iMHS: An Incremental Multi-Hypothesis Smoother

Fan Jiang, Varun Agrawal, Russell Buchanan, Maurice Fallon, and Frank Dellaert

Abstract—State estimation of multi-modal hybrid systems is an important problem with many applications in the field of robotics. However, incorporating discrete modes in the estimation process is hampered by a potentially combinatorial growth in computation. In this paper we present a novel incremental multi-hypothesis smoother based on eliminating a hybrid factor graph into a multi-hypothesis Bayes tree, which represents possible discrete state sequence hypotheses. Following iSAM, we enable incremental inference by conditioning the past on the future but we add to that the capability of maintaining multiple discrete mode histories, exploiting the temporal structure of the problem to obtain a simplified representation that unifies the multiple hypothesis tree with the Bayes tree. In the results section we demonstrate the generality of the algorithm with examples in three problem domains: lane change detection (1D), aircraft maneuver detection (2D), and contact detection in legged robots (3D).

I. INTRODUCTION

State estimation of multi-modal hybrid systems is an important problem with many compelling applications. A sample of these applications include fault detection [7], [9], air traffic control (ATC) [11], [29], decoding cortical motor activity [28], and contact dynamics in legged robots [8]. In the last example, accurate determination of contact state in manipulation and/or walking is challenging yet vital when executing accurate manipulation or locomotion plans.

However, incorporating discrete modes in the estimation process suffers from combinatorial growth in computation. Hence, there is a long history of work on filtering-based approaches, the most well-known of which are model adaptive estimation (MMAE) [18] and the interacting multiple model (IMM) [1]. This line of work has been further expanded to switching dynamics Bayes nets by Murphy [19], [20]. Optimizing over the entire time history or smoothing is much more challenging, although it has been approximated using Markov Chain Monte Carlo (MCMC) sampling [23], [22] and variational inference [6]. Various methods which keep track of sets of hypotheses has been proposed in the multi-target tracking literature, including the now classical multiple hypothesis tracking (MHT) filter [24], [4], the probabilistic MHT [25], JPDAF [1]. Both particle-filtering [26] and MCMC smoothing [16], [17], [21] have been applied.

In this paper we present a novel incremental multi-hypothesis smoother based on eliminating a hybrid factor graph into a multi-hypothesis Bayes tree, which represents possible discrete state sequence hypotheses. We base our work on iSAM [15], [13], which enables incremental inference by conditioning the past on the future, but we add to that the capability of maintaining multiple discrete mode histories. In contrast to sampling-based and variational approaches, the computational cost is bounded by meaningful thresholds on the probability of state histories, and incremental inference is constant-time. While our approach is similar in spirit to MH-iSAM2 by Hsiao and Kaess [10], we exploit the temporal structure of the problem to obtain a simplified representation that unifies the multiple hypothesis tree with the Bayes tree. In addition, we support stochastic transitions over the mode sequence, and can properly handle model selection when modes are not all equally informative about the dynamics or measurement models.

A small example in Figure 1 demonstrates the basic idea. The batch problem is represented as a factor graph with the state at three time instants, and a Markov chain of discrete modes that select a different process model at each time step. The unary factors are measurements and will help determine which process model is most likely. In the case of a binary mode variable, the mode selects between
two process models. For this example with $K = 3$ states there are eight possible discrete mode sequences. A naive algorithm would eliminate the factor graph separately for each sequence. However, Figure 2b shows that the eliminated factor graphs can be combined into a multiple-hypothesis tree of conditionals, conditioned on the future states.

In the results section we show examples from vehicle lane-change detection, aircraft tracking, and contact estimation of legged robots, all of which can be modeled in the above framework. In general, any system described by a dynamic Bayes net with a combination of discrete and continuous variables can be accommodated.

In summary, the contributions of this work are:

- A novel, incremental smoother algorithm for hybrid/multi-modal systems.
- A novel data structure that unifies multiple hypothesis trees and Bayes trees.
- A demonstration of the generality of the algorithm with examples in three problem domains: lane change detection (1D), aircraft maneuver detection (2D), and contact detection in legged robots (3D).

II. BACKGROUND

In this paper we rely heavily on the use of graphical models, including dynamic Bayes nets, factor graphs, and incremental inference, which we will review in this section.

A. Problem statement and notation

Continuous smoothing is the problem of recovering the posterior density

$$p(X^K | M^{K-1}, Z^K),$$

where $X^K = \{x_k\}_{k=0}^{K-1}$ is a sequence of $K$ states from time $t_0$ up to and including time $t_{K-1}$, with $x_k \in \mathcal{X}$ representing the state at time $t_k$. Given a similarly defined sequence of discrete modes $M^{K-1}$ and measurements $Z^K$, with $m_k \in \mathcal{M}$ and $z_k \in \mathcal{Z}$, we model the joint density over all variables using the following dynamic Bayes net [20] (Figure 2a):

$$p(X^K, M^{K-1}, Z^K) = P(x_0)p(z_0|x_0)p(m_0) \times \prod_{k=1}^{K-1} p(m_k, x_k, z_k|m_{k-1}, x_{k-1}).$$

We assume the mode sequence $M^{K-1}$ is modeled by a Markov chain $P(m_k|m_{k-1})$. The continuous states evolve via a switching motion model $p(x_k|x_{k-1}, m_{k-1})$, and measurements are modeled on the continuous states as $p(z_k|x_k)$. One time slice in the dynamic Bayes net is given by

$$p(m_k, x_k, z_k|m_{k-1}, x_{k-1}) = P(m_k|m_{k-1})p(x_k|x_{k-1}, m_{k-1})p(z_k|x_k).$$

B. Factor Graphs and Trajectory Optimization

When both discrete modes $M^{K-1}$ and measurements $Z^K$ are known, it is convenient to represent the posterior density of $X^K$ using a factor graph [5] as illustrated in Figure 2b

$$\phi(X^K) \propto p(X^K | M^{K-1}, Z^K)$$

$$= P(X^K)p(M^{K-1})$$

with the factors defined in correspondence with (2).

The maximum a posteriori (MAP) estimate can be recovered by maximizing $\phi(X^K)$ with respect to the continuous states $X^K$, also known as trajectory optimization:

$$X^{K*} = \arg \min_{X^K} \phi(X^K).$$

By fitting a quadratic to the log-posterior, e.g., using the estimate $A^T A$ to the Fisher information matrix $F$, where

$$A = -\frac{\partial \log \phi(X^K)}{\partial X^K},$$

we can obtain an approximation to the posterior as

$$p(X^K | M^{K-1}, Z^K) \approx \mathcal{N}(X^K; X^{K*}, F^{-1}),$$

where $\mathcal{N}(x; \mu, \Sigma) \propto \exp \{-0.5 ||x - \mu||^2_\Sigma\}$ is the multivariate normal distribution with mean $\mu$ and covariance $\Sigma$.

C. The linear case: Kalman smoothing

In the common case when all continuous densities are linear-Gaussian and (w.l.o.g.) time invariant, we have

$$p(x_k|x_{k-1}, m) = \mathcal{N}(x_k; F_m x_{k-1} + B_m u_{k-1}, Q_m)$$

$$p(z|x, m) = \mathcal{N}(z; H_m x, R_m)$$

with a vector-valued state $X^K \in \mathbb{R}^n$, a control vector $u^k \in \mathbb{R}^p$, and a vector-valued measurement $z_k \in \mathbb{R}^m$. Matrices $F_m, B_m,$ and $H_m$, as well as the process and noise covariances $Q_m$ and $R_m$ can depend on the mode $m$. It is convenient to define factors in negative log-space, and to
In this case the approximation (7) is exact, with $A$ the sparse Jacobian of $f(X^K)$, and the mean $X^K$. These can be found in closed form by solving the normal equations

\[(A^T A)X^K* = A^T b\]  

with the right-hand size (RHS) $b$ suitably defined.

D. Incremental Inference

Recovering the posterior $p^*(X^K)$ -solving the normal equations in the linear case- can be done using sparse elimination [5]. If we eliminate with a natural, time-aligned ordering then under our assumptions the posterior density $p^*(X^K) \doteq p(X^K|M^{K-1}, Z^K)$ will have the form of a chain of conditional densities,

\[p^*(X^K) = \left\{ \prod_{k=1}^{K-1} p^*(x_{k-1}|x_k) \right\} p^*(x_{K-1}),\]  

where we use $*$ to indicate that these are conditioned on the measurements and mode sequence. As illustrated in Figure 3 with this factorization the past is conditioned on the future, which enables a simple incremental inference scheme.

The goal of incremental inference is to obtain the posterior density $p^*(X^{K+1})$ at the next time step given the current posterior density $p^*(X^K)$, the new discrete mode $m_{k-1}$, and a new measurement $z_k$:

\[p(X^{K+1}|M^K, Z^{K+1}) \leftarrow p(X^K|M^{K-1}, Z^K)\]  

Following iSAM2 [13] it suffices to assemble the following factor graph fragment

\[\phi(x_{K-1})\phi^m_k(x_{K-1}, x_K)\phi^*_k(x_K)\]  

where the first factor $\phi(x_{K-1})$ corresponds to the density $p^*(x_{K-1})$ from [15]. We then eliminate this into a replacement for $p^*(x_{K-1})$, obtaining two conditionals:

\[p^*(x_{K-1}|X^K)p^*(x_K).\]  

In the linear case, the incremental update can be done in square-root information form using QR factorization [2]. [14]:

\[
U \begin{bmatrix} A_{K-1} & b_{K-1} \\ I_n & -F_m & B_m u_{K-1} \\ H_m & 0 & z_K \end{bmatrix} \rightarrow \begin{bmatrix} R_m & S_m & u_m \\ T_m & v_m & e_m \end{bmatrix}
\]

where the left matrix, augmented with its RHS, corresponds to the factor graph fragment, and $U \doteq \text{diag}(P, Q_m, R)^{-0.5}$ collects the noise covariances. The upper-triangular matrix on right, also with RHS, corresponds to the resulting Bayes net fragment [18]. Extracting the latter yields the two Gaussian conditional densities on $X_k$ and $x_k$:

\[p^*(x_{K-1}|x_K) = N(x_{K-1}; R_m^{-1}(u_m - S_m x_K), R_m^{-1}R_m^-T)\]

\[p^*(x_K) = N(x_K; T_m^{-1}v_m, T_m^{-1}T_m^-T). \]

When the factor graph fragment matrix has rank exactly $2n$, the error at the minimum of the associated quadratic is zero; otherwise, it is $0.5 e_m^T e_m$. We will use this result below.

In the nonlinear case, ISAM [15] and iSAM2 [13] provide incremental management of linearization points, and also handle more general factor graph topologies.

III. APPROACH

All of the above assumed known hybrid modes, in which case incremental smoothing is both well known and efficient. In this section we present the main contribution of the paper, which uses a multiple hypothesis framework to effectively run many smoothers in parallel, but arranged in a tree to accommodate shared history. The probability of a particular mode sequence depends both on a Markov chain prior and on how compatible the estimated continuous state trajectory is with a particular mode sequence. For example, in a lane change scenario, the changing lateral position of the car provides evidence that a lane change maneuver is in progress.

A. Hybrid State Estimation

When the hybrid modes $M^{K-1}$ are unknown, we have a hybrid state estimation problem. In this section, we add the mode sequence $M^{K-1}$ as unknowns to the factor graph:

\[\phi(X^K, M^{K-1}) \doteq \phi(m_0)\phi(x_0)\phi_0(x_0)\times \prod_{k=1}^{K-1} \phi(m_{k-1}, m_k)\phi(x_{k-1}, x_k, m_k)\phi^*_k(x_k)\]

Above we added a prior on $m_0$, a mode transition model $\phi(m_{k-1}, m_k)$, and the motion model factors $\phi^m_k(x_{k-1}, x_k, m_k)$ are now also a function of the mode $m_k$.

Within a batch smoother, when we eliminate the states $X^K$ with the oldest time-aligned ordering, and then eliminate the entire mode sequence, we can obtain a chain of state densities $p(x_{K-1}|x_k, M^K)$ conditioned on the future, and on the mode.
from the MHT [24]. As discussed above, Equation 22 hides structure that has the same structure as the hypothesis tree $B$. The Multi-hypothesis Smoother

$$\tau_{tially with \tau is a table of many continuous densities. The cardinality of a sequence $M^k$ up to that time:

$$p(X^K, M^{K−1}|Z^K) = \left\{ \prod_{k=1}^{K−1} p^*(x_{k−1}|x_k, M^k) \right\} p^*(x_{K−1}|M^{K−1}) p^*(M^{K−1})$$

(22)

The cardinality of a sequence $M^k$ is $|\mathcal{M}|^k$, grows exponentially with $k$. Hence, each conditional in the product above is a table of many continuous densities.

Eliminating the last state $x_{K−1}$ yields a new factor $\tau(M^{K−1})$ on the entire mode sequence:

$$\tau(M^{K−1}) = C_m \int_{x_{K−1}} q^*(x_{K−1})$$

(23)

where $q^*(x_{K−1})$ is the unnormalized posterior density on $x_{K−1}$, and $C_m$ is a mode-dependent constant that derives from the motion model and measurement factors in (4).

B. The Multi-hypothesis Smoother

The Bayes net chain (22) can be arranged in a data structure that has the same structure as the hypothesis tree from the MHT [24]. As discussed above, Equation 22 hides $|\mathcal{M}|^{K−1}$ different chains, one for each mode sequence. A disadvantage of working with that representation is that the arity of each conditional increases over time, and the multi-hypothesis nature of the smoothing problem is hidden from view. However, because each conditional density on $x_{k−1}$ only depends on the subset of modes $M^k = \{m_i\}_{i=1}^{k−1}$, we can arrange the conditionals in a tree. As an example, the tree corresponding to Figure 4a is shown in Figure 5.

Formally, the tree distributes every conditional density $p^*(x_{k−1}|x_k, M^k)$ in the switching Bayes net (22) over sets of $|\mathcal{M}|^{K−1}$ single-mode switching conditional densities, indexed by their mode history prefix $M^{k−1}$:

$$p^*(x_{k−1}|x_k, M^k) = \{p^*_m(x_{k−1}|x_k, m_{k−1})\}_{M^{k−1}}$$

(24)

E.g., the first three conditionals correspond to these sets,

$$p^*(x_0|x_1, m_0) = \{p^*_m(x_0|x_1, m_0)\}_{M^{0} = \emptyset}$$

$$p^*(x_1|x_2, m_0, m_1) = \{p^*_m(x_1|x_2, m_1)\}_{M^{1} = \{0\}}$$

$$p^*(x_2|x_3, m_0, m_1, m_2) = \{p^*_m(x_2|x_3, m_2)\}_{M^{2} = \{00, 01, 11\}}$$

which correspond exactly to the inner nodes shown in Figure 4b. The final layer in the tree is the set of densities on the final state $x_{K−1}$ conditioned on the entire mode sequence $M^{K−1}$:

$$p^*(x_{K−1}|M^{K−1}) = \{p^*_m(x_{K−1})\}_{M^{K−1} = \ldots}$$

(25)

Recovering the optimized state sequence $X^{K*}$ for any leaf node $p^*_m(x_{K−1})$ can be done simply by back-substitution in reverse time order, following the directed edges in the tree until the root. In particular, the estimate corresponding to the highest value of $\tau(M^{K−1})$ (Eqn. 23) corresponds to the MAP estimate. The corresponding optimal mode sequence $M^{K−1*}$ can be recovered in a similar fashion, if one takes care to annotate the chosen mode on the edges in the tree.

C. Incremental Inference

In an incremental scheme this tree grows over time, as illustrated in Figures 5[b]. At each incremental smoothing iteration, all of the leaves of the tree are replicated $|\mathcal{M}|$ times. Hence, we have to consider an exponentially growing number of hypotheses at each step.

However, extending each leaf node towards the future is simple and fast, and we can incrementally keep track of the probability of each mode sequence. In particular, after $K$ steps, when estimating the next state $x_{K}$, assembly and elimination of the factor graph fragment proceeds as in Section 2-D. We then use 23 to calculate $\tau(M^K)$ and add it to the previous value $\tau M^{K−1}$ to get an updated $\tau$ value. Book-keeping is simple as each leaf only has to store one $\tau$ value: the number of leaves is exactly $|\mathcal{M}|^K$ after extending, which is the size of the new $\tau(M^K)$ table.

The integral (23) depends on the problem, but can be done in closed form in the linear-Gaussian case. In particular, the QR factorization (19) yields

$$q^*(x_{K}) = \exp \left\{ -\frac{1}{2} \| T_m x_K - v_m \|^2 - \frac{1}{2} e^T_m e_m \right\}$$

(26)
where the constant $C_m$ derives from (12) and (13):

$$C_m = \sqrt{|I_m I_z|} \text{ where } I_m = Q_m^{-1} \text{ and } I_z = R_m^{-1}$$  \tag{27}$$

Note that $C_m$ can be omitted if the measurements and process covariances are not dependent on the mode $m$. With this, integration of (23) for the factor on the extended sequence $M^K$ is given by

$$\tau(M^K) = \exp\left\{ -\frac{1}{2} e_m^T e_m \right\} \sqrt{\frac{|T_m T_z|}{|T_m| |T_z|}}$$  \tag{28}$$

The first factor above penalizes error, and the latter is the ratio between the mode-dependent information in the factors (numerator) and the information in the posterior (denominator). The discrete factor $\tau(M^K)$, of size $|M|^K$, can then be multiplied with $p(m_{K-1}|m_{K-2})$ into $p^*(M^{K-1})$ to yield the posterior $p^*(M^K)$ on the new mode sequence $M^K$.

Working in log-space avoids numerically unstable multiplication of many small numbers. Taking the negative log of (28) yields

$$\frac{1}{2} e_m^T e_m + \log |T_m| - \frac{1}{2} \log |I_m I_z|$$  \tag{29}$$

where we made use of the fact that $T_m$ is square.

**D. Pruning and Marginalization**

Unbounded computational cost can be avoided by setting a pruning threshold $\theta$, and marginalizing out older states.

We chose a pruning scheme that evaluates the posterior on the mode sequence $M^{K-1}$ before extending the leaves. Any leaf with an associated probability less than $\theta$ will not be extended. In practice we choose $\theta = 1\%$ or $\theta = 0.1\%$, which provides a good compromise between accuracy and computation in the examples we tried.

Marginalizing out older states, e.g., after a fixed lag $N$, is straightforward: we simply discard parts of the tree further than $N$ levels from the current time step. Note that this can lead to the tree becoming a forest, but this is not an issue as leaves can continue to be expanded. Another scheme is to simply drop the nodes before the last fork. With aggressive pruning this can occur at a quite shallow depth.

Choosing a high value for $N$ does not incur a high cost. The computational cost of keeping a linear chain before the last fork is zero, and memory load is linear in time. Note however that recovering optimal state sequences at each time does require computation. If this is needed, a wildfire threshold as in iSAM2 [13] can be adopted here as well.

**E. Metrics**

Finally, evaluating estimated mode sequences needs some care. Each leaf in the tree with parameters $\Theta$ corresponds to a hypothesis for the corresponding mode history. When given a ground truth mode sequence $Y^{K-1}$, we are faced with evaluating how good our estimate is with respect to
where the system matrices $F$ and $C$ are given by

$$
F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
C = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}
$$

with corresponding noise covariance matrices $Q$ and $R$,

$$
Q = \text{diag}(\sigma^2_T, \sigma^2_V),
R = \begin{bmatrix} \sigma^2_T & 0 \\ 0 & \sigma^2_V \end{bmatrix}
$$

Fig. 8 shows the detected lane change events for a car driving changing lanes for 4 times in a 500m path.

**Aircraft Maneuver Detection**

To quantitatively evaluate the iMHS algorithm, we run the algorithm on the 2-mode switching aircraft tracking task from [11]. The aircraft model has 2 discrete modes, constant velocity (CV) and coordinated turn (CT). The dynamics of this model is given by the following set of discrete-time system equations:

$$
x(k+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T^2 & 0 \\ 0 & \frac{T^2}{2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u_i(k) + w_i(k)
$$

$$
y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k) + v_i(k)
$$

In our simulation, $w_i(k)$ and $v_i(k)$ are both zero-mean Gaussian process noise with a standard deviation of [1, 1, 1, 1]$^T$. For each mode, the two constant control inputs were assigned to be $u_{CV} = [0, 0]^T$ and $u_{CT} = [1.5, 1.5]^T$.

The results (Fig. 9) shows that the iMHS algorithm is able to accurately identify the two modes of flight. In the figure, the green and pink background colors indicate the ground truth mode. Note that the change of mode is identified without any delay, whereas filtering algorithms like RMIMM always have a delay $\geq 1$ time step at state transitions [11].

To verify the performance of our system we can also run it as a filter. In this mode the algorithm runs online and outputs the current mode estimate as soon as a measurement is taken. We ran this configuration with the more complex trajectory
data from an A1 and actual log data from an ANYbotics ANYmal quadruped (Fig. 12) walking in Keble College, Oxford. The latter robot and its sensors are described in [27], and the dataset we used comprises of several trotting sequences around a courtyard. The robot’s state estimator [3] gives us the position of the base link $T_B^O$ and the feet $p_i^O$ in an odometry frame $O$ that is initialized at the start of a run and updated using an internal state estimator.

We frame the problem as estimating the foot trajectory jointly with the contact sequence $M^{k-1}$ where the mode at time $t_k$ is binary, indicating contact with the ground. We model the motion of a foot during contact as a 3-dimensional Gaussian density, whose parameters $\mu_i^O$ and $\Sigma_i^C$ are estimated from training runs. This leads to a quadratic factor between successive foot positions $p_{k-1}^O$ and $p_k^O$

$$f_k^C(p_{k-1}^O, p_k^O) = ||p_k^O - p_{k-1}^O - \mu_i^O||^2_{\Sigma_i^C},$$

which is expected to have zero mean with a tight covariance. During swing, however, we model a constraint on the foot positions $p_{k}^B$ in the base frame, reflecting our knowledge that the feet move rigidly with respect to the body during swing. We obtain:

$$f_k^S(p_{k-1}^O, p_k^O) = ||T_{O,k}^B p_k^O - T_{O,k-1}^B p_{k-1}^O - \mu_S^B||^2_{\Sigma_S}$$

where we made use of $p_{k}^B = T_{O,k}^B p_k^O$, with $T_{O,k}^B = (T_{O,k}^B)^{-1}$. We have $T_{O,k}^B p_O = T_{O,k}^B p_O + t_{O,k}^B$ and we assume $R_{O,k}^B \approx R_{O,k-1}^B$, which allows us to simplify this into another quadratic factor,

$$f_k^S(p_{k-1}^O, p_k^O) = ||p_k^O - p_{k-1}^O - (\Delta t_{B,k-1}^O + \Delta t_{B,k-1}^O)||^2_{\Sigma_S},$$

where $\Delta t_{B,k-1}^O = t_{O,k}^O - t_{O,k-1}^O$.

Qualitative results for the A1 simulation and the Keble dataset are shown in Figures 13 and 14, respectively. For
Fig. 13: A1 simulation results. Panel (a) and (b) represents the marginal probabilities (32) and the MAP estimate for the contact sequence of all feet, with black signifying contact. Panel (c) is the error for the MAP estimate, color-coded with red for incorrectly predicted contact, green for incorrectly predicted flight, and white for no error.

Fig. 14: ANYmal B. See caption of Figure 14

V. CONCLUSIONS

The incremental Multi-Hypothesis Smoother proposed in this paper provides a solid foundation for inference over time in hybrid dynamical systems. Our results above demonstrate the applicability in various domains, but we are most excited about the possibility of integrating this with a state of the art inertial state estimation pipeline. This would enable state estimation in a variety of legged robots without the need for contact sensors or dedicated force sensors. Smoothing rather than filtering will enable state estimators to more accurately update continuous states that depend on hybrid states.

REFERENCES

[1] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc., New York, USA, 2001.
[2] G. J. Bierman. Sequential square root filtering and smoothing of discrete linear systems. *Automatica*, 10(2):147–158, Mar. 1974.
[3] M. Blösch, M. Hutter, and M. A. H. Höpflinger. State estimation for legged robots - consistent fusion of leg kinematics and imu. RSS, 2012.
[4] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, Feb. 1996.
[5] F. Dellaert and M. Kaess. Factor Graphs for Robot Perception. *Foundations and Trends in Robotics*, 6(1-2):1–139, 2017.
[6] Z. Dong, B. A. Seybold, K. P. Murphy, and H. H. Bui. Collapsed amortized variational inference for switching nonlinear dynamical systems. In *ICML*, page 10, 2020.
[7] P. Eide and P. Maybeck. An MMAE failure detection system for the F-16. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3):1125–1136, July 1996.
[8] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955–1988, Aug. 2014.
[9] P. Hanlon and P. Maybeck. Multiple-model adaptive estimation using a residual correlation Kalman filter bank. *IEEE Transactions on Aerospace and Electronic Systems*, 36(2):393–406, Apr. 2000.
[10] M. Hsiao and M. Kaess. MH-iSAM2: Multi-hypothesis iSAM using Bayes Tree and Hypo-tree. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1274–1280, Montreal, QC, Canada, May 2019. IEEE.
[11] I. Hwang, H. Balakrishnan, and C. Tomlin. State estimation for hybrid systems: Applications to aircraft tracking. *IEEE Proceedings - Control Theory and Applications*, 153(5):556–566, Sept. 2006.
[12] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter. Dynamic locomotion on slippery ground. *IEEE Robotics and Automation Letters*, 4(4):4170–4176, 2019.
[13] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, Feb. 2012.
[14] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1670–1677, Rome, Italy, Apr. 2007. IEEE.
[15] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, Dec. 2008.
[16] Z. Khan, T. Balch, and F. Dellaert. MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 34, 2005.
[17] Z. Khan, T. Balch, and F. Dellaert. MCMC Data Association and Sparse Factorization Updating for Real Time Multitarget Tracking with Merged and Multiple Measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):13, 2006.
[18] P. S. Maybeck. Stochastic Models, Estimation and Control. Academic Press, 1979.
[19] K. P. Murphy. Switching Kalman Filters: Technical report, U.C. Berkeley, 1998.
[20] K. P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, U.C. Berkeley, 2002.
[21] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, Mar. 2009.
[22] S. M. Oh, J. Rehg, T. Balch, and F. Dellaert. Learning and inference in parametric switching linear dynamic systems. In *Tenth IEEE International Conference on Computer Vision (ICCV’05)* Volume 1, pages 1161–1168 Vol. 2, Beijing, China, 2005. IEEE.
[23] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Data-Driven MCMC for Learning and Inference in Switching Linear Dynamic Systems. In AAAI, page 6, 2005.

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

[25] R. L. Streit and T. E. Lugibuhl. Probabilistic multi-hypothesis tracking. Technical report, Naval Undersea Warfare Center Division, Feb. 1995.

[26] J. Vermaak, S. Godsill, and P. Perez. Monte Carlo filtering for multi target tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41, Jan. 2005.

[27] D. Wisth, M. Camurri, and M. Fallon. Robust legged robot state estimation using factor graph optimization. *IEEE Robotics and Automation Letters*, 4(4):4507–4514, 2019.

[28] W. Wu, M. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. Donoghue. Modeling and Decoding Motor Cortical Activity Using a Switching Kalman Filter. *IEEE Transactions on Biomedical Engineering*, 51(6):933–942, June 2004.

[29] J. L. Yepes, I. Hwang, and M. Rotea. New Algorithms for Aircraft Intent Inference and Trajectory Prediction. *Journal of Guidance, Control, and Dynamics*, 30(2):370–382, Mar. 2007.