Markov Chain Monte Carlo Multi-target Tracking

John Lambert* and Brian Jackson†

I. Motivation

State estimation of a single object from noisy measurements is a very well-studied problem and has been the principal area of focus for the current class. However, in many realistic applications there exists more than one object whose state should be tracked. Under some circumstances, measurements of the targets can be unambiguously associated with their target origin, in which case the issue of tracking multiple targets decomposes into several simultaneous and independent single-object state estimation problems which can be tackled using the algorithms discussed in class. Unfortunately, many scenarios exist wherein the association between measurements and their sources is ambiguous. This has given rise to the field of multiple target tracking (MTT). MTT algorithms have many important, modern applications in areas such as surveillance, defense, and mobile robot navigation. Our motivation for this project is to expand the principles taught in class to more complex scenarios where the issue of data association must be addressed.

The structure of the paper is as follows: (TODO)

II. Objective

The objective of this project is to implement one or more forms of the multiple hypothesis tracking (MHT) algorithm on a simulated scene of mobile robots.

III. Related Work

Many approaches to solving the challenging MTT problem have been investigated throughout the years. Classical approaches decompose the problem into two different problems: data association and state estimation. Assuming that the data association is correct, estimating the state using Kalman [2] filters and its derivatives is straightforward. At the intersection of these two problems, however, lies the heart of the challenge of MTT, for each stage relies upon the other. It is desirable that the problem of data association be informed by accurate state estimation and reason about the effect of data association on estimating the state over time, since and incorrect data association at one time step may have significant impact on estimation of the state many time steps later.

MTT algorithms must tackle at least five key challenges. First, the surveillance region will contain an unknown number of targets because targets die as they exit and are suddenly born as they enter the surveillance region. Next, one most model the interaction of targets because they will certainly interact and rules of physics govern collisions and the simultaneous occupation of free space. Third, the dynamics models of each target must be accounted for (often known as *system identification*) because automobiles and humans will move quite differently through a surveillance region. The last two challenges derive from measurement ambiguity: unknown data association, as we remarked earlier, and potentially false sensor readings that could lead to spurious detections(*false alarms*).

A goal of MTT is to partition sensor data into sets of observations, “tracks,” that are produced by the same target. Once tracks are formed and confirmed, quantities of interest, including target velocity, future predicted position, and target classification, can be computed for each track.

A. Multiple Hypothesis Tracking

The key principle of multiple hypothesis tracking (MHT) is that difficult data association decisions should be deferred until more data are received. Gating is a common method for eliminating observations from consideration to all previous tracks by selecting a maximum acceptable measurement plus tracking error magnitude [8].

B. Classical MHT

MHT algorithms date back to Reid’s original 1979 paper [5]. Reid proposes a recursive algorithm that addressed the dependence of data association on subsequent data by formulating the problem in terms of joint hypotheses, for hypothesis that a single measurement corresponds to a particular target. By evaluating the tree of branching hypotheses, Reid arrives at a recursive algorithm that allows data association based on subsequent, as well as previous, measurements, known as a*multi-scan* approach. On the other hand, a *single-scan* method can utilize only the current set of measurements. Reid’s paper has become the foundational paper for the MHT approach to MTT.

Another approach to MTT is referred to as the Joint Probabilistic Data Association (JPDA) method, as proposed by Bar-Shalom in 1975 [4]. In this approach, each measurement is assigned a probability of association with each target. This increases the Kalman filter covariance to account for association uncertainty. However, this tends to exacerbate the problem since an increased covariance allows even more measurements to be associated with a particular target. It also tends to coalesce closely spaced targets into a single target [8].

Many of the papers over the subsequent decades focused on refining these approaches. Blackman’s survey paper provides a broad overview of different approaches and focuses on highlighting the benefits of the MHT approach originally proposed by Reid, and proposes combining the MHT method with multiple filter models, such as the interacting multiple

*Department of Computer Science, Stanford University
†Department of Mechanical Engineering, Stanford University

model (IMM) [8]. The MHT relies upon generating new, feasible hypotheses from the previous hypotheses. Of course, there is a potential combination explosion in the number of hypotheses (and tracks within hypotheses), so clustering, hypothesis and track pruning, and track merging are required.

C. Optimization-based MHT

Tracking can also be framed as an optimization problem with constraints, solved via Lagrangian relaxation [8]. The objective function of interest to be maximized is the hypothesis score, i.e. the sum of all track scores in a hypothesis. Constraints are needed in order to ensure that no tracks in the hypothesis share the same observations, or, in other words, that an observation can be used by at most a single track. Lagrange multipliers are used to approximate the constraints, ensuring that constraint violations are given high costs.

D. Sequential Monte Carlo

The Probability Hypothesis Density (PHD) filter propagates only the first moment (or PHD) instead of the full multi-target posterior. Vo *et al.* [7] propose Sequential Monte Carlo (SMC) implementations for both the Bayes multi-target filter and the Probability Hypothesis Density filter. They do so in the context of Finite Set Statistics (FISST), where random finite sets represent multi-target states and observations, in order to use a measure theoretic and random finite set framework.

E. Bayesian MHT

Other approaches focus on formulating the problem within a Bayesian framework. Mori et. al. derived a Bayesian approach for cases where targets do not have *a priori* identification [6]. In more recent years emphasis has shifted to using numerical techniques for estimating the joint multitarget probability density (JMPD), primarily using particle filters, where each particle represents a single joint hypothesis as set forth in the MHT formulation [10], [11]. Stone et. al. have compiled many of these approaches within a standardized mathematical approach in their book, which will be serving as a primary reference for this project [13].

Khan *et al.* [9] use a Markov Chain Monte Carlo (MCMC) framework with the Metropolis-Hastings and an MRF motion prior. Oh *et al.* [12] present a single-scan MCMCDA algorithm that approximates JPDA in polynomial time, along with a multi-scan MCMC-DA Bayesian algorithm that can handle detection failures, false alarms, and track initiation and termination. Kim *et al.* [14] use modern object detectors built with convolutional neural networks to provide input to their multi-hypothesis tracker, focusing on an appearance model, rather than motion model.

For our project, we base our approach off of the MCMC-DA Bayesian algorithm by Oh *et. al.* [12].

IV. Problem Formulation

A. General Formulation of MTT

The general MTT problem can be formulated as follows: let $T \in \mathbb{Z}^+$ be the duration of surveillance, and K be the unknown number of targets to track within a surveillance region $\mathcal{R}$ with volume V.. Each object k moves in $\mathcal{R}$ for an unknown duration $[t_i^k, t_f^k] \subseteq [1, T]$. Each target k disappears with probability p_z and is detected with probability p_d at each time step. The apparition of new targets is defined by a Poisson distribution with parameter $\lambda_b V$, where λ_b is the birth rate of new targets per unit time, per unit volume. False alarms are also generated with a Poisson distribution, with parameter λ_f.

Each target moves with dynamics $x_{t+1}^k = F^k(x_t^k)$ and reports observations according to $y_t^j = H^j(x_t^k)$, where $y_t^j \in \mathbb{R}^{n_y}$ is the jth observation at time t for $j = 1, \dots, n_t$, for n_t observations of dimension n_y.

The multi-target tracking problem is then to estimate K, $\{t_i^k, t_f^k\}$, and $\{x_t^k : t_i^k \leq t \leq t_f^k\}$ for $k = 1, \dots, K$ from observations.

B. Current Problem Formulation

For this project we will consider a simplification of the general problem given in Section IV-A. We will assume the number of targets K is known, and assume knowledge of controls and dynamics of each robot. All robots report one measurement at each time, with no spurious detections. Robots do not leave or enter the observation region R during the observation time T. This basically reduces the problem to the issue of data association, which is the main focus of this project. The observation window is set a size of $x \in [-10, 10], y \in [-10, 10]$.

The dynamics of each robot will be modeled as differential drive robots with Gaussian white noise with covariance Q. The controls are assumed, in general, to be time-varying.

The measurement model consists of two range measurements from beacons at $(-10, -10)$ and $(10, -10)$, with Gaussian white noise with covariance R. See Figure 1 for an image from the simulator.

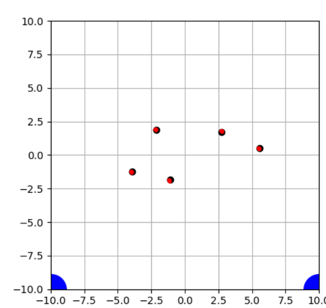


Fig. 1. Still image capture of "arena" simulator. The black and red dots represent the true and estimated location of each robot, respectively. The large blue markers in the corners represent the location of the beacons.

C. Optimal Bayesian Filter Formulation

The problem stated previously can be phrased within a Bayesian framework, similar to what we have seen in class. Let $X_t = (X_t^1, \dots, X_t^K)$ be the joint state at time t. We assume we have the prior distribution $P(X_0)$.

Prediction Step: The prediction step can be written

$$\begin{aligned} P(X_t|y_{1:t-1}) &= \int P(X_t|X_{t-1}, y_{1:t-1})P(X_{t-1}|y_{1:t-1})dX_{t-1} \\ &= \int P(X_t|X_{t-1})P(X_{t-1}|y_{1:t-1})dX_{t-1} \end{aligned} \tag{1}$$

where $P(X_t|X_{t-1})$ is the dynamics model, and the step from the first to second line is given by the Markoviantiy of the problem formulation.

Measurement Update: The measurement update is given using an application of Bayes rule:

$$P(X_t|y_{1:t}) = \frac{P(y_t|X_t, y_{1:t-1})P(X_t|y_{1:t-1})}{\int P(y_t|X_t, y_{1:t-1})P(X_t|y_{1:t-1})dx_t} \tag{2}$$

however, for the multi-target tracking problem the data association is unknown, which means the measurement likelihood $P(y_t|X_t, y_{1:t-1})$ is not available. To compensate for this, we introduce a latent variable $\omega_t \in \Omega_t$, which represents a possible set of associations between n_t measurements and K targets, and Ω_t represents the set of all possible associations. Using the theorem of total probability, we can then compute the measurement likelihood as follows:

$$P(X_t|y_{1:t}) = \sum_{\omega_t \in \Omega} P(X_t|\omega_t, y_{1:t})P(\omega_t|y_{1:t}) \tag{3}$$

where

$$P(X_t|\omega_t, y_{1:t}) = \frac{P(y_t|X_t, \omega_t, y_{1:t-1})P(X_t|y_{1:t-1})}{\int P(y_t|X_t, \omega_t, y_{1:t-1})P(X_t|y_{1:t-1})dX_t} \tag{4}$$

The term $P(X_t|\omega_t, y_{1:t})$ can easily computed since ω specifies the association between measurements and targets. However, the sum over all $\omega_t \in \Omega_t$ has exponentially terms and quickly becomes intractable. Additionally, this causes the posterior $P(X_t|y_{1:t})$ to become exponentially more complex after each update step. The two filters presented in the next section are approximations of the optimal Bayes filter presented in this section.

V. Methodology

Two different separate filters were implemented to tackle the multi-target tracking problem. The first was a Multiple-Hypothesis Kalman-Filter (MHKF), similar to the approach taken in one of the problem sets. The second was the MCMC-DA filter proposed in [12]. An overview of the implementation of each filter is given below.

A. MHKF

The key innovation behind the MHKF is that rather than keeping a Gaussian state object for each target, the states of all K targets are concatenated into a single Gaussian object comprising a single distribution. In practice, MHKF algorithms preserve a set of N_t Gaussian objects, where each Gaussian object represents a possible hypothesis regarding associations (matching measurements with targets to form tracks). A new set of Gaussian objects is created at each time step to demonstrate all possible new permutations of associations, and in order to prevent $O(K!)$ branching, this set is quickly pruned to the most probable hypotheses (Gaussian objects).

The MHKF can be seen as employing a Gaussian-Mixture-Model (GMM) in order to estimate the joint state X_t. The estimate at each time step is represented by the GMM:

$$p(x_t|y_{1:t}) = \sum \alpha^i_{t|t} \mathcal{N}(\mu^i_{t|t}, \Sigma^i_{t|t}) \tag{5}$$

Prediction Step: For the general form of MHKF, we assume M transition different transition models. Summing over these transition models yields the following:

$$p(x_{t+1} \mid y_{1:t}) = \sum_j^M \sum_i^{N_t} \beta^j \alpha^j_{t|t} \mathcal{N}(\mu^{ij}_{t+1|t}, \Sigma^{ij}_{t+1|t}) \tag{6}$$

where $\mathcal{N}(\mu^{ij}_{t+1|t}, \Sigma^{ij}_{t+1|t})$ is the Kalman Filter (KF) prediction step. For our simulation, the dynamics remained the same, but we assumed we didn't know the association of the control signal to each target. Therefore, $M = K$.

Measurement Update: For the measurement update, we assume n_y measurements are received, again with unknown association to targets. The prediction step is given as:

$$P(X_{t+1}|y_{t:t+1}) = \sum_i^L \sum_k^{MN_t} \eta\gamma^i \alpha^k_{t+1|t} \mathcal{N}(\mu^{ik}_{t+1|t+1}, \Sigma^{ik}_{t+1|t+1}) \tag{7}$$

where $\mathcal{N}(\mu^{ik}_{t+1|t+1}, \Sigma^{ik}_{t+1|t+1})$ is the KF measurement update, and

$$\begin{aligned} \bar{\alpha}_{t+1|t+1} &= \eta\gamma_i \alpha^k_{t+1|t} \\ &= \gamma_i \alpha^k_{t+1|t} p^{ik}(y_{t+1}|y_{1:t}) \end{aligned} \tag{8}$$

and the score is effectively the probability of seeing the new measurement given your previous measurements, computed using the Gaussian probabiility density function, centered at your dynamics-propagated prediction of the mean, which we call

$$p^{ik}(y_{t+1} \mid y_{1:t}) = \mathcal{N}(C^i_{t+1}\mu^k_{t+1|t}), R^i + C^i_{t+1}\Sigma_{t+1|t}(C^i_{t+1})^T) \tag{9}$$

the measurement likelihood posterior. This is the state after dynamics propagation, meaning $\mu^k_{t+1|t} = g(\mu^k_t)$, and C^i_{t+1} is the i^{th} permutation matrix.

After each time step, the number of Gaussian components expands in a combinatorial manner. To simplify the problem, we take the N_t Gaussians with the highest weights and prune the rest. In our formulation, we chose $N_t = 3$ hypotheses, and we note that it is preferable to keep as hypotheses many as possible given your available time and memory resource budget because this will render the approximation closer to the optimal Bayesian filter.

B. Markov Chain Monte Carlo Data Association (MCMC-DA)

One of the principle issues with the MHKF filter is that is tracks the joint state X_t, meaning a single state which represents the concenated states of all K objects in the surveillance region, which clearly grows in size proportional to the number of targets tracked. The filter operations, especially matrix inversions, begin to take a heavy toll on computation time as the number of targets grows, resulting in an explosion in computation time (as is shown in Section VI). In order to alleviate this issue, the MCMC-DA algorithm makes a key independence assumption:

$$P(X_t|y_{1:t}) = P(X_t^1, \dots, X_t^K|y_{1:t}) \approx \prod_{k=1}^{K} P(X_t^k|y_{1:t}) \quad (10)$$

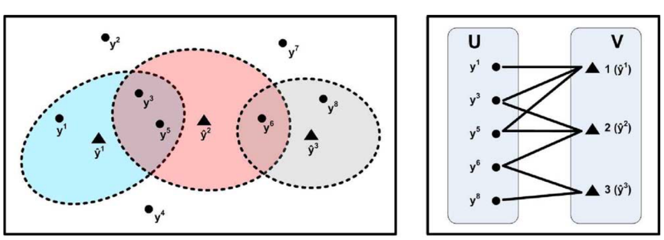
This assumption allows the joint state to be approximated by K independent filters. However, the exponential complexity in estimating the association between measurements and targets still applies. However, if it can be accurately estimated, the tracking problem reduces to a straight-forward filtering problem with K independent filters. The focus of this algorithm, as will be shown, is therefore to accurately and efficiently estimate the data association.

One of the convenient aspects of the MCMC-DA algorithm is that it can work with any type of independent, single-target filter, such as KF, EKF, UKF, or particle filter. For this project, we implemented the algorithm with an EKF.

The value of the MCMC-DA algorithm lies in two key innovations: (1) casting the problem as bipartite graph matching, and (2) efficient sampling via Markov Chain Monte Carlo.

1) Data Association as a Bipartite Graph: MCMC-DA introduces an alternative perspective to data association: the problem is simply bipartite graph matching.

In particular, we can encode the association information into a bi-partite graph $G = (U, V, E)$, where $U = \{y_t^j, 1 \leq j \leq N\}$ is the set of validated observations, $V = \{k, 1 \leq k \leq K\}$ is a vertex set of targets, and $E = \{(u, v) : u \in U, v \in V, \hat{P}^v(u|y_{1:t-1}) \geq \delta^u\}$ are the edges indicating the validated measurements, where $N \leq n_y$ is the number of validated measurements. An example from [12] is given in Figure 2, and examples of feasible partitions are given in Figure 4.



(a) (b)

Fig. 2. (a) An example of measurement validation with a 2D Gaussian estimate of $P^k(y|y_{1:t-1})$, with $K = 3$ targets (triangles) and $N = 8$ measurements (disks). (b) Measurement validation from (a) encoded as a bi-partite graph, where an edge between y^j and k indicates that a validated measurement between measurement j and target k. Taken from [12].

Not all bijective matchings between U and V will be likely. We can keep only the likely edges, and consider this to be the distribution representing the set of all possible graphs from which we can sample. Of course, one would want to sample efficiently from such a large set instead of enumerating all possible assignments.

C. Markov Chain Monte Carlo (MCMC) Preliminaries

MCMC is a method for approximating a complicated distribution via taking transitions in a Markov chain between states [1] [3]. We do not always accept high-probability transitions, which allows us to sample more than just the mode.

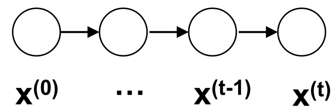
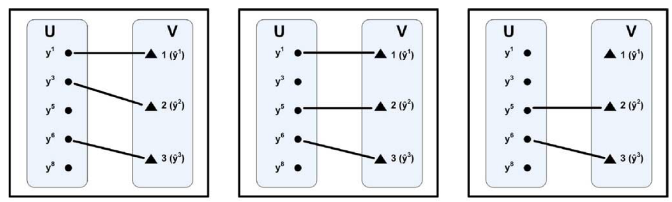


Fig. 3. A visualization of states in a Markov chain. After a burn-in period for which the chain can mix, Monte Carlo statistics are accumulated about instantiations of the distribution.

In the data association scenario, transitioning between assignment states denotes modifying the edges E of the graph G. Three moves exist: (1) switching the source or sink of an edge, (2) adding an edge to the graph, (3) or deleting an edge from the graph.



(a) (b) (c)

Fig. 4. Examples of feasible partitions of the graph shown in Figure 2(b). Movement from (a) to (b) is a an example of a switch move, and (b) to (c) a delete move.

We now describe the steps of the MCMC-DA algorithm.

Prediction Step: The first step is, naturally, the prediction step. The prediction step on each target can be written:

$$\hat{P}(X_t^k|y_{1:t-1}) := \int P(X_t^k|x_{t-1}^k)\hat{P}(x_{t-1}^k|y_{1:t-1})dx_{t-1}^k \quad (11)$$

which is simply the prediction step for each of the K independent filters.

Measurement Validation: The second step in the MCMC-DA algorithm reduces the complexity of the data association problem by eliminating unlikely associations from being considered. This is referred to as *measurement validation*. Let $\hat{P}^k(Y_t^j|y_{1:t-1})$ be the probability density of having observation Y_t^j given the previous measurements, assuming the measurement originated from target k. Then, for each (j, k) pair compute

$$\hat{P}^k(Y_t^j|y_{1:t-1}) = \int \hat{P}^k(Y_t^j|x_t^k)P(x_t^k|y_{1:t-1})dx_t^k \quad (12)$$

For the linear-Gaussian models used in the current project, this is identical to the measurement likelihood posterior given in Eq. 9. Measurement j is validated for target k if and only if

$$\hat{P}^k(y_t^j|y_{1:t-1}) \geq \delta^k \quad (13)$$

where δ^k is an appropriate threshold, and a tuning parameter for the algorithm. For our simulation, we found values of 0.05

to work well when paired with an R matrix with diagonal components of 0.5. The δ^k values essential act as a threshold on the distance between measurements and the estimated location of the robot that causes them to be considered possible matches. Therefore, measurements far away from the estimated location will not be considered. It is also important to note that, given appropriate thresholds, when the targets are far apart only one measurement per target is validated, and the problem simplifies significantly since the data association problem becomes trivial. The computation time is tightly related to the number of validated measurements per target.

Measurement Update: The measurement update for the MCMC-DA algorithm is where nearly all of the work is really accomplished. Here, we make use of the latent variable ω introduced in Eq. 3, which we now more formally define. Let $\omega = \{(j,k)\}$ represent a set of feasible associations between measurement j and target k for n_t measurements and K targets. Let $\Omega_t = \{\omega\}$ be the set of all feasible associations at time t. An association event ω is defined to feasible when:

1) for each $(j,k) \in \omega, y_t^j$ is validated for target k
2) an observation is associated with at most one target, and
3) a target is associated with at most one observation

To compute Ω in practice, we created two lists of sets: u_{adj} of length N and v_{adj} of length K. The j^{th} element of $u_{adj}^j = \{(u,v) : u = j\} \cup \emptyset$, and the k^{th} element of $u_{adj}^k = \{(u,v) : v = k\} \cup \emptyset$, which are the edges leaving each vertex, joined with a null set to capture the option of a vertex not having any assigned edges. We then define two sets: u_ω and v_ω, which are the Cartesian products of lists u_{adj} and v_{adj}, respectively. These capture all possible permutations of feasible edges from either U or V. The elements of u_ω will all be of length N and the elements of v_ω will be of length K, one for each measurement or target. We then define $u_{\tilde{\omega}}$ and $v_{\tilde{\omega}}$, where $u_{\tilde{\omega}}^j = \{(u,v) \in u_\omega^j : (u,v) \neq \emptyset\}$, and the elements of $v_{\tilde{\omega}}$ are similarly defined. These sets contain sets of edges, each one representing a possible association event. We can then calculate Ω_t, the list of all *feasible* association events, as $\Omega_t = u_{\tilde{\omega}} \cup v_{\tilde{\omega}}$.

We can re-write Eq. 3 as

$$\hat{P}(X_t^k|y_{1:t}) = \sum_{j=0}^{N} \beta_{jk} \hat{P}(X_t^k|\omega_{jk}, y_{1:t}) \tag{14}$$

Given ω_{jk} (the data association), calculating the posterior becomes a straight-forward application of the measurement update for the independent filter of choice. The association probability β_{jk}, on the other hand, is defined by the distribution

$$\beta_{jk} = \hat{P}(\omega_{jk}|y_{1:t}) = \sum_{\omega:(j,k)\in\omega} \hat{P}(\omega|y_{1:t}) \tag{15}$$

which requires a summation over exponentially many association events. Here we employ Markov Chain Monte Carlo (MCMC) sampling to efficiently sample from the posterior distribution $\hat{P}(\omega|y_{1:t})$ and calculate β_{jk}.

MCMC Sampling Step: Prior to running the MCMC sampling steps, it is convenient to calculate and store $\prod = \{\pi(\omega) : \forall\omega \in \Omega_t\}$, which is used as the proposal distribution for the acceptance probability in the Metropolis-Hastings algorithm. This is calculated as:

$$P(\omega|y_{1:t}) \approx \pi(\omega) = \frac{1}{Z}\lambda_f^{N-|\omega|} p_d^{|\omega|} (1-p_d)^{K-|\omega|} \prod_{(u,v)\in\omega} \hat{P}^v(u|y_{1:t-1}) \tag{16}$$

Algorithm 1: MCMC-DA($G, n_{mc}, n_{bi}, \prod$)

$\hat{\beta_{jk}} = 0, \forall j,k$
sample $\omega^{(0)}$ randomly from Ω_t
for $n = 1$ *to* n_{mn} **do**
  $w^{(n)} =$ MCMC-step$(G, \omega^{(n-1)}, \prod)$
  if $n > n_{bi}$ **then**
    foreach $(j,k) \in \omega^{(n)}$ **do**
      $\hat{\beta}_{jk} += 1/(n_{mc} - n_{bi})$
    end
  else
  end
end
Result: $\hat{\beta}_{jk}$

Algorithm 2: MCMC-step($G, \omega, \prod$

sample Z from Unif[0,1]
if $Z < 1/2$ **then**
  $\omega' = \omega$
else
  sample $e = (u,v) \in E$ uniformly at random
  if $e \in \omega$ **then**
    $\omega' = \omega - e$ `// deletion`
  else if *both* u *and* v *are unmatched in* ω **then**
    $\omega' = \omega + e$ `// addition`
  else if *exactly one of* u *and* v *is matched in* ω *and* e' *is the matching edge* **then**
    $\omega' = \omega + e - e'$ `// switch`
  else
    $\omega' = \omega$
end
$\omega = \omega'$ with probability $A(\omega, \omega')$

VI. RESULTS

VII. QUANTITIVE EVALUATION

We now compare the quantitative performance of the two algorithms.

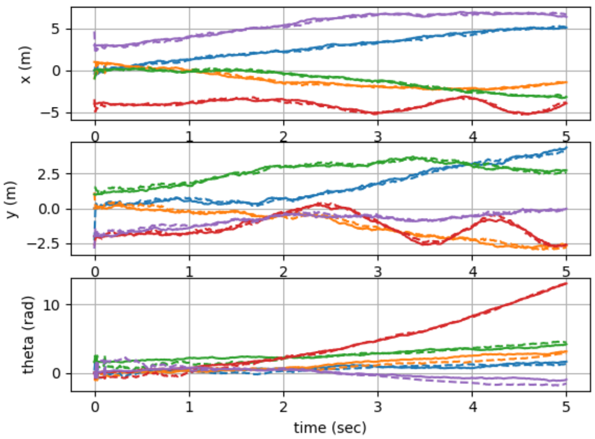


Fig. 5. Actual and estimated state values for a simulation with 5 robots, run with the MHKF filter.

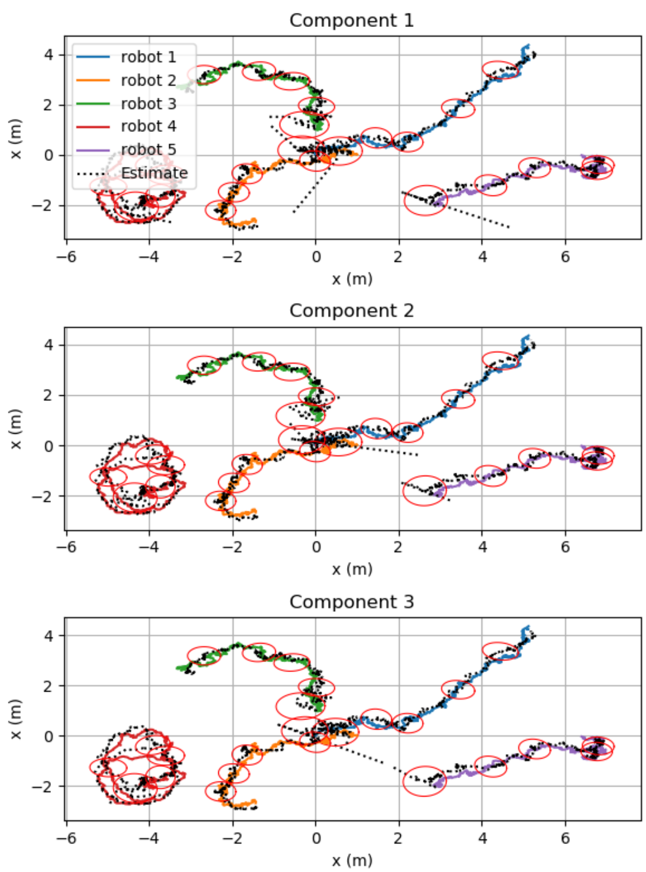


Fig. 6. Actual and estimated trajectories for a simulation with 5 robots, run with the MHKF filter.

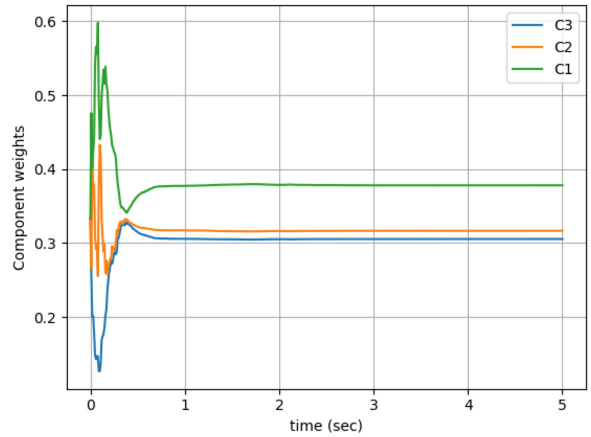


Fig. 7. Weights of the MHKF filter at each time step for a simulation with 5 robots

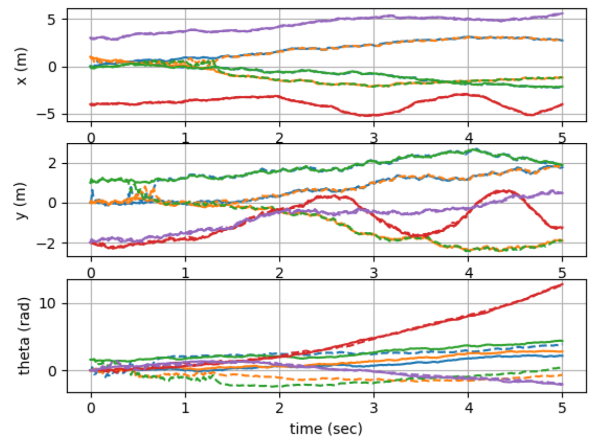


Fig. 8. Actual and estimated state values for a simulation with 5 robots, run with the MCMC-DA filter

A. Robot Dynamics

We use the following control laws for the $K = 3$ robot scenario:

$$\begin{aligned} u^1(t) &= \begin{bmatrix}\cos(0.1\cdot t) & \sin(0.1\cdot t)\end{bmatrix} \\ u^2(t) &= \begin{bmatrix}-\cos(0.2\cdot t) & \sin(0.2\cdot t)\end{bmatrix} \\ u^3(t) &= \begin{bmatrix}\cos(0.1\cdot t) & \sin(0.2\cdot t)\end{bmatrix} \\ u^4(t) &= \begin{bmatrix}t & t\end{bmatrix} \\ u^5(t) &= \begin{bmatrix}\sin(t)+0.4 & \cos(t)\end{bmatrix} \end{aligned} \tag{17}$$

B. Tracking Error

We note that when the measurement covariance R is increased above $0.5I_{2\times 2}$, with our $\delta^k = 0.05$ parameter, no measurements pass the measurement validation threshold. Thus, in all experiments we set $R \leq 0.5I_{2\times 2}$. However, as Table I shows, varying R within this range did not matter significantly, as data association for nearby robots can still be challenging.

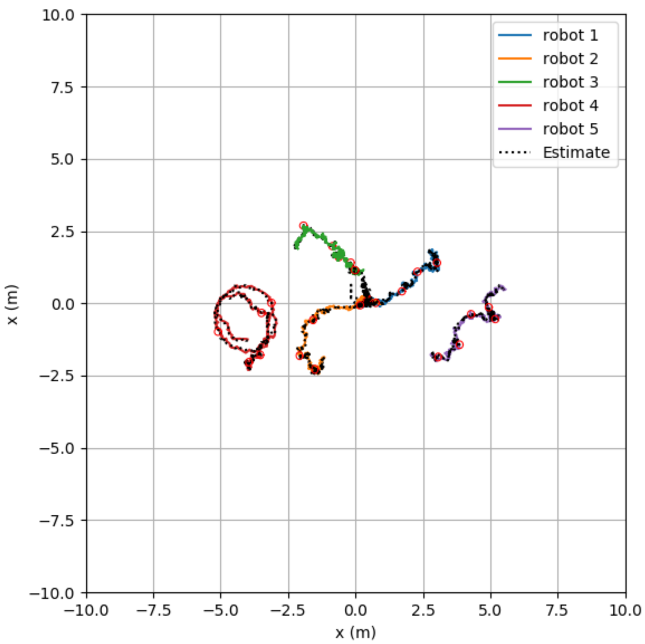


Fig. 9. Actual and estimated trajectories for a simulation with 5 robots, run with the MHKF filter.

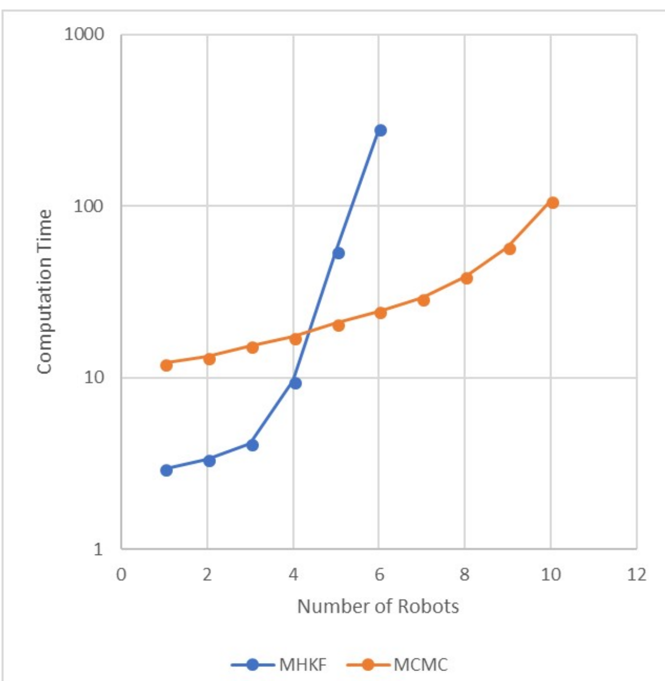


Fig. 10. We visualize computation time per filter step spent versus the number of robots tracked. The computational advantages of the MCMC-DA filter over the MHKF filter are evident. Note: y-axis is log-scale.

	3 Targets	4 Targets	5 Targets
$R = 0.5I_{2\times2}$			
MHKF	1.0948	-	-
MCMC-DA	0.3816	0.62615	0.5627
$R = 0.1I_{2\times2}$			
MHKF	1.0947	-	-
MCMC-DA	0.43945	0.31867	-

TABLE I

WE REPORT THE AVERAGE L2 NORM ERROR BETWEEN THE FILTER'S PREDICTED ROBOT LOCATION AND THE ACTUAL ROBOT LOCATION (SUMMED ACROSS ALL K ROBOTS) PER TIMESTEP. RESULTS ARE AVERAGED OVER 10 TRIALS, FOR A 10 SECOND TRACKING PERIOD WITH $\delta t = 0.01$ SECOND.

VIII. CONCLUSION

In conclusion, the performance of both filters is impressive and both perform qualitatively very well.

However, the MCMC-DA algorithm is advantageous in runtime complexity and in mean tracking error (See Figure 10, Table I) when tracking a large number of objects ($K > 4$). This can be attributed to its use of an independent filter on each target, whereas MHKF is forced to model a joint state. The MCMC-DA runtime appears linear in the number of targets.

We make our implementation publicly available at https://github.com/bjack205/MultiRobotTracking

IX. FUTURE WORK

The MHKF and MCMC-DA formulations are just two of many MTT formulations that exist today. Complicated distributions, such as the concatenated (joint) state of all robots that the MHKF models in each hypothesis, could also be approximated with a non-Gaussian object such as a Particle Filter.

We leave to future work the implementation of the Probability Density Hypothesis (PHD) filter , a FSST Sequential Monte Carlo filter, and a comparison with the Particle Filter for our MTT scenario. In addition, we also leave a comparison of ID F1 score, which may be more informative about the data association performance than the average error per timestep that we present in Table I.

We implemented a single-scan MCMC-DA filter, but a multi-scan algorithm would certainly be advantageous. Our simulation arena environment does also not account for the birth and death of targets, and we did not experiment with spurious measurements, dropped measurements, or uncertain robot motion models. We leave all of these extensions to future work.

REFERENCES

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. DOI: 10.1063/1.1699114. [Online]. Available: http://link.aip.org/link/?JCP/21/1087/1.

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[3] W. K. HASTINGS, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970. eprint: /oup/backfile/content_public/journal/biomet/57/1/57.1.97/4/57-1-97.pdf. [Online]. Available: http://dx.doi.org/.
[4] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
[5] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
[6] S. Mori, C.-Y. Chong, E. Tse, and R. Wishner, "Tracking and classifying multiple targets without a priori identification," *IEEE Transactions on Automatic Control*, vol. 31, no. 5, pp. 401–409, 1986.
[7] B.-n. Vo and S. Singh, "Sequential monte carlo implementation of the phd filter for multi-target tracking," in *In Proceedings of the Sixth International Conference on Information Fusion*, 2003, pp. 792–799.
[8] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.
[9] Z. Khan, T. Balch, and F. Dellaert, "An mcmc-based particle filter for tracking multiple interacting targets," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 279–290, ISBN: 978-3-540-24673-2.
[10] C. Kreucher, K. Kastella, and A. O. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, 2005.
[11] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A bayesian approach to multiple target detection and tracking," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1589–1604, 2007.
[12] 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, Mar. 2009, ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2012975.
[13] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*. Artech House, 2013.
[14] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15, Washington, DC, USA: IEEE Computer Society, 2015, pp. 4696–4704, ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.533. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.533.