Multi-target tracking using Markov chain Monte Carlo sampling

João P. Hespanha, Fellow, IEEE
Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106

Liraz Mudrik
Department of Mechanical & Aerospace Engineering, Naval Postgraduate School, Monterey, CA, 93943

Isaac Kaminer
Department of Mechanical & Aerospace Engineering, Naval Postgraduate School, Monterey, CA, 93943

***Abstract*— This paper addresses the joint estimation of the state of multiple targets based on noisy sensor measurements that lack a clear association between measurements and individual targets. We propose an algorithm that uses Markov Chain Monte Carlo (MCMC) sampling to approximate the maximum a-posteriori estimator. Specifically, the algorithm generates multi-target tracking solutions with high a-posteriori probabilities. To construct this algorithm we develop an MCMC sampling method for Boltzmann probability measures over general measure spaces, which combines Gibbs sampling with an adaptive form of parallel tempering. The effectiveness of the results is illustrated through numerical simulations, which show significant benefits with respect to the state of the art in scenarios with significant track confusion, at the expense of a moderate computational cost.**

I. Introduction

Multi-target tracking refers to the estimation of the state of multiple dynamical systems, each referred to as a *target*, using a stream of noisy partial-state measurements. When each measurement can be unambiguously associated with one of the targets, the multi-target tracking problem can be decoupled into multiple independent state estimation problems, one for each target. However, the terminology *multi-target tracking*, generally refers to scenarios where the sensors cannot distinguish between targets, in which case this decoupling is not possible and the problem's complexity is significantly worse. Such scenarios arise in many domains, including the estimation of position/velocity of aircraft based on radar data, estimation of cars and pedestrians for autonomous driving, road traffic measurement based on measurements from cameras and/or inductive loops, animal monitoring based on video feeds or infrared detectors, airport or building security to track people, and the tracking of parts in manufacturing/processing plant, among many others (see [1, 2] and references there in).

Within the multi-target tracking literature, the algorithm proposed here solves the *batch smoothing problem*, which takes a time window of measurements and produces target estimates for the whole window in a single shot. In contrast, the most widely used algorithms for multi-target tracking rely on recursive computations ("filtering") to estimate the target's state. While smoothing is generally computationally more expensive, it can yield significantly better estimates than filtering [2].

Maximum likelihood (ML) and maximum a-posteriori (MAP) are gold-standard approaches for multi-target tracking due to their strong asymptotic properties (consistency, efficiency, etc.) as well as their good performance, even when only a small number of measurements is available. However, the associated optimizations are typically nonconvex and with a strong combinatorial flavor due to the data association problem of matching sensor measurements to targets. To overcome this complexity issue, the most widely used approaches break the multi-target tracking problem into two steps: a *data association* step that tries to assign measurements to targets, followed by a per-target *filtering* step that combines all the measurements that have been associated with a single target with a motion model to reduce noise. This type of simplification is carried out to the extreme in the Global Nearest Neighbor (GNN) filter, which does data association by selecting, for each predicted target location, the (single) closest measurement. While GNN is computationally very efficient, it leads to poor performance when the targets are not well separated. The Joint Probabilistic Data Association filter (JPDAF) [1, 3], improves upon GNN by replacing a single association assignment with a joint distribution of data association probabilities. JPDAF's computational complexity grows exponentially with the number of targets and number of measurements, but can be ameliorated by introducing some form of approximation [4–10]. Among these, we highlight [10] that introduces the MCMC data association (MCMCDA) filter, which approximates the JPDAF by running a Metropolis–Hastings chain over the assignment vector, instead of enumerating all joint associations. This sampler is a fully-polynomial randomized approximation scheme, which yields association probabilities whose relative error can be made arbitrarily small, giving JPDAF-level accuracy without its exponential cost. Multiple hypotheses tracking (MHT) [11–14] defers data association at each time step and, instead maintains an evolving list of all possible data associations. The total number of hypotheses increases

June 3, 2025. This material is based upon work supported by the U.S. Office of Naval Research MURI grant n. N00014-23-1-2708; the Research Science of Autonomy Program grant n. N0001425GI01545 and N0001423WX01698; by an appointment to the NRC Research Associateship Program at the Naval Postgraduate School, administered by the Fellowships Office of the National Academies of Sciences, Engineering, and Medicine.

exponentially with time and therefore pruning/merging algorithms are needed to reduce computation.

The partition of multi-target tracking into data association and filtering is computationally very attractive but can lead to significant performance penalties in scenarios that are prone to data association errors, e.g., when measurements are very noisy and many targets share a small physical space. In this paper, we show that maximum a-posteriori (MAP) joint estimation of target positions and data association is a computationally viable approach to multi-target tracking and can obtain surprisingly good results in problems for which classical methods fail.

We focus our attention on linear target dynamics, for which, while it is straightforward to explicitly compute the exact a-posteriori probability distribution of the state (see Section II), computing the associated MAP estimator requires the solution of an optimization over a "mixed" set of variables, which includes the (continuous) target states and (discrete) variables that associate measurements to targets. Our key contribution is the development of a Markov Chain Monte Carlo (MCMC) sampling algorithm that "solves" the MAP optimization by extracting samples over the set of mixed optimization variables, using a probability measure that favors the highest values for the a-posteriori distribution.

The use of MCMC sampling to solve combinatorial optimizations dates back to pioneering work in the mid 1980s [15, 16], where it was proposed to regard the minimization criterion as the energy for a Boltzmann distribution, which would naturally assign the highest probability to the lowest cost. This approach has recently met with renewed interest due to the observation that it can explore hardware parallelization to greatly speed up the sampling process, making it very attractive for difficult optimization problems [17–19].

The application of MCMC methods to multi-target tracking requires (1) the introduction of a Boltzmann distribution over a general measure space that includes both continuous and discrete variables and (2) the construction of a Markov chain whose state converges to this distribution (see Section III). To accomplish this, we propose a transition kernel over general measure spaces that generalizes Gibbs sampling [20, 21] with parallel tempering [22, 23]. Gibbs kernels facilitate sampling by working on one subset of variables at a time, and take advantage of the observation that it is generally much easier to extract samples from a single "simple" distribution (in our case Gaussian or discrete), rather than from the joint distribution of many (correlated) random variables. Parallel tempering improves the mixing time of the Markov chain, by drawing samples for a joint Boltzmann distribution over a range of temperatures. The "low temperatures" provide the desired samples with minimum energy/cost, whereas the high temperatures provide high-probability "shortcuts" between different regions of the state-space. By exchanging samples between different temperatures, it is possible to maintain the desired invariant distribution for the low-temperature chain, while permitting samples to "easily" jump from one low-cost/energy location to another low-cost/energy location, without passing through high-cost/energy regions.

The choice of temperatures for parallel tempering uses a new adaptive algorithm to automatically adjust the temperatures to provide optimal mixing. Inspired by [24], we keep the lowest temperature constant and adjust the remaining temperatures to maintain the same sample-exchange probability between all adjacent temperatures. However, we differ from [24] in that, rather than deriving an optimal temperature schedule for a specific chemical systems of interest, we use feedback to continuously adjust the temperatures to achieve a specific target swap probability. The use of feedback to adjust temperatures was previously proposed in [25] to maintain exchange probabilities within bounds and in [26] to minimize the samples "round-trip time" between the two extreme temperatures.

The development of MCMC sampling methods for Boltzmann distributions over general measurable spaces is applicable to optimization problems beyond multi-target tracking and is therefore presented in Section IV for optimizations more general than those arising in multi-target tracking. When applied to solve the multi-target tracking MAP estimation problem, we obtain in Section V an algorithm that produces samples with high a-posteriori probability with modest computation.

We show through numerical examples in Section VI that the quality of the estimates of our algorithm compares favorably with the classical JPDA [3] and the MCMCDA [10] algorithms, both using the Rauch–Tung–Striebel (RTS) smoother [27], *even when those algorithms use ground-truth track initializations and our algorithm does not.* The downside of our algorithm is a higher computation cost. However, the computation time required to solve a problem with 1000 measurements that must be disambiguated among 10 targets (corresponding to 10^{1000} possibilities for data association) still takes less than 30sec on a laptop computer.

It is important to emphasize that the results reported here require the integration of several independent contributions: MCMC sampling over mixed spaces using the proposed Gibbs and parallel tempering kernels, adaptive tempering to adjust temperatures, and a new algorithm that uses parallel computation to significantly reduce the computation time. Without any one of these, the resulting algorithms would not be competitive.

The remainder of this paper is organized as follows: The basic multi-target tracking problem and its associated maximum a-posteriori estimator are introduced in Section II. Section III considers a generic optimization problem and shows how a temperature-dependent Boltzmann distribution can be used to extract optimal samples with high-probability. This distribution is defined over

a general measure space, so that it can be used for the "mixed" discrete/continuous spaces needed for multi-target tracking. Section IV contains the key technical results of the paper, which include the constructions of the Gibbs and parallel tempering Markov Kernels, and the adaptive tempering algorithm. These are combined in Section V into an algorithm that can be used to extract estimates for the multi-target tracking with high a-posteriori probability, which is validated in Section VI through numerical experiments. Section VII concludes the paper with an outlook for future work.

II. Multi-target tracking using maximum a-posteriori estimation

We consider linear target dynamics of the form

$$x_{t+1}^{\ell} = Ax_t^{\ell} + d_t^{\ell}, \quad \forall \ell \in \mathcal{L} \coloneqq \{1, \dots, N\}, \quad \forall t \in \{1, \dots, T-1\}, \quad (1)$$

where $x_t^{\ell} \in \mathbb{R}^{n_x}$ denotes the state of target ℓ at time t and $d_t^{\ell} \in \mathbb{R}^{n_x}$ a zero-mean uncorrelated Gaussian process. To estimate the target states, we have a sequence of time-stamped measurements $\{(t_1, y_1), \dots, (t_K, y_K)\}$ that satisfy

$$y_k = Cx_{t_k}^{\sigma_k} + n_k, \quad \forall k \in \{1, \dots, K\}, \quad (2)$$

for an unknown sequence $\sigma \coloneqq (\sigma_1, \dots \sigma_K)$ that associates the measurement $y_k \in \mathbb{R}^{n_y}$ with the target $\sigma_k \in \mathcal{L}$; and where $n_k \in \mathbb{R}^{n_y}$ denotes a zero-mean uncorrelated Gaussian process, which represents measurement noise and is independent of the d_t^{ℓ}. We assume that the target dynamics matrix A, the measurement matrix C, and the covariance matrices

$$\mathrm{E}[x_1^{\ell} x_1^{\ell\prime}] = \Sigma_1 \in \mathbb{R}^{n_x \times n_x}, \quad \forall \ell \in \mathcal{L}, \quad (3a)$$
$$\mathrm{E}[d_t d_t'] = \Sigma_d \in \mathbb{R}^{n_x \times n_x}, \quad \forall t \in \{1, \dots, T-1\}, \quad (3b)$$
$$\mathrm{E}[n_k n_k'] = \Sigma_n \in \mathbb{R}^{n_y \times n_y}, \quad \forall k \in \{1, \dots, K\}, \quad (3c)$$

are all known; and also that the state representation has been chosen so that $\mathrm{E}[x_1^{\ell}] = 0$, $\forall \ell \in \mathcal{L}$. Under this model, our goal is to estimate the state sequences

$$X \coloneqq \{x_t^{\ell} : t \in \{1, \dots, T\},\ \ell \in \mathcal{L}\} \in \mathbb{R}^{NTn_x}$$

and the measurement index sequence

$$\sigma \coloneqq (\sigma_1, \dots \sigma_K) \in \mathcal{L}^K.$$

For the model above, the disturbances d_t^{ℓ} and the noise n_k are uniquely determined by the states in X and the sequence σ, which enables us to write the logarithm of the joint conditional probability density function (pdf) of X and $Y \coloneqq \{y_k : k \in \{1, \dots, K\}\}$, given σ, as

$$\log p(X, Y|\sigma) = \bar{c} - \frac{1}{2}\sum_{\ell=1}^{N} x_1^{\ell\prime}\Sigma_1^{-1}x_1^{\ell} - \frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T-1}(x_{t+1}^{\ell} - Ax_t^{\ell})'\Sigma_d^{-1}(\underbrace{x_{t+1}^{\ell} - Ax_t^{\ell}}_{d_t^{\ell}}) - \frac{1}{2}\sum_{k=1}^{K}(y_k - Cx_{t_k}^{\sigma_k})'\Sigma_n^{-1}(\underbrace{y_k - Cx_{t_k}^{\sigma_k}}_{n_k})$$

for an appropriately defined normalizing constant $\bar{c}$, selected so that this pdf integrates to 1 (see, e.g., [28]). Bayes' rule then allows us to obtain the logarithm of the a-posteriori distribution of X and σ as

$$\log p(X, \sigma|Y) = \log \frac{p(X, Y|\sigma)p_{\sigma}(\sigma)}{p_Y(Y)} = c + \log p_{\sigma}(\sigma) - \frac{1}{2}\sum_{\ell=1}^{N} x_1^{\ell\prime}\Sigma_1^{-1}x_1^{\ell} - \frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T-1}(x_{t+1}^{\ell} - Ax_t^{\ell})'\Sigma_d^{-1}(x_{t+1}^{\ell} - Ax_t^{\ell}) - \frac{1}{2}\sum_{k=1}^{K}(y_k - Cx_{t_k}^{\sigma_k})'\Sigma_n^{-1}(y_k - Cx_{t_k}^{\sigma_k}), \quad (4)$$

where $p_Y(Y)$ denotes the marginal distribution of Y, $p_{\sigma}(\sigma)$ the a-priori distribution of σ, and $c \coloneqq \bar{c} - \log p_Y(Y)$.

For a given sequence σ, maximizing (4) with respect to the state sequences in X is a straightforward (convex) quadratic minimization, as noted in the proposition that follows, which is proved in the Appendix. However, the joint optimization of the continuous states in X and the discrete sequence σ is not and the remainder of this paper is devoted precisely to address this challenge.

PROPOSITION 1 *For a fixed $\sigma \coloneqq (\sigma_1, \dots \sigma_K) \in \mathcal{L}^K$, the maximum of (4) with respect to the state-sequences vector $X \in \mathbb{R}^{NT}$ is given by*

$$c + \log p_{\sigma}(\sigma) - q^*(\sigma)$$

where

$$q^*(\sigma) \coloneqq \frac{1}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k - \frac{1}{2}b(\sigma)'\big(M_0 + M_1(\sigma)\big)^{-1}b(\sigma) \quad (5)$$

and the maximum takes place at

$$X^*(\sigma) = \big(M_0 + M_1(\sigma)\big)^{-1}b(\sigma), \quad (6)$$

with

$$M_0 \coloneqq \sum_{\ell=1}^{N}\sum_{t=1}^{T}(E_{t+1}^{\ell} - AE_t^{\ell})'\Sigma_d^{-1}(E_{t+1}^{\ell} - AE_t^{\ell}) + \sum_{\ell=1}^{N} E_1^{\ell}\Sigma_1^{-1}E_1^{\ell} \quad (7)$$

$$M_1(\sigma) \coloneqq \sum_{k=1}^{K}(CE_{t_k}^{\sigma_k})'\Sigma_n^{-1}CE_{t_k}^{\sigma_k},$$

$$b(\sigma) \coloneqq \sum_{k=1}^{K}(CE_{t_k}^{\sigma_k})'\Sigma_n^{-1}y_k;$$

where $E_t^{\ell} \in \mathbb{R}^{n_x \times (NT)}$ denotes a matrix that "extracts" from X the state x_t^{ℓ}, as in $x_t^{\ell} = E_t^{\ell}X$. □

Extensions

Several extensions to the problem formulation above are supported by the methods developed in the remaining of this paper. For simplicity of presentation (and notation), we do not pursue them fully but enumerate them below:

a: *Heterogeneous targets*

The model defined by (1), (2), and (3) assumes that the targets are "homogeneous" in the sense that the model matrices A, C and covariances $\Sigma_1, \Sigma_d, \Sigma_n$ are the same for every target $\ell \in \mathcal{L}$. However, the problem could be generalized for heterogeneous targets by simply indexing all these matrices by the target index ℓ in Proposition 1 and in the results that follow from it.

b: *False alarms*

The model above assumes that every measurement y_k arises from some target $\sigma_k \in \mathcal{L}$. In practice, some measurements may be "false alarms" in the sense that they are not associated with any target. This can be addressed, by considering a "dummy target" (say corresponding to the first index $\ell = 1$) with "known" $x_t^\ell = 0$, $\forall t$ and for which the noise covariance matrix Σ_n^ℓ reflects the distribution of the "false alarms." We used a superscript $\cdot^\ell$ in this covariance matrix, because this would typically require "heterogeneous" noise covariances to distinguish the distribution of the false alarms from that of the noise associated with regular targets.

c: *A-priori distribution of σ*

While (4) considers a general joint distribution for the sequence measurement index sequence σ, in some of the results that follow we assume that this distribution is uniform. This assumption simplifies some of the formulas because it enables us to absorb $p_\sigma(\sigma)$ into the constant c, which essentially means that we can remove $p_\sigma(\sigma)$ from (4). However, there are several other alternatives that lead to (longer) but still simple formulas. A common distribution arises from assuming that the K measurements are obtained from independent trials, each of which picks target ℓ with the probability p_ℓ. In this case, the number of times that each target is selected is given by a multinomial distribution and we have that $p_\sigma(\sigma)$ is proportional to

$$p_\sigma(\sigma) \propto \frac{K!}{n_1(\sigma)! \cdots n_L(\sigma)!} p_1^{n_1(\sigma)} \cdots p_L^{n_L(\sigma)},$$

where $n_\ell(\sigma)$ denotes the number of time that the target ℓ appears in the sequence σ. For the homogeneous case with all $p_\ell = 1/L$, this simplifies to $p_\sigma(\sigma)$ proportional to

$$p_\sigma(\sigma) \propto \frac{1}{n_1(\sigma)! \cdots n_L(\sigma)!}. \tag{8}$$

III. Boltzmann sampling

Consider a general minimization of the form

$$J^* \coloneqq \min_{z \in \mathcal{Z}} J(z). \tag{9}$$

To solve this problem using sampling, we would like to construct a sampler that "selects" points $z \in \mathcal{Z}$ with high probability when $J(z)$ is low. When the set $\mathcal{Z}$ is finite, it is common to draw samples from a random variable Z with a Boltzmann distribution with energies defined by the cost $J(z)$, $z \in \mathcal{Z}$:

$$\mathrm{P}(Z = z) = \frac{e^{-\beta J(z)}}{Q(\beta)}, \ \forall z \in \mathcal{Z}, \quad Q(\beta) \coloneqq \sum_{\bar{z} \in \mathcal{Z}} e^{-\beta J(\bar{z})},$$

where $\beta \geqslant 0$ is the *temperature parameter*. This choice for the distribution guarantees that, as $\beta \to \infty$, the probability of getting a sample at a minima of $J(z)$ becomes much larger than any other sample with a higher cost.

The extension of this approach to more general sets $\mathcal{Z}$ requires considering an "ambient" measure space $(\mathcal{Z}, \mathcal{F}, \mu)$ and defining a temperature-dependent *Boltzmann probability measure* $\pi(\cdot; \beta)$ on this space as follows:

$$\pi(A; \beta) \coloneqq \int_{\mathcal{Z}} \mathbf{1}_A(z) \frac{e^{-\beta J(z)}}{Q(\beta)} \mu(dz), \quad \forall A \in \mathcal{F},$$
$$Q(\beta) \coloneqq \int_{\mathcal{Z}} e^{-\beta J(z)} \mu(dz), \tag{10}$$

where $\mathbf{1}_A : \mathcal{Z} \to \{0, 1\}$ is the indicator function of the set A.

For multi-target tracking, the maximization of the MAP criterion (4) combines continuous variables (corresponding to the state sequences in X) with discrete variables (corresponding to the sequence σ). This corresponds to sets $\mathcal{Z}$ that are "mixed" in the sense that $\mathcal{Z} = \mathbb{R}^n \times \mathcal{S}$ with $\mathcal{S} = \{s_1, \ldots, s_N\}$ finite. In this case, we can pick μ to be the product of the Lebesgue measurable μ_{Leb} in $\mathbb{R}^n$ with the discrete measure

$$\mu_{\mathcal{S}}(A) \coloneqq \sum_{i=1}^{N} \mathbf{1}_A(s_i), \quad \forall A \subset \mathcal{S}, \tag{11}$$

which leads to a distribution in (10) of the form

$$\pi(A; \beta) \coloneqq \sum_{i=1}^{N} \int_{\mathbb{R}^n} \mathbf{1}_A(z, s_i) \frac{e^{-\beta J(z, s_i)}}{Q(\beta)} dz,$$
$$Q(\beta) \coloneqq \sum_{i=1}^{N} \int_{\mathbb{R}^n} e^{-\beta J(z, s_i)} dz, \quad \forall A \in \mathcal{F}. \tag{12}$$

We refer the reader to [29, p. 62] for the formal definition of product measures.

The rationale for the general measure in (10) is easier to understand in the context of the special case (12): If the two points $z_1 = (x_1, s_1), z_2 = (x_2, s_2) \in \mathcal{Z} \coloneqq \mathbb{R}^n \times \mathcal{S}$ leads to different costs, say $J(x_1, s_1) < J(x_2, s_2)$, then, as $\beta \to \infty$, the probability of getting a sample in a small ball around the lower-cost point (x_1, s_1) becomes much larger than in a ball around the higher-cost point (x_2, s_2). This can be confirmed by noting that, for some small radius $\epsilon > 0$, the distribution in (12) leads to

$$\frac{\pi(B_\epsilon(x_1) \times \{s_1\}; \beta)}{\pi(B_\epsilon(x_2) \times \{s_2\}; \beta)} = \frac{\int_{B_\epsilon(x_1)} e^{-\beta J(x, s_1)} dx}{\int_{B_\epsilon(x_2)} e^{-\beta J(x, s_2)} dx}.$$

If the criterion $J(x,s)$ is continuous with respect to x, this means that

$$\lim_{\epsilon\downarrow 0^+} \frac{\pi(B_\epsilon(x_1)\times\{s_1\};\beta)}{\pi(B_\epsilon(x_2)\times\{s_2\};\beta)} = e^{-\beta(J(x_1,s_1)-J(x_2,s_2))},$$

which confirms that, as $\beta\to\infty$, the probability of getting a sample in a small ball around the lower-cost point (x_1,s_1) becomes much larger than around the higher-cost point (x_2,s_2).

IV. Markov chain Monte Carlo sampling

Our approach to maximize the MAP criterion (4) relies on extracting samples from a Markov chain whose state-space $\mathcal{Z}$ combines continuous and discrete variables. To that effect we start by reviewing a few key results on Markov chains with non-countable state-spaces following terminology adapted from [30, 31].

A. Markov Chains on general spaces

A *(1-step) transition probability kernel defined on a measurable space* $(\mathcal{Z},\mathcal{F})$ is a function $P(A,z)$, $z\in\mathcal{Z}$, $A\in\mathcal{F}$ with the properties that

(i) $P(\cdot,z)$ is a probability measure on $(\mathcal{Z},\mathcal{F})$, for every $z\in\mathcal{Z}$; and

(ii) $P(A,\cdot)$ is a non-negative measurable function on $\mathcal{Z}$ for every $A\in\mathcal{F}$.

Given a (1-step) transition probability kernel P, we can recursively define the *K-step transition probability kernel* P^k for every $k\geqslant 0$ using

$$P^o(A,z) \coloneqq \mathbf{1}_A(z),$$
$$P^{k+1}(A,z) \coloneqq \int_{\mathcal{Z}} P^k(A,y)P(dy,z), \quad \forall k\geqslant 1,$$

$\forall z\in\mathcal{Z}, A\in\mathcal{F}$. A transition kernel defines a *(discrete-time time-homogeneous) Markov chain*, which is a stochastic process $\{\boldsymbol{z}_1,\boldsymbol{z}_2,\dots\}$ taking values on $\mathcal{Z}$ such that

$$\mathrm{P}(\boldsymbol{z}_{k+1}\in A \mid \boldsymbol{z}_k,\dots,\boldsymbol{z}_1) = P(A,\boldsymbol{z}_k), \quad \forall A\in\mathcal{F}$$

and, more generally,

$$\mathrm{P}(\boldsymbol{z}_{k+K}\in A \mid \boldsymbol{z}_k,\dots,\boldsymbol{z}_1) = P^K(A,\boldsymbol{z}_k), \quad \forall A\in\mathcal{F},\ k,K\geqslant 0$$

[30, Section 3.4]. A probability measure π on $(\mathcal{Z},\mathcal{F})$ is an *invariant distribution for the transition probability kernel* P if

$$\pi(A) = \int_{\mathcal{Z}} P(A,z)\pi(dz). \tag{13}$$

It turns out that, under appropriate assumptions, the invariant distribution is unique and the distribution of the state $\boldsymbol{z}_k$ converges to it. To formalize these assumptions, we need the following definitions: A transition kernel P is called *φ-irreducible* for a measure φ on $(\mathcal{Z},\mathcal{F})$ if

$$\varphi(A)>0 \quad\Rightarrow\quad \exists n_{z,A}>0 : P^{n_{z,A}}(A,z)>0, \tag{14}$$

$\forall z\in\mathcal{Z}$, $A\in\mathcal{F}$ [30, Proposition 4.2.1]; a φ-irreducible transition kernel P is called *aperiodic* if there exists no sequence of nonempty disjoint sets $\{E_0,E_1,\dots,E_{d-1}\}\subset\mathcal{F}$ with $d\geqslant 2$ elements such that

$$P(E_{(i+1) \bmod d}, z) = 1, \quad \forall z\in E_i,\ i\in\{0,\dots,d-1\},$$

[31]; and a measure ψ on the measurable space $(\mathcal{Z},\mathcal{F})$ is absolutely continuous with respect to another measure φ on the same space if

$$\varphi(A)=0 \quad\Rightarrow\quad \psi(A)=0, \quad \forall A\in\mathcal{F}.$$

We are now ready to present the result mentioned above

THEOREM 1 ([31, Theorem 1, Corollary 1]) *Consider a transition kernel P defined on a measurable space $(\mathcal{Z},\mathcal{F})$, with an invariant distributions π and assume that*

1) *P is π-irreducible, i.e., $\forall A\in\mathcal{F}$, $z\in\mathcal{Z}$,*
$$\pi(A)>0 \quad\Rightarrow\quad \exists n>0 : P^n(A,z)>0;$$
2) *$P(\cdot,z)$ is absolutely continuous with respect to π, for every $z\in\mathcal{Z}$, i.e.,*
$$\forall A\in\mathcal{F},\ z\in\mathcal{Z}: \quad \pi(A)=0 \quad\Rightarrow\quad P(A,z)=0$$
3) *P is aperiodic;*

then π is the unique invariant distribution of P and

$$\lim_{n\to\infty} \|P^n(\cdot,z)-\pi\| = 0, \quad \forall z\in\mathcal{Z}. \tag{15}$$

This result lays the path for what follows: we need to construct a Markov transition kernel P with invariant distribution π given by (10), so that samples from this chain favor with high probability values with small cost. The approach pursued here to accomplish this combines three ingredients: Gibbs kernels, parallel tempering kernels, and adaptive temperatures adjustment; which are described in the subsequent sections.

B. Gibbs transition kernels

The basic idea behind Gibbs kernels consists of decomposing the state-space $\mathcal{Z}$ as the Cartesian product of multiple subspaces $\mathcal{Z}_1,\dots,\mathcal{Z}_L$, each associated with a subset of the optimization variables. For each of these subspaces $\mathcal{Z}_i$, we construct a transition kernel P_i^β that only affect the variables in $\mathcal{Z}_i$ and that is invariant for the desired distribution π given by (10). The composition $P_1^\beta\cdots P_L^\beta$ of all these kernels will continue to be invariant for π, but will now affect all variables. This type of construction is the basis for Gibbs sampling and benefits for the fact that it can be much easier to sample for "simple" individual random variables, rather than from the joint distribution of many random variables. In our problem, the "simple" random variables will be Gaussian and discrete random variables.

To formalize this idea, consider an ambient measure space $(\mathcal{Z},\mathcal{F},\mu)$ and suppose that we want to sample from a distribution of the form (10). Our goal is to construct a transition probability kernels $P(A,z)$ for which (10) is an invariant distribution, i.e., for which

$$\pi(A;\beta) = \int_{\mathcal{Z}} P(A,z)\pi(dz;\beta), \quad \forall A\in\mathcal{F}.$$

When $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2$ is the Cartesian product of two sets and $\mu = \mu_1 \times \mu_2$ is the product of two measures on the measurable spaces $(\mathcal{Z}_1, \mathcal{F}_1)$, $(\mathcal{Z}_2, \mathcal{F}_2)$, we can define the following transition probability kernel

$$P(A, (z_1, z_2)) \coloneqq \int_{\mathcal{Z}_1} \mathbf{1}_A(\bar{z}_1, z_2) \frac{e^{-\beta J(\bar{z}_1, z_2)}}{Q_2(\beta, z_2)} \mu_1(d\bar{z}_1), \quad \forall z_1 \in \mathcal{Z}_1, z_2 \in \mathcal{Z}_2, A \in \mathcal{F}, \quad (16)$$

on the product measurable space $(\mathcal{Z}, \mathcal{F}) = (\mathcal{Z}_1 \times \mathcal{Z}_2, \mathcal{F}_1 \otimes \mathcal{F}_2)$ with

$$Q_2(\beta, z_2) \coloneqq \int_{\mathcal{Z}_1} e^{-\beta J(\bar{z}_1, z_2)} \mu_1(d\bar{z}_1), \ \forall z_2 \in \mathcal{Z}_2.$$

In essence, the transition kernel (16) preserves the value of the variable $z_2 \in \mathcal{Z}_2$ and ignores the "old" value of $z_1 \in \mathcal{Z}_1$, which is resampled with the distribution

$$\mathrm{P}(z_1 \in A_1) = \int_{\mathcal{Z}_1} \mathbf{1}_{A_1}(\bar{z}_1) \frac{e^{-\beta J(\bar{z}_1, z_2)}}{Q_2(\beta, z_2)} \mu_1(d\bar{z}_1), \ A_1 \in \mathcal{F}_1 .$$

When μ_1 is a Lebesgue measure in $\mathcal{Z}_1 \coloneqq \mathbb{R}^n$, this corresponds to a continuous distribution with probability density function given by $\frac{e^{-\beta J(\cdot, z_2)}}{Q_2(\beta, z_2)}$; and when μ_1 is a uniform discrete measure, this corresponds to a discrete distribution with probability weights given by $\frac{e^{-\beta J(\bar{z}_1, z_2)}}{Q_2(\beta, z_2)}$, $\bar{z}_1 \in \mathcal{Z}_1$.

The formula in (16) is formally justified by the following result, which essentially states that (16) defines a transition kernel with the desired invariant distribution (10).

THEOREM 2 *Assume that the function $z \mapsto e^{-\beta J(z)}$ belongs to $L_1(\mu)$ and that the function $z_1 \mapsto e^{-\beta J(z_1, z_2)}$ belongs to $L_1(\mu_1)$, $\forall z_2 \in \mathcal{Z}_2$. Then (16) defines a transition probability kernel and (10) defines a probability measure that is invariant for that kernel.* □

For the multi-target tracking problem outlined in Section II, the state-space of the Markov chain is of the form $\mathcal{Z} \coloneqq \mathbb{R}^{NTn_x} \times \mathcal{L}^K$ that we can decompose as a Cartesian product in different ways, each leading to a distinct Gibbs transition kernel. The following two decompositions will be used later in the paper:

1) Regarding the set $\mathcal{Z} \coloneqq \mathbb{R}^{NTn_x} \times \mathcal{L}^K$ as the Cartesian product

$$\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2, \quad \mathcal{Z}_1 \coloneqq \mathbb{R}^{NTn_x}, \quad \mathcal{Z}_2 \coloneqq \mathcal{L}^K$$

where $\mathcal{Z}_1$ is associated with the vector X and $\mathcal{Z}_2$ with all the σ_k; the transition kernel (16) preserves the value of the variables $\sigma_1, \ldots, \sigma_K \in \mathcal{L}$ and ignores the "old" value of $X \in \mathbb{R}^{NTn_x}$, which is resampled with a probability density function given by

$$f(\bar{X}) = \frac{e^{-\beta J(\bar{X}, \sigma_1, \ldots, \sigma_K)}}{Q_2(\beta, \sigma_1, \ldots, \sigma_K)}, \quad \bar{X} \in \mathbb{R}^{NTn_x}, \quad (17)$$

with $Q_2(\beta, \sigma_1, \ldots, \sigma_K)$ selected so that this density integrates to 1. For a cost function $J(X, \sigma)$ given by the negative of the MAP criterion (4), the density function in (17) corresponds to a multivariable Gaussian distribution with mean $X^*(\sigma)$ and covariance matrix $\beta^{-1}\big(M_0 + M_1(\sigma)\big)^{-1}$ (see Propositions 1 and 2).

2) Regarding the set $\mathcal{Z} \coloneqq \mathbb{R}^{NTn_x} \times \mathcal{L}^K$ as the Cartesian product

$$\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2, \quad \mathcal{Z}_1 \coloneqq \mathcal{L}, \quad \mathcal{Z}_2 \coloneqq \mathbb{R}^{NTn_x} \times \mathcal{L}^{K-1}$$

where $\mathcal{Z}_1$ is associated with σ_k and $\mathcal{Z}_2$ with X and the remaining $\sigma_{-k} \coloneqq (\sigma_1, \ldots, \sigma_{k-1}, \sigma_{k+1}, \ldots, \sigma_K)$; the transition kernel (16) preserves the value of the variables $X, \sigma_{-k} \in \mathcal{Z}_2$ and ignores the "old" value of $\sigma_k \in \mathcal{L}$, which is resampled with probabilities

$$p(\bar{\sigma}_k) = \frac{e^{-\beta J(\bar{\sigma}_k, X, \sigma_{-k})}}{Q_2(\beta, X, \sigma_{-k})}, \quad \forall \bar{\sigma}_k \in \mathcal{L},$$

with $Q_2(\beta, X, \sigma_{-k})$ selected so that these probabilities add up to 1. For a cost function $J(X, \sigma)$ given by the negative of the MAP criterion (4), these probabilities are of the form

$$p(\bar{\sigma}_k) = \bar{c} \exp\Big(\beta \log p_\sigma(\bar{\sigma}_k, \sigma_{-k}) + - \frac{\beta}{2}(y_k - C x_{t_k}^{\bar{\sigma}_k})' \Sigma_n^{-1} (y_k - C x_{t_k}^{\bar{\sigma}_k})\Big), \ \forall \bar{\sigma}_k \in \mathcal{L}, \quad (18)$$

where $\bar{c}$ is a normalizing constant that does not depend on $\bar{\sigma}_k$ and should thus be selected so that these probabilities add up to 1.

When $p_\sigma(\cdot)$ is a uniform distribution, the term $\beta \log p_\sigma(\bar{\sigma}_k, \sigma_{-k})$ in (18) is constant and can be removed by absorbing it into the constant $\bar{c}$. Alternatively, when $p_\sigma(\cdot)$ comes from the multinomial distribution (8), we have

$$\frac{p_\sigma(\bar{\sigma}_k, \sigma_{-k})}{p_\sigma(\sigma)} = \frac{n_1(\sigma)! \cdots n_L(\sigma)!}{n_1(\bar{\sigma}_k, \sigma_{-k})! \cdots n_L(\bar{\sigma}_k, \sigma_{-k})!} = \begin{cases} 1 & \bar{\sigma}_k = \sigma_k \\ \frac{n_{\sigma_k}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1} & \bar{\sigma}_k \neq \sigma_k \end{cases}$$

and therefore

$$\log p_\sigma(\bar{\sigma}_k, \sigma_{-k}) = \log p_\sigma(\sigma) + \begin{cases} 0 & \bar{\sigma}_k = \sigma_k \\ \log \frac{n_{\sigma_k}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1} & \bar{\sigma}_k \neq \sigma_k. \end{cases}$$

In this case, (18) simplifies to

$$\begin{aligned} p(\bar{\sigma}_k) &= \bar{c} e^{-\frac{\beta}{2}(y_k - C x_{t_k}^{\bar{\sigma}_k})' \Sigma_n^{-1} (y_k - C x_{t_k}^{\bar{\sigma}_k})} \times \begin{cases} 1 & \bar{\sigma}_k = \sigma_k, \\ e^{\beta \log \frac{n_{\sigma_k}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1}} & \bar{\sigma}_k \neq \sigma_k, \end{cases} \\ &= \bar{c} e^{-\frac{\beta}{2}(y_k - C x_{t_k}^{\bar{\sigma}_k})' \Sigma_n^{-1} (y_k - C x_{t_k}^{\bar{\sigma}_k})} \times \begin{cases} 1 & \bar{\sigma}_k = \sigma_k, \\ \left(\frac{n_{\sigma_k}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1}\right)^\beta & \bar{\sigma}_k \neq \sigma_k, \end{cases} \quad \forall \bar{\sigma}_k \in \mathcal{L}, \end{aligned} \quad (19)$$

where we absorbed the term $e^{\beta \log p_\sigma(\sigma)}$ into the normalizing constant $\bar{c}$. In essence, the term

$\left(\frac{n_{\sigma_k}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1}\right)^{\beta}$ favors moving σ_k to $\bar{\sigma}_k$ when $n_{\sigma_k}(\sigma)$ is large and $n_{\bar{\sigma}_k}(\sigma)$ is small; which has an "equalizing" effect on the distribution of measurements across targets.

C. Parallel tempering kernels

As noted in Section III, to obtain a minimum to (9) with high probability, we should pick a large value for the temperature parameter β in the distribution (10). However, the convergence guaranteed by (15) is typically very slow for large values of β. Intuitively, this is because the probability $P^n(A,z)$ of reaching the set A from the state z in n steps becomes very small if the path from z to A needs to go through states with high cost. Note that, when β is large, such states would have very low probability. Essentially, as β grows, the chain starts to lose π-irreducibility, as defined in (14).

To mitigate this, rather than sampling with a single temperature parameter β, we select a range of parameters

$$\mathcal{B} \coloneqq \{\beta_1 < \beta_2 < \beta_3 < \cdots < \beta_M\}, \tag{20}$$

and draw samples from an M-tuple of random variables

$$\boldsymbol{z} \coloneqq (\boldsymbol{z}^{\beta_1}, \ldots, \boldsymbol{z}^{\beta_M}) \in \mathcal{Z}^M,$$

with a desired joint distribution

$$\pi(A;\mathcal{B}) \coloneqq \int_{\mathcal{Z}} \cdots \int_{\mathcal{Z}} \mathbf{1}_A(z^{\beta_1}, \ldots, z^{\beta_M}) \times \frac{e^{-\beta_1 J(z^{\beta_1})}}{Q(\beta_1)} \cdots \frac{e^{-\beta_M J(z^{\beta_M})}}{Q(\beta_M)} \mu(dz^{\beta_M}) \cdots \mu(dz^{\beta_1}), \tag{21}$$

$\forall A \in \mathcal{F}\otimes\cdots\otimes\mathcal{F}$. This corresponds to independent distributions for the $\boldsymbol{z}_k^{\beta_i}$, each with a different temperature parameter $\beta_i \in \mathcal{B}$. The independent sampling of the different chains, can be viewed as a joint "diagonal" transition kernel P^{diagonal} and the terminology "parallel tempering" arises from the observation that this independent sampling can be carried out in parallel.

To take advantage of the multiple temperatures, the "diagonal" transition kernel P^{diagonal} is composed with a "tempering kernel" P^{flip} that flips samples across different temperatures, while still preserving the desired joint invariant distribution. For optimization purposes, we only use the samples for $\boldsymbol{z}_k^{\beta_M}$ that favor low costs the most. However, the tempering kernel brings to the highest-β chain $\boldsymbol{z}_k^{\beta_M}$ samples with low cost that may have been "discovered" by the lower β chains. The low value for the temperature parameter permits these chains to explore the space $\mathcal{Z}$ much faster and makes sure that the joint chain does not come close to losing π-irreducibility, in spite of large values for β_M.

The tempering kernel for (21) will actually be composed of multiple simpler kernels, each flipping samples just between two adjacent temperatures β_i and β_{i+1}. To construct such a two-temperature version of (21), we start with an ambient space $(\mathcal{Z},\mathcal{F},\mu)$ and consider the joint distribution on the product measure space $(\mathcal{Z}\times\mathcal{Z}, \mathcal{F}\otimes\mathcal{F}, \mu\times\mu)$ given by

$$\pi(A;J_i,J_{i+1}) \coloneqq \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1}) \frac{e^{-J_i(z_i)}}{Q(J_i)} \times \frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})} \mu(dz_{i+1})\mu(dz_i), \forall A \in \mathcal{F}\otimes\mathcal{F}, \tag{22}$$

where $\mathbf{1}_A : \mathcal{Z}\times\mathcal{Z} \to \{0,1\}$ is the indicator function of the set A;

$$Q(J_i) \coloneqq \int_{\mathcal{Z}} e^{-J_i(z)}\mu(dz);$$

and $J_i, J_{i+1} \geqslant 0$ are two parameter-dependent energy functions, typically each associated with a specific temperature. In particular, to flip samples between the chains associated with β_i and β_{i+1}, we would consider

$$J_i(z) \coloneqq \beta_i J(z), \quad J_{i+1}(z) \coloneqq \beta_{i+1} J(z), \quad \forall z \in \mathcal{Z}. \tag{23}$$

Our goal is now to construct transition probability kernels $P^{\text{flip}}(A,(z_i,z_{i+1}))$ on the product measurable space $(\mathcal{Z}\times\mathcal{Z}, \mathcal{F}\otimes\mathcal{F})$ for which (22) is an invariant distribution, i.e., for which

$$\pi(A;J_i,J_{i+1}) = \int_{\mathcal{Z}} P^{\text{flip}}(A,(z_i,z_{i+1}))\pi(dz;J_i,J_{i+1}), \quad \forall A \in \mathcal{F}. \tag{24}$$

To this effect, we define

$$P^{\text{flip}}(A,(z_i,z_{i+1})) = (1-p_{\text{flip}}(z_i,z_{i+1}))\mathbf{1}_A(z_i,z_{i+1}) + p_{\text{flip}}(z_i,z_{i+1})\mathbf{1}_A(z_{i+1},z_i), \tag{25}$$

$\forall z_i, z_{i+1} \in \mathcal{Z}, A \in \mathcal{F}\otimes\mathcal{F}$ for some function $p_{\text{flip}} : \mathcal{Z}\times\mathcal{Z} \to [0,1]$. This transition kernel essentially replaces an "old" state $(z_i,z_{i+1}) \in \mathcal{Z}\times\mathcal{Z}$ by the flipped state $(z_{i+1},z_i) \in \mathcal{Z}\times\mathcal{Z}$ with probability $p_{\text{flip}}(z_i,z_{i+1})$ and, otherwise, keeps the original state.

THEOREM 3 1) *When $p_{\text{flip}} : \mathcal{Z}\times\mathcal{Z} \to [0,1]$ is $\mathcal{F}\otimes\mathcal{F}$-measurable, (25) defines a transition probability kernel on $(\mathcal{Z}\times\mathcal{Z}, \mathcal{F}\otimes\mathcal{F})$.*

2) *When the functions $z_i \mapsto e^{-J_i(z_i)}$ and $z_{i+1} \mapsto e^{-J_{i+1}(z_{i+1})}$ belong to $L_1(\mu)$, (22) defines a probability measure on $(\mathcal{Z}\times\mathcal{Z}, \mathcal{F}\otimes\mathcal{F}, \mu\times\mu)$.*

3) *When the two conditions above hold and, in addition,*

$$p_{\text{flip}}(z_{i+1},z_i)e^{J_i(z_i)-J_{i+1}(z_i)-J_i(z_{i+1})+J_{i+1}(z_{i+1})} = p_{\text{flip}}(z_i,z_{i+1}), \quad \forall z_i, z_{i+1} \in \mathcal{Z} \tag{26}$$

the probability measure defined by (22) is invariant for the transition probability kernel defined by (25). □

Note that (26) holds for the function p_{flip} in (27) which, for the costs in (23), simplifies to (28). This last formula shows that if $J(z_i) \leqslant J(z_{i+1})$, then the sample z_i that corresponds to the lower cost, but was originally associated with the lower β_i will always transition to the higher β_{i+1}. Conversely, if $J(z_i) > J(z_{i+1})$ the flip will only occur with a reasonable probability if the difference is not very large. This shows the mechanism by which

$$p_{\text{flip}}(z_i, z_{i+1}) = \begin{cases} e^{J_i(z_i) - J_{i+1}(z_i) - J_i(z_{i+1}) + J_{i+1}(z_{i+1})} & J_i(z_i) - J_{i+1}(z_i) - J_i(z_{i+1}) + J_{i+1}(z_{i+1}) \leqslant 0, \\ 1 & J_i(z_i) - J_{i+1}(z_i) - J_i(z_{i+1}) + J_{i+1}(z_{i+1}) > 0, \end{cases} \tag{27}$$

$$p_{\text{flip}}(z_i, z_{i+1}) = \begin{cases} e^{(\beta_{i+1} - \beta_i)(J(z_{i+1}) - J(z_i))} & J(z_{i+1}) - J(z_i) \leqslant 0, \\ 1 & J(z_{i+1}) - J(z_i) > 0. \end{cases} \tag{28}$$

low-cost samples discovered by chains with low values of β will migrate to the chains with high values of β.

D. Adaptive tempering

Tempering is only effective if samples can flow easily between the chains $\boldsymbol{z}_k^{\beta_i}$ and $\boldsymbol{z}_k^{\beta_{i+1}}$, which generally requires the parameters β_i and β_{i+1} to be "sufficiently" close to each other so that the distributions of the different chains are not too far apart. However, it is typically not easy to know in advance how close to each other the β_i should be placed.

To address this challenge, we dynamically adjust all but the last temperature parameters $\beta_1, \ldots, \beta_{M-1}$ to keep the flip probabilities close to a desired value that we denote by $p_{\text{desired}} \in (0, 1)$. The last parameter β_M is kept fixed to make sure that we retain a sufficiently large value β_M, from which we extract the minimum energy/cost samples.

In view of (28), we can get $p_{\text{flip}}(z_i, z_{i+1}) = p_{\text{desired}}$ as long as

$$\beta_{i+1} = \beta_{i+1}^{\text{desired}} \coloneqq \beta_i - \frac{\log(p_{\text{desired}})}{J(z_i) - J(z_{i+1})} \tag{29}$$

or, alternatively,

$$\beta_i = \beta_i^{\text{desired}} \coloneqq \beta_{i+1} + \frac{\log(p_{\text{desired}})}{J(z_i) - J(z_{i+1})}. \tag{30}$$

To achieve this, we update β_{i+1} and the subsequent $\beta_{i+2}, \ldots, \beta_{M-1}$ towards (29) using proportional feedback on the logarithm of the β_i:

$$\log \beta_j \mapsto \log \beta_j + \kappa_P \big(\log \beta_{i+1}^{\text{desired}} - \log \beta_{i+1} \big), \quad \forall j \in \{i+1, \ldots, M-1\}, \tag{31}$$

for some feedback gain $\kappa_P > 0$, as long as this update rule results in a value for β_{M-1} that is still strictly smaller than β_M. Otherwise, we update β_i and the previous $\beta_1, \ldots, \beta_{i-1}$ towards (30) using

$$\log \beta_j \mapsto \log \beta_j + \kappa_P \big(\log \beta_i^{\text{desired}} - \log \beta_i \big), \quad \forall j \in \{1, \ldots, i\}. \tag{32}$$

We make four exceptions to this rule:

1) When $J(z_i) < J(z_{i+1})$, a lower-cost sample is flipped from β_i to β_{i+1} with probability $p_{\text{flip}}(z_i, z_{i+1}) = 1$, which is the desired behavior, and we make no adjustment to any of the temperatures.
2) When $J(z_i) = J(z_{i+1})$, the formulas (29)–(30) are not well defined (division by zero), but we still have $p_{\text{flip}}(z_i, z_{i+1}) = 1 > p_{\text{desired}}$. In this case, instead of (29)–(30), we use
$$\beta_i^{\text{desired}} \coloneqq \beta_i^2 / \beta_{i+1} \tag{33}$$
or, alternatively,
$$\beta_{i+1}^{\text{desired}} \coloneqq \beta_{i+1}^2 / \beta_i; \tag{34}$$
either option would further separate β_i from β_{i+1} by squaring the ratio β_i / β_{i+1}.
3) When (30) would lead to a negative temperature, we use instead (34), which still separates the temperatures, but not enough to actually get the desired flip probability (which would not be possible anyway).
4) No updates are made if (32) would lead to β_1 smallest than a minimum value $\beta_{\min}$ beyond which the corresponding Boltzmann distributions essentially degenerates into a uniform distribution.

Note that these updates rules never change β_M, preserve the order of all the β_i, maintain all the $\beta_i \geqslant \beta_{\min}$, and have a fixed point at $p_{\text{flip}}(z_i, z_{i+1}) = p_{\text{desired}}$, $\forall i$. Moreover, this fixed point is asymptotically stable as long as $\kappa_P > 0$.

V. MCMC for multi-target tracking

We now have all the elements required to define the algorithm used to maximize the MAP criterion (4), or to be more precise, to generate random samples that favor high-values for the a-posteriori probability.

Motivated by parallel tempering, we select a set $\mathcal{B}$ of temperature parameters, as in (20), and our goal is to draw samples from an M-tuple

$$\boldsymbol{z} \coloneqq (\boldsymbol{z}^{\beta_1}, \ldots, \boldsymbol{z}^{\beta_M}) \in \mathcal{Z}^M,$$

with a desired distribution given by (21) for a cost function given by the negative of the MAP criterion (4). Each $\boldsymbol{z}^\beta$ is a pair

$$\boldsymbol{z}^\beta = (\boldsymbol{X}^\beta, \boldsymbol{\sigma}^\beta) \in \mathcal{Z} \coloneqq \mathbb{R}^{NTn_x} \times \mathcal{L}^K$$

with $\boldsymbol{X}^\beta \in \mathbb{R}^{NTn_x}$ associated with the target state sequences and $\boldsymbol{\sigma}^\beta \in \mathcal{L}^K$ with the measurement index sequence. For the purpose of maximizing (4), we only use the samples $\boldsymbol{X}^{\beta_M}$ and $\boldsymbol{\sigma}^{\beta_M}$ associated with the largest value in $\mathcal{B}$.

To generate these samples, we construct a discrete time Markov chain $\boldsymbol{z}[1], \boldsymbol{z}[2], \ldots$, with each $\boldsymbol{z}[m+1]$

sampled from the previous $\boldsymbol{z}[m]$ using a transition kernel $P_m(\cdot, \boldsymbol{z}[m])$, for which (21) is always an invariant distribution. We use 3 types of transition kernels:

1) Gibbs kernels to resample the continuous variables
$$\boldsymbol{X}^\beta \in \mathbb{R}^{NTn_x}, \quad \beta \in \mathcal{B}$$
without changing the discrete variables $\boldsymbol{\sigma}^\beta$. These kernels use the probability density function in (17) to sample each $\boldsymbol{X}^\beta[m+1]$ given $\boldsymbol{\sigma}^\beta[m]$, independently across different values of β.
2) Gibbs kernels to resample each discrete variable
$$\boldsymbol{\sigma}_k^\beta \in \mathcal{L}, \quad \beta \in \mathcal{B},\ k \in \{1, \dots, K\},$$
without changing neither the remaining discrete variables $\boldsymbol{\sigma}_{\bar{k}}^\beta$, $\bar{k} \neq k$ nor the continuous variables in $\boldsymbol{X}^\beta$. These kernels use the probability density function in (18) to sample each $\boldsymbol{\sigma}_k^\beta[m+1]$ given $\boldsymbol{X}^\beta[m]$, $\boldsymbol{\sigma}^\beta[m]$, independently across different values of β.
3) Parallel tempering kernels to flip samples $\boldsymbol{z}^{\beta_i}$ and $\boldsymbol{z}^{\beta_{i+1}}$, $i \in \{1, \dots, M-1\}$ using the flip probability given by (28) to accept a flip from $\boldsymbol{z}[m]$ to $\boldsymbol{z}[m+1]$.

Since all kernels are invariant for (21), they can be combined in any order while preserving invariance. In the algorithm described below, we start with the Gibbs kernels for the $\boldsymbol{X}^\beta$, followed by the tempering kernels, and then the Gibbs kernels for the $\boldsymbol{\sigma}_k^\beta$. This order has the benefit that sampling the Gibbs kernels for the $\boldsymbol{X}^\beta$ provides, as a byproduct, the costs needed for the tempering kernels and thus saves computation. The full algorithm, including the β adjustment described in (31)–(32) for adaptive tempering, is summarized below:

ALGORITHM 1 (MCMC sampling for multi-target tracking) *Start with an arbitrary*

$$\boldsymbol{X}[1] = (\boldsymbol{X}^\beta[1] \in \mathbb{R}^{NTn_x} : \beta \in \mathcal{B}), \quad \boldsymbol{\sigma}[1] = (\boldsymbol{\sigma}^\beta[1] \in \mathcal{L}^K : \beta \in \mathcal{B}),$$

set $m = 1$, and repeat the following steps until enough samples are collected:

1) Gibbs sampling for $\boldsymbol{X}$*: For each $\beta \in \mathcal{B}$:*
   a) *Perform the following Cholesky factorization*
   $$LL' = M_0 + M_1(\boldsymbol{\sigma}^\beta[m]). \tag{35}$$
   b) *Pick a sample $\boldsymbol{\varphi}^\beta \in \mathbb{R}^{NTn_x}$ from a standard Gaussian random vector (with zero mean and covariance matrix equal to the identity).*
   c) *Update the sample*
   $$\boldsymbol{X}^\beta[m+1] = \big(LL'\big)^{-1} b(\boldsymbol{\sigma}^\beta[m]) + \beta^{-\frac{1}{2}} L^{-1} \boldsymbol{\varphi}^\beta \tag{36}$$
   d) *Compute the associated cost*
   $$\begin{aligned} J^\beta[m+1] &\coloneqq J(\boldsymbol{X}^\beta[m+1], \boldsymbol{\sigma}^\beta[m+1])) \\ &= \beta^{-1}\big(L^{-1}\boldsymbol{\varphi}^\beta\big)' LL' \big(L^{-1}\boldsymbol{\varphi}^\beta\big) \\ &\quad - q^*(\boldsymbol{\sigma}^\beta[m]) - c - \log p_\sigma(\boldsymbol{\sigma}^\beta[m]), \end{aligned} \tag{37}$$
   with $M_0, M_1(\boldsymbol{\sigma}^\beta[m]), b(\boldsymbol{\sigma}^\beta[m]), q^*(\boldsymbol{\sigma}^\beta[m])$ given by Proposition 1.
2) Adaptive tempering sweep*: For each $i \in \{1, \dots, M-1\}$*
   a) *Compute the flip probability*
   $$p_{\text{flip}} = \min\Big\{e^{(\beta_{i+1}-\beta_i)(J^{\beta_{i+1}}[m]) - J^{\beta_i}[m])}, 1\Big\}, \tag{38}$$
   and set
   $$(\boldsymbol{X}[m+1], \boldsymbol{\sigma}[m+1]) = \begin{cases} (\tilde{\boldsymbol{X}}[m], \tilde{\boldsymbol{\sigma}}[m]) & \text{with prob. } p_{\text{flip}} \\ (\boldsymbol{X}[m], \boldsymbol{\sigma}[m]) & \text{with prob. } 1 - p_{\text{flip}}, \end{cases}$$
   where $\tilde{\boldsymbol{X}}[m]$ and $\tilde{\boldsymbol{\sigma}}[m]$ are versions of $\boldsymbol{X}[m]$ and $\boldsymbol{\sigma}[m]$ with the entries corresponding to β_i and β_{i+1} flipped.
   b) *If $J^{\beta_{i+1}}[m]) \leqslant J^{\beta_i}[m])$, update the temperature parameters using*
   $$\log \beta_j \mapsto \log \beta_j + \kappa_P\big(\log \beta_{i+1}^{\text{desired}} - \log \beta_{i+1}\big), \quad \forall j \in \{i+1, \dots, M-1\},$$
   if this keeps $\beta_{M-1} < \beta_M$; otherwise use
   $$\log \beta_j \mapsto \log \beta_j + \kappa_P\big(\log \beta_i^{\text{desired}} - \log \beta_i\big), \quad \forall j \in \{1, \dots, i\},$$
   if this keeps $\beta_1 \geqslant \beta_{\min}$; otherwise do not change any of the $\log \beta_j$. The β_i^{desired} are given by
   $$\beta_{i+1}^{\text{desired}} \coloneqq \beta_i - \frac{\log(p_{\text{desired}})}{J^{\beta_i}[m] - J^{\beta_{i+1}}[m])}$$
   $$\beta_i^{\text{desired}} \coloneqq \beta_{i+1} + \frac{\log(p_{\text{desired}})}{J^{\beta_i}[m] - J^{\beta_{i+1}}[m]}$$
   or by (33)–(34), according to the rules in Section D.
   c) *Increment m and proceed to the next value of i.*
3) Gibbs sampling sweep for $\boldsymbol{\sigma}$*: For each $\beta \in \mathcal{B}$, and each each $k \in \{1, \dots, K\}$,*
   a) *Sample $\boldsymbol{\sigma}_k^\beta[m+1]$ with distribution*
   $$p(\bar{\sigma}_k) = \bar{c} \exp\Big(\beta \log p_\sigma(\bar{\sigma}_k, \boldsymbol{\sigma}_{-k}^\beta[m]) + - \frac{\beta}{2}(y_k - C x_{t_k}^{\bar{\sigma}_k})' \Sigma_n^{-1} (y_k - C x_{t_k}^{\bar{\sigma}_k})\Big), \ \forall \bar{\sigma}_k \in \mathcal{L}, \tag{39}$$
   with the target state $x_{t_k}^{\bar{\sigma}_k}$ taken from $\boldsymbol{X}^\beta[m]$.
   b) *Set the remaining $\boldsymbol{\sigma}_{\bar{k}}^\beta[m+1] = \boldsymbol{\sigma}_{\bar{k}}^\beta[m]$, $\forall \bar{k} \neq k$ and $\boldsymbol{X}^\beta[m+1] = \boldsymbol{X}^\beta[m]$,*
   c) *Increment m and proceed to the next value of k.* □

A few remarks about this algorithm are in order:

1) It is sometimes the case that we have little prior information regarding the initial states x_1^ℓ, $\ell \in \mathcal{L}$. In this case, we can omit the term $\sum_{\ell=1}^N E_1^\ell \Sigma_1^{-1} E_1^\ell$ from (7), which would correspond to an "infinitely large" covariance matrix Σ_1 for the prior on these states. Alternatively, we can replace this term by ϵI to make sure that we always apply the Cholesky factorization to a nonsingular matrix.
2) When $p_\sigma(\cdot)$ is a uniform distribution, the term $\log p_\sigma(\boldsymbol{\sigma}^\beta[m])$ in (37) can be removed by absorbing it into the constant c; and the term $\beta \log p_\sigma(\bar{\sigma}_k, \boldsymbol{\sigma}_{-k})$ in (39) is also constant and can be removed by absorbing it into the constant $\bar{c}$.

When, instead, $p_\sigma(\cdot)$ comes from the multinomial distribution (8), (37) becomes

$$\begin{aligned} J^\beta[m+1] &\coloneqq J(\boldsymbol{X}^\beta[m+1], \boldsymbol{\sigma}^\beta[m+1])) \\ &= \beta^{-1}\big(L^{-1}\boldsymbol{\varphi}^\beta\big)' LL' \big(L^{-1}\boldsymbol{\varphi}^\beta\big) - q^*(\boldsymbol{\sigma}^\beta[m]) \\ &- c + \log n_1(\boldsymbol{\sigma}^\beta[m])! + \cdots + \log n_L(\boldsymbol{\sigma}^\beta[m])!, \end{aligned} \tag{40}$$

where we absorbed into the constant c the normalization constant missing from (8); and (39) becomes

$$p(\bar{\sigma}_k) = \bar{c} e^{-\frac{\beta}{2}(y_k - C x_{t_k}^{\bar{\sigma}_k})' \Sigma_n^{-1} (y_k - C x_{t_k}^{\bar{\sigma}_k})} \times \\ \times \begin{cases} 1 & \bar{\sigma}_k = \boldsymbol{\sigma}_k^\beta[m], \\ \left(\frac{n_{\boldsymbol{\sigma}_k^\beta[m]}(\sigma)}{n_{\bar{\sigma}_k}(\sigma)+1}\right)^\beta & \bar{\sigma}_k \neq \boldsymbol{\sigma}_k^\beta[m], \end{cases} \quad \forall \bar{\sigma}_k \in \mathcal{L}. \tag{41}$$

3) The loops over temperatures in steps 1 and 3 can be parallelized since the sampling is independent across different $\beta \in \mathcal{B}$. This provides significant opportunities to speedup sampling in multi-core CPUs or in GPUs, with larger gains obtained when the number of temperatures M is an integer multiple of the number of processor cores.
4) Numerical results indicate that computing the flip probability in (38) with costs defined by

$$J^\beta[m+1] = -q^*(\boldsymbol{\sigma}^\beta[m]) - c - \log p_\sigma(\boldsymbol{\sigma}^\beta[m]). \tag{42}$$

instead of (37) results in faster convergence when the number of temperatures is relatively small. The explanation for this seems to lie in the observation that, when β is not very large, the costs given by (42) tend to be closer to each other, which creates more opportunities for flipping samples. Using (42) instead of (37) violates the invariance condition in (26), but for large values of β (which are the ones we care about) the term $\beta^{-1}\big(L^{-1}\boldsymbol{\varphi}^\beta\big)' LL' \big(L^{-1}\boldsymbol{\varphi}^\beta\big)$ that is missing in (42) makes little difference, so we still have "approximate" invariance for large β.

VI. Numerical results

We now illustrate the performance of the algorithm outlined in Section V in a numerical example for a two-dimensional "constant velocity" model that corresponds to the system in (1)–(3), with

$$A \coloneqq \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C \coloneqq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$\Sigma_D \coloneqq \begin{bmatrix} 10^{-6} & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix}, \; \Sigma_N \coloneqq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

a target state in $\mathbb{R}^4$ consisting of 2 positions stacked on top of 2 velocities; and position measurements in $\mathbb{R}^2$. The top-left submatrix of Σ_D shows uncorrelated stochastic fluctuations in the velocity with standard deviation 10^{-3}, the bottom-right submatrix of Σ_D stochastic fluctuations in the acceleration with standard deviation 10^{-2}, and the matrix Σ_N uncorrelated additive measurements error with unit standard deviation. We assume no prior information on the initial target states and thus replaced $\sum_{\ell=1}^N E_1^\ell \Sigma_1^{-1} E_1^\ell$ in (7) by ϵI, with $\epsilon = 10^{-5}$. We assumed that $p_\sigma(\cdot)$ comes from the multinomial distribution (8) and therefore use (40)–(41) in Algorithm 1. However, the results reported would change very little if, instead, we had assumed no prior information on σ and omitted the terms involving $p_\sigma(\cdot)$ in (37) and (39).

For all results reported in this paper, we used adaptive tempering with $\kappa_P = .02$, $\beta_{\min} = .001$ and $p_{\text{desired}} = .001$ with $M = 24$ temperatures. The initial values for the temperature parameters $\mathcal{B}$ consisted of exponentially scaled values across a 1000-fold range:

$$\beta_1 = 0.1, \quad \beta_2 = \gamma\beta_1, \quad \cdots \quad \beta_M = 100, \quad \gamma \coloneqq 1000^{\frac{1}{M-1}}.$$

With adaptive tempering, these initial values provided good results for all the scenarios discussed in this paper, for which the value of the MAP optimization criteria ranges from the 10s to the 1000s (see values in Figures 2–3). With adaptive tempering, the results also do not seem to be very sensitive to the choice of the number of temperatures M, and similar results were obtained with M as low as 12 and as high as 36 temperatures. For maximum speed, we selected the number of temperatures to be an integer multiple of the number of CPU cores (which was 6 in the computer we used).

To test the algorithm, we generated synthetic data for a time horizon of length $T = 100$ time steps and several values of the number of targets $N \in \{3, 4, 6, 7, 8, 9, 10\}$ and total number of measurements $K \in \{10\%NT, 20\%NT, 50\%NT, 70\%NT, 80\%NT, 90\%NT, 100\%NT\}$. Figure 1 shows plots of the ground truth for a few of these combinations together with the corresponding solutions obtained with Algorithm 1.

To judge the quality of the results, we will compare the value of the MAP criterion achieved by Algorithm 1 with the maximum $q^*(\sigma_{\text{actual}})$ in (5) for the ground-truth sequence σ_{actual} containing the actual targets associated with each measurement. We shall see that the proposed algorithm generally matches and often improves upon these values. We can see an example of this in Figure 1(e), where we see a few mismatches between the ground truth (on the left) and the MAP estimate (on the right). However, Algorithm 1 was actually able to find a solution that has a higher log-likelihood (of -122.00) than the actual solution (with -130.01). This sometimes happens in problems with very few measurements and many targets confined to a small space: in this particular case 10 targets with large overlap in time and space and only 10% measurements, which leads to very high track ambiguity.

Figure 2 shows (the negative) of the "optimal" value for the MAP criterion (4) found by Algorithm 1 for 15 Monte Carlo runs, each running for 1250 iterations. We can see that most of the Monte Carlo runs found values for the MAP criterion that equal or exceed the value $q^*(\sigma_{\text{actual}})$ that would have been obtained with the values of the actual sequence σ_{actual}. For a small number of targets this happens about 10secs or less, but can take as long as 30-40secs for 10 targets. Figure 3 confirms that 1250 iterations suffice to obtain solutions with MAP comparable to ground truth for essentially all the combinations of N and K listed above (only a subset of which appear in Figure 2).

To compare the results obtained with Algorithm 1 with alternative approaches, we ran a simulation study that generated 1000 Monte Carlo datasets constructed using the parameters described above and solved the same multi-target tracking using the JPDA [3] and MCMCDA [10] algorithms. We run these algorithms recursively and then use the Rauch–Tung–Striebel (RTS) smoother [27] to improve the estimates at all the time steps. The JPDAF average and maximal running times were 0.06 s and 0.29 s, respectively, and for MCMCDA the average and maximal running time are 0.19 s and 0.6 s, respectively, using a consumer-grade computer, equipped with a single 6-core Intel i7 CPU running at 2.6 GHz. In each run we randomize the process and measurement noises according to the covariance matrices defined above. We observed that, for the parameters above, both algorithms could not converge in many of the scenarios given just the available a-priori information; especially in the cases with many targets and few measurements. Thus, we provided JPDA and MCMCDA with the true initial states for each one of the targets and we set their initial covariance matrix to be Σ_D, generating a "genie" version of the results. Figure 4 shows (the negative) of the value for the MAP criterion (4) found by the genie versions of these algorithms. We can see that MCMCDA outperforms JPDAF, especially in the challenging cases of many targets and fewer measurements. However, none of these algorithms is able to achieve the results shown in Figure 3 for Algorithm 1, even though this algorithm was only given much looser a-priori information. Figure 5 shows typical estimates obtained by JPDAF and MCMCDA, for the actual paths in Figure 1. Visual inspection confirms the conclusion drawn from Figures 3–4 that a small number of measures typically results in poor estimates for JPDAF and MCMCDA.

VII. Conclusions

We have shown that MCMC sampling can compute solutions to linear multi-target tracking problem with large a-posteriori probabilities. The computational cost of the proposed algorithm is higher than state-of-the-art alternatives, but can produce significantly better solutions when the measurement noise is large and there is significant track confusion.

Promising areas for future research include incorporating known state constraints for the target and considering nonlinear dynamics. Regarding the latter, we note that it is generally not difficult to explicitly write the a-posteriori join distribution of the (continuous) target state X and measurement assignment sequence σ when the target model is nonlinear. However, the maximization of this pdf with respect to X, which has the closed-form solution found in Proposition 1 for linear target dynamics, becomes more difficult for nonlinear targets. Nevertheless, it can often be well approximated by an extended Kalman filter or one of its variants.

While our motivation was multi-target tracking, the tools developed here for MCMC sampling can be used for more general MAP estimation problems. Finding other applications for these tools in another promising area for future research.

Appendix

Proof of Proposition 1:
The definitions of X and the E_t^ℓ enable us to re-write (4) as

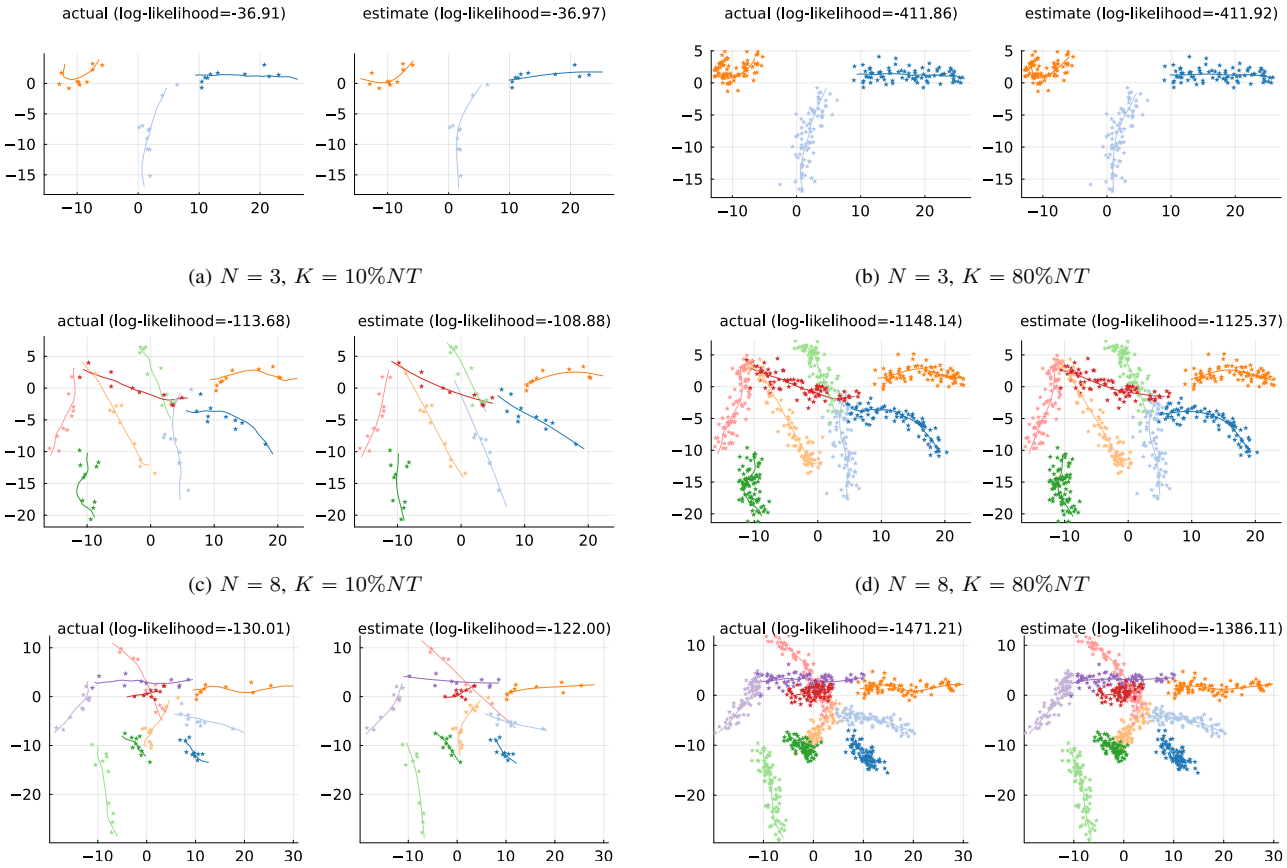
$$
\begin{aligned}
& c + \log p_\sigma(\sigma) - \frac{1}{2}\sum_{\ell=1}^{N} x_1^{\ell\prime}\Sigma_1^{-1}x_1^\ell \\
& \quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T-1}(x_{t+1}^\ell - Ax_t^\ell)'\Sigma_d^{-1}(x_{t+1}^\ell - Ax_t^\ell) \\
& \quad - \frac{1}{2}\sum_{k=1}^{K}(y_k - Cx_{t_k}^{\sigma_k})'\Sigma_n^{-1}(y_k - Cx_{t_k}^{\sigma_k}) \\
& = c + \log p_\sigma(\sigma) - \frac{1}{2}\sum_{\ell=1}^{N} X'E_1^\ell\Sigma_1^{-1}E_1^\ell X \\
& \quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T-1} X'(E_{t+1}^\ell - AE_t^\ell)'\Sigma_d^{-1}(E_{t+1}^\ell - AE_t^\ell)X \\
& \quad - \frac{1}{2}\sum_{k=1}^{K}(y_k - CE_{t_k}^{\sigma_k}X)'\Sigma_n^{-1}(y_k - CE_{t_k}^{\sigma_k}X) \\
& = c + \log p_\sigma(\sigma) - \frac{1}{2}X'\big(M_0 + M_1(\sigma)\big)X \\
& \quad + \frac{1}{2}b(\sigma)'X + \frac{1}{2}X'b(\sigma) - \frac{1}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k.
\end{aligned}
$$

To maximize this function, we set to zero its gradient with respect to X and conclude that the maximum takes place at:

$$
\begin{aligned}
& \big(M_0 + M_1(\sigma)\big)X = b(\sigma) \\
& \qquad\qquad \Leftrightarrow \quad X^*(\sigma) = \big(M_0 + M_1(\sigma)\big)^{-1}b(\sigma)
\end{aligned}
$$

and, at this point, the value of (4) is given by

$$
c + \log p_\sigma(\sigma) - \frac{1}{2}X^*(\sigma)'(M_0 + M_1(\sigma))X^*(\sigma)
$$



(a) $N = 3$, $K = 10\%NT$

(b) $N = 3$, $K = 80\%NT$

(c) $N = 8$, $K = 10\%NT$

(d) $N = 8$, $K = 80\%NT$

(e) $N = 10$, $K = 10\%NT$

(f) $N = 10$, $K = 80\%NT$

Fig. 1: Ground truth and corresponding solutions obtained with Algorithm 1 for different values of the number of targets N and number of measurements K, for a horizon of length $T = 100$ time steps. In every plot, the dots denote the noisy measurements and the solid lines the actual (left plot in each pair) or estimated (right plots in each pair) target positions. Different tracks are represented by different colors.

$$
\begin{aligned}
&+ \frac{1}{2}X^*(\sigma)'b(\sigma) + \frac{1}{2}b(\sigma)'X^*(\sigma) - \frac{1}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k \\
&= c + \log p_\sigma(\sigma) - \frac{1}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k \\
&+ \frac{1}{2}b(\sigma)'\big(M_0 + M_1(\sigma)\big)^{-1}b(\sigma) \\
&= c + \log p_\sigma(\sigma) - q^*(\sigma).
\end{aligned}
$$

■

PROPOSITION 2 *For a cost function $J(X,\sigma)$ given by the negative of (4): The probability density function in (17) corresponds to a multivariable Gaussian distribution with mean $X^*(\sigma)$ and covariance matrix $\beta^{-1}\big(M_0+M_1(\sigma)\big)^{-1}$. A sample for it can be obtained using*

$$\boldsymbol{s} \coloneqq X^*(\sigma) + \beta^{-\frac{1}{2}}L^{-1}\boldsymbol{\varphi},$$

where LL' is a Cholesky factorization of $M_0+M_1(\sigma)$ and $\boldsymbol{\varphi} \in \mathbb{R}^{NTn_x}$ is a standard Gaussian random vector (with zero mean and covariance matrix equal to the identity). For this sample,

$$J(\boldsymbol{s},\sigma) = \beta^{-1}\big(L^{-1}\boldsymbol{\varphi}\big)'LL'\big(L^{-1}\boldsymbol{\varphi}\big)$$

$$- q^*(\sigma) - c - \log p_\sigma(\sigma). \qquad \square$$

Proof of Proposition 2:
We saw in the proof of Proposition 1 that the negative of (4) can be written as

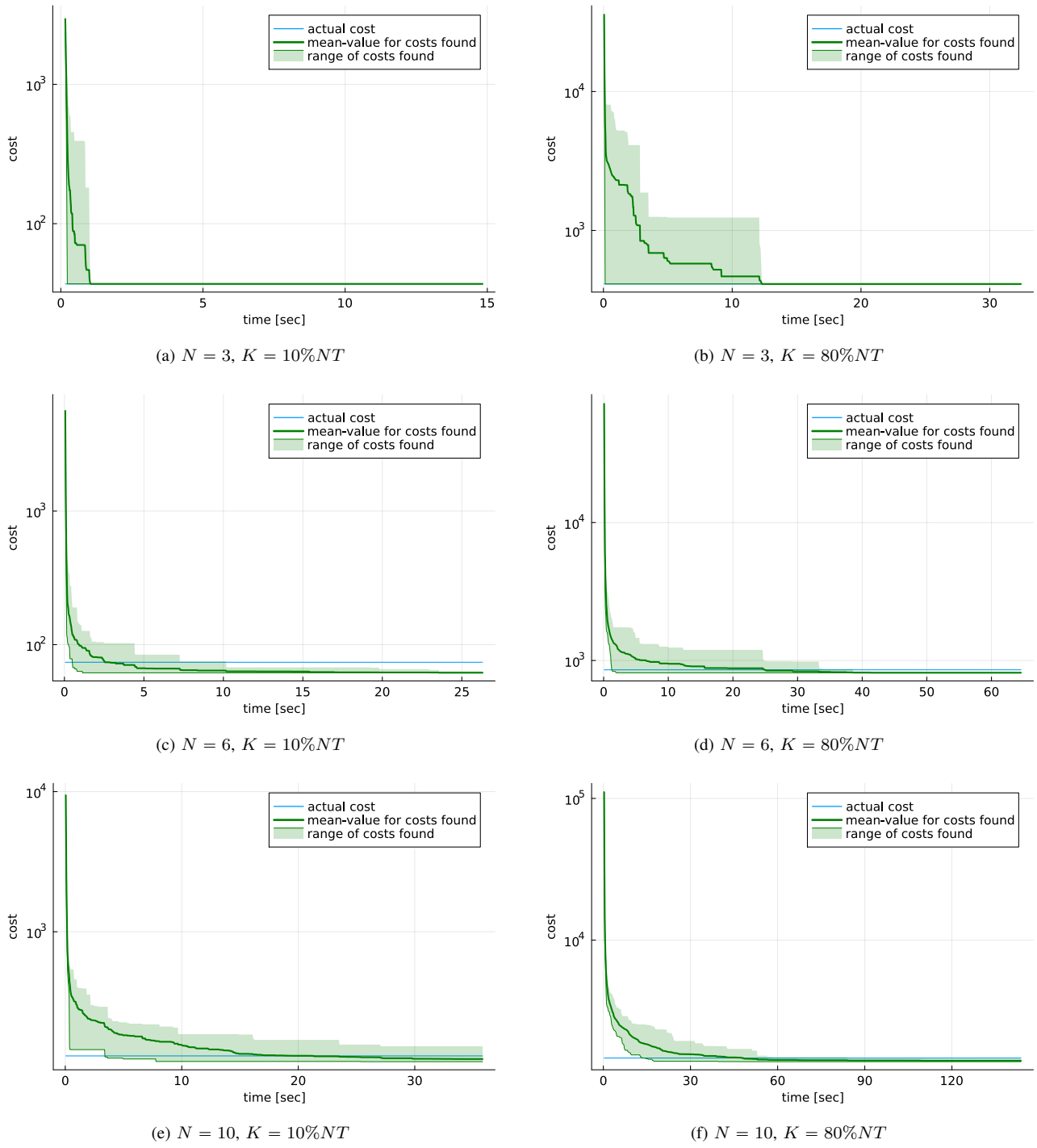
$$
\begin{aligned}
&- c - \log p_\sigma(\sigma) + \frac{1}{2}X'\big(M_0 + M_1(\sigma)\big)X - \frac{1}{2}b(\sigma)'X \\
&- \frac{1}{2}X'b(\sigma) + \frac{1}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k. \qquad (43)
\end{aligned}
$$

We can use the cost in (43) to write the exponent in (17) as

$$
\begin{aligned}
&\beta J(\bar{X},\sigma) = -\beta c - \beta\log p_\sigma(\sigma) - \frac{\beta}{2}b(\sigma)'\bar{X} \\
&- \frac{\beta}{2}\bar{X}'b(\sigma) + \frac{\beta}{2}\bar{X}'\big(M_0 + M_1(\sigma)\big)\bar{X} + \frac{\beta}{2}\sum_{k=1}^{K} y_k'\Sigma_n^{-1}y_k.
\end{aligned}
$$

In view of Proposition 1, this quadratic function is of the form

$$
\begin{aligned}
&\beta J(\bar{X},\sigma) = -\beta q^*(\sigma) - \beta c - \beta\log p_\sigma(\sigma) \\
&+ \beta\big(\bar{X} - X^*(\sigma)\big)'\big(M_0 + M_1(\sigma)\big)\big(\bar{X} - X^*(\sigma)\big)
\end{aligned}
$$



(a) $N = 3$, $K = 10\%NT$

(b) $N = 3$, $K = 80\%NT$

(c) $N = 6$, $K = 10\%NT$

(d) $N = 6$, $K = 80\%NT$

(e) $N = 10$, $K = 10\%NT$

(f) $N = 10$, $K = 80\%NT$

Fig. 2: Cost (i.e., negative of the MAP criterion) found by Algorithm 1 as a function of run time, for different values of the number of targets N and number of measurements K, for a horizon of length $T = 100$ time steps and $M = 24$ temperatures, and $p_{\text{desired}} = .001$. Each plot shows the cost found by the chain with the largest temperature parameter β_M versus run time, for 15 Monte Carlo runs of the algorithm: the solid green lines show the average over 15 runs, whereas the shaded areas show the full range of values obtained over those runs. For comparison, the solid blue line shows the cost $-q^*(\sigma_{\text{actual}})$ obtained for the ground truth assignments σ_{actual} of measurements to targets. The span of the x-axis corresponds to a total of 1250 iterations. All run times refer to a Julia implementation on a 2018 MacBook Pro with a 2.6GHz 6-Core Intel Core i7 CPU.

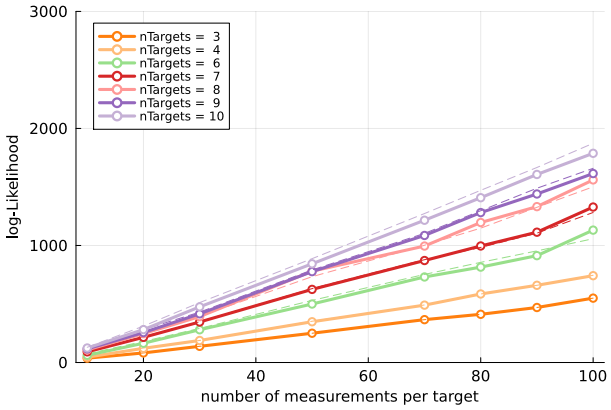


Fig. 3: Cost (i.e., negative of the MAP criterion) found by Algorithm 1 in 1250 iterations, for a range of combinations of the number of targets N and number of measurements K (on the x-axis), for a horizon of length $T = 100$ time steps, $M = 24$ temperatures, and $p_{\text{desired}} = .001$. The solid lines in the plot show an average of the costs obtained over 15 Monte Carlo runs of the algorithm and the dashed lines show the cost $-q^*(\sigma_{\text{actual}})$ obtained for the ground truth assignments σ_{actual} of measurements to targets.

This shows that (17) must be a multivariable Gaussian distribution with mean $X^*(\sigma)$ and covariance matrix $\beta^{-1}\big(M_0 + M_1(\sigma)\big)^{-1}$. Since $\boldsymbol{s}$ is a linear function of a Gaussian random variable it is also Gaussian and since $\boldsymbol{\varphi}$ has zero mean, $\mathrm{E}[\boldsymbol{s}] = X^*(\sigma)$. To verify that $\boldsymbol{s}$ has the desired covariance, we compute

$$\begin{aligned}&\mathrm{E}[(\boldsymbol{s} - X^*(\sigma))(\boldsymbol{s} - X^*(\sigma))']\\ &\quad = \mathrm{E}[\beta^{-1}L^{-1}\boldsymbol{\varphi}\boldsymbol{\varphi}'L'^{-1}]\\ &\quad = \beta^{-1}L^{-1}L'^{-1} = \beta^{-1}(LL')^{-1}.\end{aligned}$$

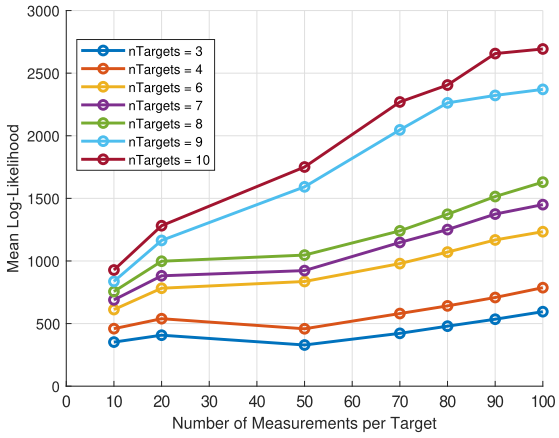
For this sample,

$$\begin{aligned}J(\boldsymbol{s},\sigma) &= \beta^{-1}\big(L^{-1}\boldsymbol{\varphi}\big)'\big(M_0 + M_1(\sigma)\big)\big(L^{-1}\boldsymbol{\varphi}\big)\\ &\quad - q^*(\sigma) - c - \log p_\sigma(\sigma)\\ &= \beta^{-1}\big(L^{-1}\boldsymbol{\varphi}\big)'LL'\big(L^{-1}\boldsymbol{\varphi}\big)\\ &\quad - q^*(\sigma) - c - \log p_\sigma(\sigma).\end{aligned}$$

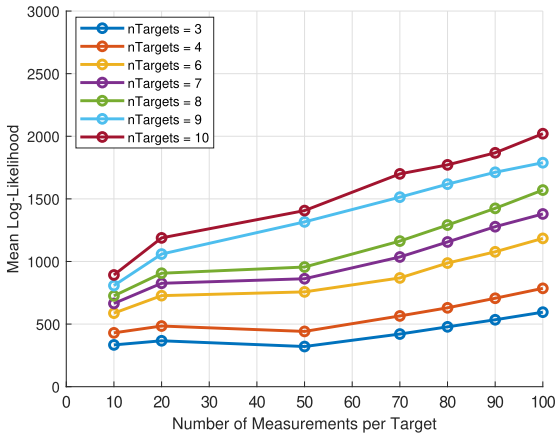
■

Proof of Theorem 2:
Since $z \mapsto e^{-\beta J(z)}$ belong to $L_1(\mu)$, the equation (10) defines a probability measure on $(\mathcal{Z}, \mathcal{F}, \mu)$ that is absolutely continuous with respect to μ and $z \mapsto e^{-\beta J(z)}/Q(\beta)$ is the Radon-Nikodym derivative $\frac{d\pi(\cdot;\beta)}{d\mu}$ of $\pi(\cdot;\beta)$ with respect to μ [29, Theorem 3.8].

Since $z_1 \mapsto e^{-\beta J(z_1,z_2)}$ belongs to $L_1(\mu_1)$, $\forall z_2 \in \mathcal{Z}_2$, we have that $Q_2(\beta, z_2)$ is well-defined and finite $\forall z_2 \in \mathcal{Z}_2$. Consequently, $P(\cdot, z)$ in (16) is a probability measure for every $z \in \mathcal{Z}$. This allow us to further conclude that $P(A, z) \in [0, 1]$, $\forall z \in \mathcal{Z}$, $A \in \mathcal{F}$ and therefore $z \mapsto P(A, z)$ belongs to $L_1(\pi)$. We have thus established that (16) defines a transition probability kernel.



(a) JPDA results



(b) MCMCDA results

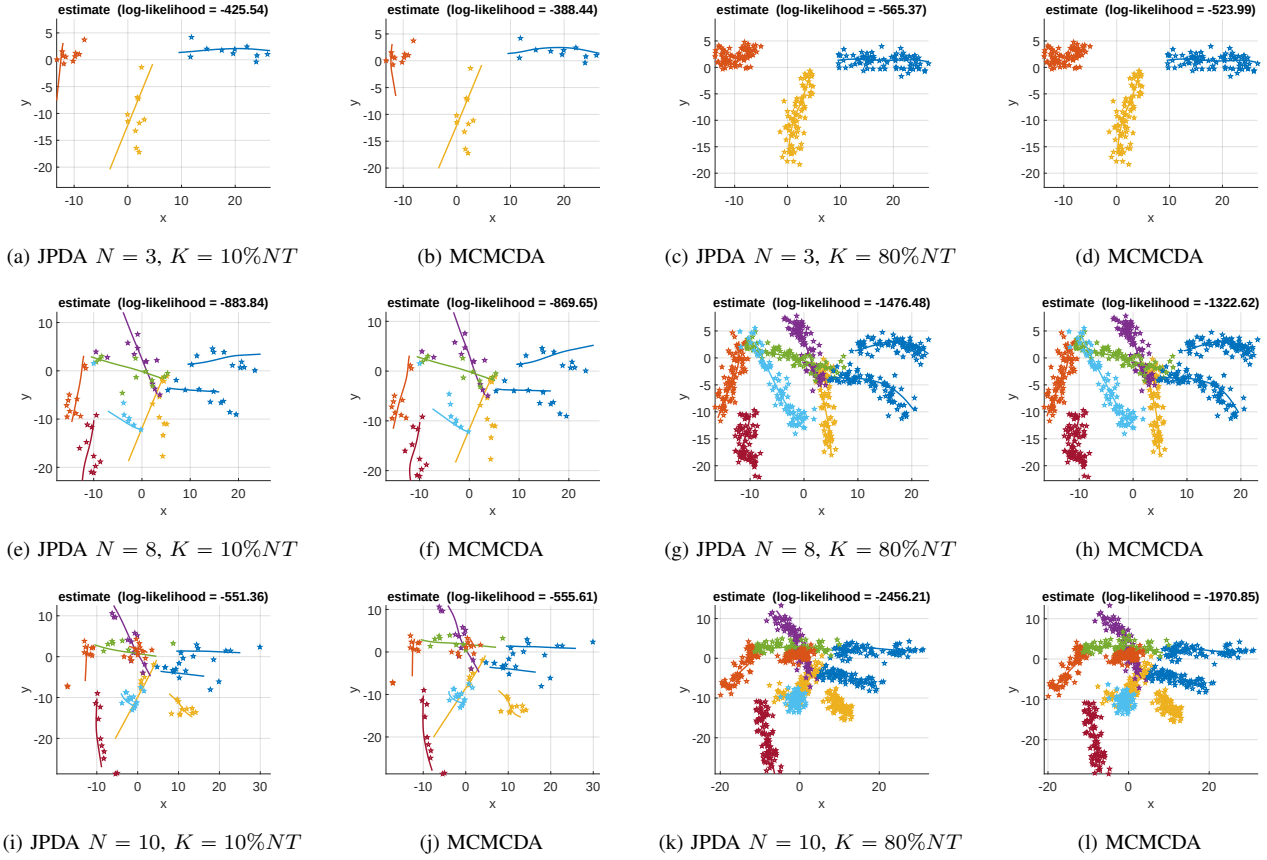
Fig. 4: The mean log-likelihood of Monte Carlo runs using the genie version of the JPDAF and MCMCDA algorithms.

Since $z \mapsto P(A, z)$ belongs to $L_1(\pi)$ and we also conclude that

$$\begin{aligned}\int_{\mathcal{Z}} P(A,z)\pi(dz;\beta) &= \int_{\mathcal{Z}} P(A,z)\frac{d\pi(\cdot;\beta)}{d\mu}\mu(dz)\\ &= \int_{\mathcal{Z}} P(A,z)\frac{e^{-\beta J(z)}}{Q(\beta)}\mu(dz)\end{aligned}$$

[29, Theorem 3.9]. We now use the Fubini-Tonelli Theorem [29, Theorem 2.37] to evaluate the integral above as an iterated integral to conclude that

$$\begin{aligned}&\int_{\mathcal{Z}} P(A,z)\pi(dz;\beta)\\ &= \int_{\mathcal{Z}_1}\int_{\mathcal{Z}_2} P(A,(z_1,z_2))\frac{e^{-\beta J(z_1,z_2)}}{Q(\beta)}\mu_2(dz_2)\mu_1(dz_1)\\ &= \int_{\mathcal{Z}_1}\int_{\mathcal{Z}_2}\Big(\int_{\mathcal{Z}_1}\mathbf{1}_A(\bar{z}_1,z_2)\frac{e^{-\beta J(\bar{z}_1,z_2)}}{Q_2(\beta,z_2)}\mu_1(d\bar{z}_1)\Big)\times\\ &\frac{e^{-\beta J(z_1,z_2)}}{Q(\beta)}\mu_2(dz_2)\mu_1(dz_1)\end{aligned}$$



(a) JPDA $N = 3$, $K = 10\%NT$ (b) MCMCDA (c) JPDA $N = 3$, $K = 80\%NT$ (d) MCMCDA

(e) JPDA $N = 8$, $K = 10\%NT$ (f) MCMCDA (g) JPDA $N = 8$, $K = 80\%NT$ (h) MCMCDA

(i) JPDA $N = 10$, $K = 10\%NT$ (j) MCMCDA (k) JPDA $N = 10$, $K = 80\%NT$ (l) MCMCDA

Fig. 5: Estimates obtained using JPDA and MCMCDA for the paths shown in Figure 1. The dots denote the noisy measurements and the solid lines the estimated target positions. Different tracks are represented by different colors.

$$
\begin{aligned}
&= \int_{\mathcal{Z}_1}\int_{\mathcal{Z}_2} \mathbf{1}_A(\bar{z}_1, z_2)\frac{e^{-\beta J(\bar{z}_1,z_2)}}{Q_2(\beta, z_2)}\times \\
&\frac{\int_{\mathcal{Z}_1} e^{-\beta J(z_1,z_2)}\mu_1(dz_1)}{Q(\beta)}\mu_2(dz_2)\mu_1(d\bar{z}_1) \\
&= \int_{\mathcal{Z}_1}\int_{\mathcal{Z}_2} \mathbf{1}_A(\bar{z}_1, z_2)\frac{e^{-\beta J(\bar{z}_1,z_2)}}{Q_2(\beta, z_2)}\times \\
&\frac{Q_2(\beta, z_2)}{Q(\beta)}\mu_2(dz_2)\mu_1(d\bar{z}_1) \\
&= \int_{\mathcal{Z}_1}\int_{\mathcal{Z}_2} \mathbf{1}_A(\bar{z}_1, z_2)\frac{e^{-\beta J(\bar{z}_1,z_2)}}{Q(\beta)}\mu_2(dz_2)\mu_1(d\bar{z}_1) \\
&= \int_{\mathcal{Z}} \mathbf{1}_A(z)\frac{e^{-\beta J(z)}}{Q(\beta)}\mu(dz) = \pi(A;\beta),
\end{aligned}
$$

which establishes the invariance of $\pi(A;\beta)$. ∎

Proof of Theorem 3:
Defining

$$A_{\text{flip}} \coloneqq \{(z_i, z_{i+1}) \in \mathcal{Z}\times\mathcal{Z} : (z_{i+1}, z_i) \in A\}, \quad \forall A \in \mathcal{F},$$

we can re-write (25) as

$$
\begin{aligned}
P^{\text{flip}}(A,(z_i, z_{i+1})) &= (1 - p_{\text{flip}}(z_i, z_{i+1}))\mathbf{1}_A(z_i, z_{i+1}) \\
&+ p_{\text{flip}}(z_i, z_{i+1})\mathbf{1}_{A_{\text{flip}}}(z_i, z_{i+1}).
\end{aligned} \tag{44}
$$

We recall that $\mathcal{F}\otimes\mathcal{F}$ is the σ-algebra generated by

$$\{A_i \times A_{i+1} : A_i, A_{i+1} \in \mathcal{F}\}$$

[29, Proposition 1.3] so, by symmetry,

$$A \in \mathcal{F}\otimes\mathcal{F} \quad \Rightarrow \quad A_{\text{flip}} \in \mathcal{F}\otimes\mathcal{F},$$

which means that $P^{\text{flip}}(\cdot,(z_i, z_{i+1}))$ defined by (44) is a discrete probability measure for every $z_i, z_{i+1} \in \mathcal{Z}$ with point masses at (z_i, z_{i+1}) and (z_{i+1}, z_i). Since we can also re-write (44) as

$$
\begin{aligned}
P^{\text{flip}}(A, z) &= \min\{1 - p_{\text{flip}}(z), \mathbf{1}_A(z)\} \\
&+ \min\{p_{\text{flip}}(z), \mathbf{1}_{A_{\text{flip}}}(z)\}
\end{aligned}
$$

we conclude that the function $z \mapsto P^{\text{flip}}(A, z)$ is measurable because it can be obtained using only sum and minimum operations on measurable functions [29, Propositions 2.6 and 2.7]. This enables us to conclude that (44) defines a transition probability kernel.

Since $z \mapsto e^{-\beta J(z)}$ belong to $L_1(\mu)$, the equation (22) defines a probability measure on $(\mathcal{Z}\times\mathcal{Z}, \mathcal{F}\otimes\mathcal{F}, \mu\times\mu)$ that is absolutely continuous with respect to $\mu\times\mu$ and $z \mapsto e^{-J_i(z_i)}e^{-J_{i+1}(z_{i+1})}/Q(J_i)Q(J_{i+1})$ is the Radon-Nikodym derivative $\frac{d\pi(\cdot;J_i,J_{i+1})}{d\mu\times\mu}$ of $\pi(\cdot;J_i,J_{i+1})$ with respect to $\mu\times\mu$ [29, Theorem 3.8].

Since $z \mapsto P^{\text{flip}}(A, z)$ belongs to $L_1(\pi)$ and we also conclude that

$$\int_{\mathcal{Z}\times\mathcal{Z}} P^{\text{flip}}(A, z)\pi(dz; J_i, J_{i+1})$$

$$= \int_{\mathcal{Z}\times\mathcal{Z}} P^{\text{flip}}(A,z)\frac{d\pi(\cdot;J_i,J_{i+1})}{d\mu\times\mu}(\mu\times\mu)(dz)$$
$$= \int_{\mathcal{Z}\times\mathcal{Z}} P^{\text{flip}}(A,z)\frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}(\mu\times\mu)(dz)$$

[29, Theorem 3.9]. We now use the Fubinni-Tonelli Theorem [29, Theorem 2.37] to evaluate the integral above as an iterated integral and (25) to conclude that

$$\begin{aligned}
&\int_{\mathcal{Z}\times\mathcal{Z}} P^{\text{flip}}(A,z)\pi(dz;J_i,J_{i+1})\\
&= \int_{\mathcal{Z}}\int_{\mathcal{Z}} P^{\text{flip}}(A,z)\frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&= \int_{\mathcal{Z}}\int_{\mathcal{Z}} \Big((1-p_{\text{flip}}(z_i,z_{i+1}))\mathbf{1}_A(z_i,z_{i+1})\\
&\quad + p_{\text{flip}}(z_i,z_{i+1})\mathbf{1}_A(z_{i+1},z_i)\Big)\times\\
&\quad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&= \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})\frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&\quad + \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_{i+1},z_i)p_{\text{flip}}(z_i,z_{i+1})\times\\
&\quad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&\quad - \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})p_{\text{flip}}(z_i,z_{i+1})\times\\
&\quad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&= \pi(A;J_i,J_{i+1})\\
&\quad + \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})p_{\text{flip}}(z_{i+1},z_i)\times\\
&\quad \frac{e^{-J_i(z_{i+1})}}{Q(J_i)}\frac{e^{-J_{i+1}(z_i)}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i)\\
&\quad - \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})p_{\text{flip}}(z_i,z_{i+1})\times\\
&\quad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i),\\
&= \pi(A;J_i,J_{i+1})\\
&\quad + \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})\Big(p_{\text{flip}}(z_{i+1},z_i)\times\\
&\quad e^{J_i(z_i)-J_{i+1}(z_i)-J_i(z_{i+1})+J_{i+1}(z_{i+1})} - p_{\text{flip}}(z_i,z_{i+1})\Big)\times\\
&\qquad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i),\\
&= \pi(A;J_i,J_{i+1}) + \int_{\mathcal{Z}}\int_{\mathcal{Z}} \mathbf{1}_A(z_i,z_{i+1})\Big(p_{\text{flip}}(z_{i+1},z_i)\times\\
&\quad e^{(\beta_{i+1}-\beta_i)(J(z_{i+1})-J(z_i))} - p_{\text{flip}}(z_i,z_{i+1})\Big)\times\\
&\quad \frac{e^{-J_i(z_i)}}{Q(J_i)}\frac{e^{-J_{i+1}(z_{i+1})}}{Q(J_{i+1})}\mu(dz_{i+1})\mu(dz_i),
\end{aligned}$$

where in the second to last equality we simply used (22) and changed the order of integration variables in one of the integrals. When (26) holds the integral is zero and we conclude that $\pi(\cdot;J_i,J_{i+1})$ is invariant. ■

REFERENCES

[1] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe
Multi-target tracking using joint probabilistic data association
In *Proc. of the 19th IEEE Conf. on Decision and Contr. Symp. on Adaptive Processes*, 1980, pp. 807–812.

[2] B. Vo *et al.*
Multitarget tracking
Wiley encyclopedia of electrical and electronics engineering, 2015.

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

[4] J. A. Roecker
A class of near optimal JPDA algorithms
IEEE Trans. Aerospace and Electronic Syst., vol. 30, no. 2, pp. 504–510, 1994.

[5] S. Maskell, M. Briers, and R. Wright
Fast mutual exclusion
In *Signal and Data Processing of Small Targets 2004*, vol. 5428. SPIE, 2004, pp. 526–536.

[6] P. Horridge and S. Maskell
Real-time tracking of hundreds of targets with efficient exact JPDAF implementation
In *2006 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–8.

[7] K. Romeo, D. F. Crouse, Y. Bar-Shalom, and P. Willett
The JPDAF in practical systems: Approximations
In *Signal and Data Processing of Small Targets 2010*, vol. 7698. SPIE, 2010, pp. 594–603.

[8] M. Baum, P. Willett, Y. Bar-Shalom, and U. D. Hanebeck
Approximate calculation of marginal association probabilities using a hybrid data association model
In *Signal and Data Processing of Small Targets 2014*, vol. 9092. SPIE, 2014, pp. 187–194.

[9] J. Williams and R. Lau
Approximate evaluation of marginal association probabilities with belief propagation
IEEE Trans. Aerospace and Electronic Syst., vol. 50, no. 4, pp. 2942–2959, 2014.

[10] S. Oh, S. Russell, and S. Sastry
Markov chain Monte Carlo data association for multi-target tracking
IEEE Transactions on Automatic Control, vol. 54, no. 3, pp. 481–497, 2009.

[11] D. Reid
An algorithm for tracking multiple targets
IEEE Trans. on Automat. Contr., vol. 24, no. 6, pp. 843–854, 1979.

[12] T. Kurien
Multitarget-Multisensor Tracking: Advanced Applications. Artech House, 1990, ch. Issues in the design of practical multitarget tracking algorithms, pp. 43–84.

[13] S. S. Blackman
Multiple hypothesis tracking for multiple target tracking
IEEE Aerospace and Electronic Systems Magazine, vol. 19, no. 1, pp. 5–18, 2004.

[14] M. Mallick, S. Coraluppi, and C. Carthel
Multitarget Tracking Using Multiple Hypothesis Tracking. John Wiley & Sons, Ltd, 2016, ch. 5, pp. 163–203. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118450550.ch05

[15] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi
Optimization by simulated annealing
Science, vol. 220, no. 4598, pp. 671–680, 1983.

[16] V. Černỳ
Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm
J. Opt. Theory and Applications, vol. 45, pp. 41–51, 1985.

[17] N. Mohseni, P. L. McMahon, and T. Byrnes
Ising machines as hardware solvers of combinatorial optimization problems
Nature Reviews Physics, vol. 4, no. 6, pp. 363–379, 2022.

[18] N. Aadit *et al.*
Massively parallel probabilistic computing with sparse Ising machines
Nature Electronics, vol. 5, no. 7, pp. 460–468, 2022.

[19] J. P. Hespanha and K. Çamsarı
Markov chain Monte Carlo for Koopman-based optimal control
IEEE Control Systems Lett., pp. 1901–1906, Jun. 2024.

[20] S. Geman and D. Geman
Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images
IEEE Trans. on Pattern Anal. and Machine Intell., vol. PAMI-6, no. 6, pp. 721–741, 1984.

[21] A. E. Gelfand and A. F. Smith
Sampling-based approaches to calculating marginal densities
J. of the American Statistical Assoc., vol. 85, no. 410, pp. 398–409, 1990.

[22] C. J. Geyer
Markov chain Monte Carlo maximum likelihood
In *Computing Science and Statistics: Proc. of the 23rd Symp. on the Interface*, 1991, pp. 156–163.

[23] K. Hukushima and K. Nemoto
Exchange Monte Carlo method and application to spin glass simulations
J. of the Physical Society of Japan, vol. 65, no. 6, pp. 1604–1608, 1996.

[24] N. Rathore, M. Chopra, and J. J. de Pablo
Optimal allocation of replicas in parallel tempering simulations
The Journal of Chemical Physics, vol. 122, no. 2, p. 024111, 12 2004. [Online]. Available: https://doi.org/10.1063/1.1831273

[25] A. Schug, T. Herges, and W. Wenzel
All-atom folding of the three-helix hiv accessory protein with an adaptive parallel tempering method
Proteins: Structure, Function, and Bioinformatics, vol. 57, no. 4, pp. 792–798, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.20290

[26] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer
Feedback-optimized parallel tempering monte carlo
Journal of Statistical Mechanics: Theory and Experiment, vol. 2006, no. 03, p. P03018, mar 2006. [Online]. Available: https://dx.doi.org/10.1088/1742-5468/2006/03/P03018

[27] H. E. Rauch, F. Tung, and C. T. Striebel
Maximum likelihood estimates of linear dynamic systems
AIAA Journal, vol. 3, no. 8, pp. 1445–1450, 1965.

[28] J. P. Hespanha, R. Chinchilla, R. R. Costa, M. K. Erdal, and G. Yang
Forecasting COVID-19 cases based on a parameter-varying stochastic SIR model
Annual Control Reviews, Pandemic Special Issue, vol. 51, pp. 460–476, 2021.

[29] G. B. Folland
Real Analysis: Modern Techniques and Their Applications, ser. Pure and Applied Mathematics. New York: John Wiley & Sons, 1984.

[30] S. P. Meyn and R. L. Tweedie
Markov Chains and Stochastic Stability, 2nd ed. New York, USA: Cambridge University Press, 2009.

[31] L. Tierney
Markov chains for exploring posterior distributions
Annals of Statistics, vol. 22, no. 4, pp. 1701–1762, 1994.