph.D. in applied mathematics from Brown, where he employed group representation theory to analyze the global energetics of thermal convection.

He worked for six years at the Center for Simulation of Advanced Rockets at the University of Illinois developing algorithms for simulating multiphase flows within solid rocket motors. Since coming to Metron in 2004, he has been the principal investigator of projects exploring such topics as cargo analysis, data association, network detection theory, and group tracking on dynamic networks.