Distributionally Robust State Estimation for Jump Linear Systems

Shixiong Wang

***Abstract*—In practice, the designed nominal model set for a jump (Markov) linear system might be uncertain: 1) Every candidate model might be inexact due to, e.g., mismatched modeling assumptions or model's identification errors; 2) The nominal model set might be incomplete (e.g., the true system has three operating modes but the designed model set includes only two of them). Moreover, the designed model transition probability matrix (TPM) might be uncertain: There is a discrepancy between the designed and true TPMs. Neglecting such model uncertainties and employing nominally optimal multi-model state estimators may cause significant performance losses. Therefore, this paper proposes a robust state estimation framework for jump linear systems that is insensitive to these three types of model uncertainty, technically by leveraging the distributionally robust optimization theory. Specifically, the model uncertainties are quantified by a collection of mixture distributions lying in a distributional ball centered at the nominal mixture distribution, where the nominal mixture distribution represents the nominal state-measurement distribution defined by nominal candidate models and their nominal model weights. Then, the robust state estimation is taken over the least-favorable distribution in the employed distributional ball. We show that the distributionally robust state estimation problem for jump linear systems can be reformulated into a tractable optimization equivalent such as a quadratic program or a positive semi-definite program, and in a special case, it can be analytically solved. Simulated and real-world experiments suggest that the proposed method is particularly useful when large model uncertainties exist.**

***Index Terms*—Jump Linear Systems, Robust Filter, Distributional Robustness, Quadratic Program, Semi-Definite Program.**

I. Introduction

A. Background

An actual physical plant or an information system may work in several different modes, and the modes may jump from one to another as time proceeds. For example, in multi-model target tracking [1], at each time step, a target can move according to any one of the following models [2]: the constant velocity (CV) model, the constant turn (CT) model, the constant acceleration (CA) model, the Singer model, etc. To track the evolution of this system, we are concerned with estimating its hidden state given measurements in the past. This paper considers Markov jump linear system models. The exactly optimal state estimation method for jump linear systems is computationally intractable because the number of the required state estimators grows exponentially as the time proceeds, and therefore, it is not implementable in practice [3], [4]. Due to the intractability of the optimal state estimator, till now, the most popular method to handle the state estimation problem of jump linear systems is the interactive multiple model (IMM) filter, which **approximates** the filtered (i.e., posterior) state distribution using a limited number of Gaussian components [3], [5]. The IMM filter is pragmatically attractive because its computational burden is as low as the generalized pseudo-Bayesian estimator of first order (GPB1) but it has high performance as the generalized pseudo-Bayesian estimator of second order (GPB2) [3]. It is believed that the IMM filter can provide an excellent compromise between the filtering performance and the computational complexity [3], [6]–[10].

S. Wang is with the Institute of Data Science, National University of Singapore, Singapore 117602 (E-mail: s.wang@u.nus.edu).

This research is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018).

B. Problem Statement

In practice, the IMM filter (and also other multi-model filters) faces the following limitations.

U1) The nominal model set might be uncertain; i.e., at some times, none of the nominal models in the nominal model set can exactly describe the true system dynamics. This can be understood from two aspects.

a) For every nominal model in the model set, it is an approximation to the true operating dynamics for the mode, and therefore, model mismatch exists. In multi-model target tracking, for instance, the nominal constant-turn (CT) model may be different from the true CT model because filter designers never exactly know the true turning rate, and as a result, a misidentified value of the turning rate may be used.

b) The nominal model set is not complete. For example, the number of the actual operating modes is larger than the size of the nominal model set. In multi-model target tracking, the target may move according to a great number of models [2], but filter designers may only use some of them (e.g., only CV, CA, and CT).

U2) The model transition probability matrix might be uncertain [7]–[9], [11], [12]. For example, in multi-model target tracking, model transition probabilities from one motion model to another may not be exactly known.

Hence, a state estimation framework for jump linear systems that is able to handle the listed two types of uncertainties in Items U1 and U2 is expected.

C. Literature Review And Research Aims

The treatment frameworks for uncertainties in a single candidate model have been comprehensively surveyed and discussed in [13], [14], including the uncertainties in system

matrices and the uncertainties in noise distributions. When model sets are complete and TPMs are exact, these methods can be applied (or extended) for jump linear systems; cf., e.g., [15] and [16]. Therefore, this paper focuses on the review of treatment methods for the ad-hoc modeling uncertainties in jump linear systems, i.e., the incompleteness of nominal model sets and the uncertainties in TPMs. Since the incompleteness of nominal model sets has never been discussed in the literature, we pay attention to the uncertainties in TPMs. Several researches, e.g., [7]–[9], [11], [12], [17]–[19], have discussed the state estimation problem for jump linear systems when the model transition probability matrix is unknown or uncertain. These works can be categorized into two streams. The first stream aims at obtaining the accurate estimate of the **unknown** TPM using the Frequentist method [8], [12], [17] or the Bayesian method [7], [11], while the second stream tries to design robust state estimators that are insensitive to the **uncertain** TPM, e.g., the compensation-based method in [9] and the $\mathcal{H}_\infty$ method in [18]–[20]. However,

L1) The Frequentist method [8], [12], [17] and the Bayesian method [7], [11] assume that the true TPM is a time-invariant matrix. When the true TPM is (significantly) time-varying, these methods cannot provide the exact estimate of the true TPM anymore. In addition, even when the true TPM is time-invariant, a sufficiently long time horizon (i.e., sufficient measurements) is expected to estimate the TPM to a satisfactory level. Also, choosing satisfactory prior distributions for the unknown TPM and determining the parameters of the prior distributions are problem-dependent, and therefore, frustrating. Last but not least, albeit the true TPM is time-invariant, the estimate of the unknown TPM can hardly be exactly the same as its true value so that there still exists uncertainty (i.e., parameter identification error) in the estimated TPM;

L2) The performance of the compensation-based method in [9] is problem-specific and it is desirable only when there exists an overwhelmingly dominating model at each time step (i.e., one of the model probabilities of candidate models is outstandingly large). The $\mathcal{H}_\infty$ method in [18]–[20] requires that, for every candidate model, the noise sequences have finite energies, which implies that the power of the noises sequences vanish as time proceeds. Also, it requires the jump systems to be mean square stable. These two assumptions are obviously not always the case for the state estimation problem for general jump linear systems (e.g., in target tracking problems [2], [4], systems are usually non-stable).[1]

When the existence of model uncertainties is noticed but we have no specific information of where and how the uncertainties exist (e.g., whether system matrices, noise distributions, or TPMs are uncertain), the generic robust multi-model state estimators in [21], [22] are applicable. However, these generic methods tend to be overly conservative because they cannot specifically respond to given types of uncertainty. To clarify further, when we exactly know that the model uncertainties exist only in TPMs, robust multi-model state estimators should not admit possible uncertainties in other components (e.g., system matrices and noise distributions) and only respond to uncertainties in TPMs.

Therefore, a unified multi-model state estimation framework for jump linear systems that is insensitive to the listed two types of uncertainty (i.e., Items U1 and U2) is expected. In particular, the new framework is supposed to flexibly respond to specified types of uncertainty.

D. Contributions

Following the conventional route [7]–[9], [11], [12], [17], this paper particularly robustifies the IMM filter. Specifically, a distributionally robust IMM filtering framework for jump linear systems that is insensitive to the two types of uncertainty listed in Subsection I-B is proposed; see (10), (11), and (12). Subsequently, the explicit optimization equivalents of the distributionally robust state estimation problem (11) subject to (12) are derived in Propositions 1 and 2. We show that these explicit optimization equivalents can be further reformulated into tractable optimization equivalents such as quadratic programs [see, e.g., (26)] or positive semi-definite programs [see, e.g., (34)], in different scenarios. Then, we show that these tractable reformulations can be efficiently solved using either the off-the-shelf solvers or the specifically-designed efficient algorithms (see, e.g., Proposition 4). Particularly, in a special case, the reformulated problem can be analytically solved; see Theorem 2. Finally, the distributionally robust IMM filter is summarized in Algorithm 1. Experiments suggest that the proposed method is particularly useful when complex and large model uncertainties exist in the nominal model; to be specific, for example, the method is especially suitable for tracking highly-maneuvering targets.

E. Notations

The space of all n-dimensional vectors is denoted by $\mathbb{R}^n$. Let $\mathbb{P}_{\mathbf{x}}$ denote the distribution of the random vector $\mathbf{x} \in \mathbb{R}^n$ (column by default). Let $p_{\mathbf{x}}(\boldsymbol{x})$ denote the probability density (resp. mass) function of $\mathbf{x}$ if $\mathbf{x}$ is continuous (resp. discrete). Whenever it is clear from contexts, we use $p(\boldsymbol{x})$ as a shorthand for $p_{\mathbf{x}}(\boldsymbol{x})$. The conditional distribution of $\mathbf{x}$ given $\mathbf{y} \in \mathbb{R}^m$ is denoted as $\mathbb{P}_{\mathbf{x}|\mathbf{y}}$. We use $p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})$ to denote the conditional probability density (or mass) function of $\mathbf{x}$ given $\mathbf{y} = \boldsymbol{y}$, shorted as $p(\boldsymbol{x}|\boldsymbol{y})$. Let $\mathbb{E}\mathbf{x}$ denote the expectation of $\mathbf{x}$ and $\mathbb{E}(\mathbf{x}|\mathbf{y})$ the conditional expectation of $\mathbf{x}$ given $\mathbf{y}$. The d-dimensional Gaussian distribution, parameterized by mean $\boldsymbol{c}$ and covariance $\boldsymbol{\Sigma}$, is denoted by $\mathcal{N}_d(\boldsymbol{c}, \boldsymbol{\Sigma})$ and the corresponding Gaussian density function is denoted by $\mathcal{N}_d(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{\Sigma})$. Given an integer N, the running index set is defined as $[N] := \{1, 2, \ldots, N\}$. Let $\mathcal{Y}_k$ denote the measurement sequence up to and including the time k, i.e., $\mathcal{Y}_k := (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k)$. A realization of $\mathcal{Y}_k$ is denoted as $\boldsymbol{Y}_k := (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k)$. Let $\boldsymbol{I}$ and $\mathbf{0}$ denote the identity and the zero matrices with appropriate dimensions, respectively. We use $\boldsymbol{M}^\top$ to denote the transpose of the matrix $\boldsymbol{M}$, and $\mathrm{Tr}\,[\boldsymbol{M}]$ its trace when

[1]However, the finite-energy assumption is reasonable for some state-feedback automatic control problems because some external disturbances are usually impulses or short-term perturbations. Besides, the mean-square-stability assumption is also reasonable for some state-feedback automatic control problems because systems can be stabilized by state feedback.

$\boldsymbol{M}$ is square. Let $\mathbb{S}^d$ denote the set of all d-dimensional symmetric matrices in $\mathbb{R}^{d\times d}$, and $\mathbb{S}^d_+$ (resp. $\mathbb{S}^d_{++}$) of all d-dimensional symmetric positive semi-definite (resp. positive definite) matrices in $\mathbb{S}^d$. If $\boldsymbol{A},\boldsymbol{B}\in\mathbb{S}^d$, $\boldsymbol{A}\succeq\boldsymbol{B}$ (resp. $\boldsymbol{A}\succ\boldsymbol{B}$) indicates that $\boldsymbol{A}-\boldsymbol{B}\in\mathbb{S}^d_+$ (resp. $\boldsymbol{A}-\boldsymbol{B}\in\mathbb{S}^d_{++}$). If $\boldsymbol{S}\in\mathbb{S}^d_+$, let $\boldsymbol{S}^{1/2}$ be a square root of $\boldsymbol{S}$ (i.e., $\boldsymbol{S}^{1/2}\boldsymbol{S}^{1/2}=\boldsymbol{S}$).

II. Preliminaries

A. Bayesian Estimation Subject to Multiple Models

Suppose the random vectors $\mathbf{x}$ and $\mathbf{y}$ have finite second moments, and the joint distribution of them is $\mathbb{P}_{\mathbf{x},\mathbf{y}}$. We are concerned with estimating the unobservable vector $\mathbf{x}$ based on the model $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ and the observation $\mathbf{y}$, in the minimum mean square error sense. In other words, we aim to find an estimator $\hat{\mathbf{x}}$ of $\mathbf{x}$, which is a function $\boldsymbol{\phi}$ of $\mathbf{y}$ [i.e., $\hat{\mathbf{x}}=\boldsymbol{\phi}(\mathbf{y})$], such that

$$\hat{\mathbf{x}}=\underset{\boldsymbol{\phi}\in\mathcal{H}_{\mathbf{y}}}{\operatorname{argmin}}\operatorname{Tr}\mathbb{E}[\mathbf{x}-\boldsymbol{\phi}(\mathbf{y})][\mathbf{x}-\boldsymbol{\phi}(\mathbf{y})]^\top, \tag{1}$$

where the expectation is taken over the joint distribution $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ and $\mathcal{H}_{\mathbf{y}}$ contains all possible estimators of $\mathbf{x}$ based on $\mathbf{y}$; for more information, see [23]. As is well-known, $\hat{\mathbf{x}}=\mathbb{E}(\mathbf{x}|\mathbf{y})$. Sometimes, we are not sure of the exact form of $\mathbb{P}_{\mathbf{x},\mathbf{y}}$. But we are confident that with probability ω_j, $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ is of the form $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ where ω_j's are weights and $\sum_{j=1}^N\omega_j=1$. In other words, $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ is a mixture of a set of distributions $\{\mathbb{P}_{j,\mathbf{x},\mathbf{y}}\}_{j=1,2,\ldots,N}$ with mixing probabilities $\{\omega_j\}_{j=1,2,\ldots,N}$, i.e., $\mathbb{P}_{\mathbf{x},\mathbf{y}}=\sum_{j=1}^N\omega_j\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$. As a result, given a measurement $\boldsymbol{y}$, the optimal estimate of $\mathbf{x}$ is

$$\hat{\boldsymbol{x}}=\textstyle\sum_{j=1}^N\mu_j\hat{\boldsymbol{x}}_j, \tag{2}$$

where $\hat{\boldsymbol{x}}_j:=\mathbb{E}(\mathbf{x}|\boldsymbol{y},j)=\int\boldsymbol{x}p_j(\boldsymbol{x}|\boldsymbol{y})\mathrm{d}\boldsymbol{x}$,

$$\mu_j:=p(j|\boldsymbol{y})=\frac{\omega_jp_j(\boldsymbol{y})}{p(\boldsymbol{y})}=\frac{\omega_jp_j(\boldsymbol{y})}{\sum_{j=1}^N\omega_jp_j(\boldsymbol{y})}, \tag{3}$$

$p_j(\boldsymbol{x}|\boldsymbol{y}):=p(\boldsymbol{x}|\boldsymbol{y},j)$ denotes the posterior distribution of $\mathbf{x}$ given $\boldsymbol{y}$ under the j^{th} model, and $p_j(\boldsymbol{y}):=p(\boldsymbol{y}|j)$ denotes the likelihood of the j^{th} model under the measurement $\boldsymbol{y}$. Hence, by Bayes's rule, $\mu_j\propto\omega_jp_j(\boldsymbol{y})$. In other words, ω_j can be understood as the prior model probability of the j^{th} model before observing $\boldsymbol{y}$, while μ_j can be seen as the posterior model probability of the j^{th} model after observing $\boldsymbol{y}$. The corresponding posterior error covariance $\mathbb{E}\big[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top\big|\boldsymbol{y}\big]$ conditioned on $\boldsymbol{y}$ equals to

$$\textstyle\sum_{j=1}^N\mu_j\big[\boldsymbol{P}_j+(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})^\top\big], \tag{4}$$

where $\boldsymbol{P}_j:=\mathbb{E}\big[(\mathbf{x}-\hat{\mathbf{x}}_j)(\mathbf{x}-\hat{\mathbf{x}}_j)^\top\big|j,\boldsymbol{y}\big]$ is the posterior error covariance of the j^{th} model given $\boldsymbol{y}$.

B. Optimal Bayesian Estimation

The following fact is well established in applied statistics.

Fact 1 ([23]): The posterior mean $\hat{\mathbf{x}}:=\mathbb{E}(\mathbf{x}|\mathbf{y})$ solves both

$$\min_{\boldsymbol{\phi}\in\mathcal{H}_{\mathbf{y}}}\operatorname{Tr}\mathbb{E}[\mathbf{x}-\boldsymbol{\phi}(\mathbf{y})][\mathbf{x}-\boldsymbol{\phi}(\mathbf{y})]^\top, \tag{5}$$

and

$$\min_{\boldsymbol{a}\in\mathbb{R}^n}\operatorname{Tr}\mathbb{E}\big\{[\mathbf{x}-\boldsymbol{a}][\mathbf{x}-\boldsymbol{a}]^\top\big|\mathbf{y}\big\},\quad\forall\mathbf{y}=\boldsymbol{y}, \tag{6}$$

where $\mathbf{x}\in\mathbb{R}^n$ and $\mathbf{y}\in\mathbb{R}^m$ are two random vectors; $\boldsymbol{y}$ is a possible realization of $\mathbf{y}$. Note that given the realization $\mathbf{y}=\boldsymbol{y}$, the real number $\boldsymbol{a}:=\boldsymbol{\phi}(\boldsymbol{y})$ is an ***estimate*** of $\mathbf{x}$. □

When $\mathbf{y}=\boldsymbol{y}$ is specified, the optimal estimate of $\mathbf{x}$ derived from (5) is $\hat{\boldsymbol{x}}=\hat{\mathbf{x}}(\mathbf{y}=\boldsymbol{y})=\mathbb{E}(\mathbf{x}|\mathbf{y}=\boldsymbol{y})$; i.e., the optimal estimate $\hat{\boldsymbol{x}}$ is specified by the optimal estimator $\hat{\mathbf{x}}$ with $\mathbf{y}$ being replaced with $\boldsymbol{y}$. However, the optimal estimate derived from (6) given $\mathbf{y}=\boldsymbol{y}$ is directly $\hat{\boldsymbol{x}}=\mathbb{E}(\mathbf{x}|\boldsymbol{y})$; i.e., there is no optimal estimator directly associated with (6). We are interested in (6) because it is not always convenient to solve (5). Hence, when we obtain a measurement $\boldsymbol{y}$, we may directly solve (6). It is for this reason that in Subsection II-A, we work on a specified $\boldsymbol{y}$ instead of the random vector $\mathbf{y}$. One may verify that, in Subsection II-A, deriving the closed-form expression of $\mathbb{E}\big[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top\big]$, i.e., $\mathbb{E}_{\mathbf{y}}\mathbb{E}_{\mathbf{x}|\mathbf{y}}\big[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top\big|\mathbf{y}\big]$ is extremely difficult. However, finding the closed-form expression of $\mathbb{E}\big[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^\top\big|\boldsymbol{y}\big]$ is relatively easy.

C. Distributional Balls

A distributional ball centered at the reference distribution $\bar{\mathbb{P}}_{\mathbf{x}}$ with radius $\theta\geq0$ is defined as $\mathcal{F}_{\mathbf{x}}(\theta):=\{\mathbb{P}_{\mathbf{x}}|\Delta(\mathbb{P}_{\mathbf{x}},\bar{\mathbb{P}}_{\mathbf{x}})\leq\theta\}$, where $\Delta(\mathbb{P}_{\mathbf{x}},\bar{\mathbb{P}}_{\mathbf{x}})$ is a proper statistical similarity measure (e.g., Wasserstein distance, Kullback–Leibler divergence, moment-based methods) between two distributions $\mathbb{P}_{\mathbf{x}}$ and $\bar{\mathbb{P}}_{\mathbf{x}}$ [13], [14]; θ is the size parameter of $\mathcal{F}_{\mathbf{x}}(\theta)$. Namely, a distributional ball is a collection of probability distributions that are close to $\bar{\mathbb{P}}_{\mathbf{x}}$. If Δ is specified by a moment-based method [13, Eq. (21)], then we need more than one parameter to define the size of a distributional ball; in this case, the distributional ball is denoted as $\mathcal{F}_{\mathbf{x}}(\boldsymbol{\theta})$ where a vector $\boldsymbol{\theta}$ (called size parameter vector) is involved. Without loss of generality, this paper uses $\mathcal{F}_{\mathbf{x}}(\boldsymbol{\theta})$ to denote a generic distributional ball.

III. Problem Formulation

We aim to estimate the unknown state $\mathbf{x}_k$ of a jump linear Markov system

$$\begin{cases}\mathbf{x}_k&=\boldsymbol{F}_{j,k-1}\mathbf{x}_{k-1}+\boldsymbol{G}_{j,k-1}\mathbf{w}_{j,k-1},\quad j=1,2,\ldots,N,\\ \mathbf{y}_k&=\boldsymbol{H}_{j,k}\mathbf{x}_k+\mathbf{v}_{j,k},\end{cases} \tag{7}$$

where k denotes the discrete time index; N is the size of the nominal model set; $\mathbf{x}_k\in\mathbb{R}^n$ is the state vector; $\mathbf{y}_k\in\mathbb{R}^m$ is the measurement vector; $\mathbf{w}_{j,k-1}\in\mathbb{R}^p$, $\mathbf{v}_{j,k}\in\mathbb{R}^m$ are the process noise and measurement noise of the j^{th} model, respectively. Typically, for every nominal linear system in the model set (i.e., for every $j=1,2,\ldots,N$), the following properties are assumed to be satisfied [24]–[26]: 1) for all k, $\mathbf{x}_k$, $\mathbf{y}_k$, $\mathbf{w}_{j,k}$, and $\mathbf{v}_{j,k}$ have finite second moments; 2) $\mathbf{x}_0\sim\mathcal{N}_n(\bar{\boldsymbol{x}}_0,\boldsymbol{M}_0)$, and for all k, $\mathbf{w}_{j,k}\sim\mathcal{N}_p(\boldsymbol{\mu}^{\boldsymbol{w}}_{j,k},\boldsymbol{Q}_{j,k})$ and $\mathbf{v}_{j,k}\sim\mathcal{N}_m(\boldsymbol{\mu}^{\boldsymbol{v}}_{j,k},\boldsymbol{R}_{j,k})$; 3) for any $k_1\neq k_2$, $\mathbf{w}_{j,k_1}$ and $\mathbf{x}_0$ are uncorrelated, so are $\mathbf{v}_{j,k_1}$ and $\mathbf{x}_0$, $\mathbf{w}_{j,k_1}$ and $\mathbf{w}_{j,k_2}$, and $\mathbf{v}_{j,k_1}$ and $\mathbf{v}_{j,k_2}$; for any k_1,k_2, $\mathbf{v}_{j,k_1}$ and $\mathbf{w}_{j,k_2}$ are uncorrelated; 4) the involved parameters $\bar{\boldsymbol{x}}_0$, $\boldsymbol{M}_0$, $\boldsymbol{\mu}^{\boldsymbol{w}}_{j,k}$, $\boldsymbol{\mu}^{\boldsymbol{v}}_{j,k}$, $\boldsymbol{Q}_{j,k}$, $\boldsymbol{R}_{j,k}$, $\boldsymbol{F}_{j,k-1}$, $\boldsymbol{G}_{j,k-1}$, and $\boldsymbol{H}_{j,k}$ are exactly known, and typically $\boldsymbol{\mu}^{\boldsymbol{w}}_{j,k}$ and $\boldsymbol{\mu}^{\boldsymbol{v}}_{j,k}$ are the zero vectors. Note that for a true operating jump system, at every time k, the true dominating working mode j is unknown.

The jump linear system (7) is called a hybrid linear system because we can treat the operating but unknown mode j_k at the time k as a discrete system state so that the augmented state vector $\{\mathrm{j}_k, \mathbf{x}_k\}$ is a hybrid state vector consisting of both a discrete state variable and a continuous state vector. Usually, the evolution of the discrete state j_k (i.e., the model transition process) is modeled by a N-state homogeneous Markov chain [3], and the model transition probability matrix (TPM) is $\mathbf{\Pi} := \{\pi_{ij}\}_{i,j=1,2,\ldots,N}$ where π_{ij} denotes the probability that the system's operating mode jumps from the i^{th} model at the time $k-1$ to the j^{th} model at the time k. From the viewpoint of Bayesian statistical signal processing, the state estimation problem for jump linear systems can be stated as finding the posterior (or filtered) state distribution of $\{\mathrm{j}_k, \mathbf{x}_k\}$, i.e., $p(j_k, \boldsymbol{x}_k|\boldsymbol{Y}_k)$, at the time k, based on the system model (7), the TPM $\mathbf{\Pi}$, and the past measurements $\boldsymbol{Y}_k := (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k)$.

The nominal system (7) defines two discrete-time stochastic processes $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$, $k = 1, 2, \cdots$. Suppose the nominal joint state-measurement distribution defined by the nominal system model (7) is $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathcal{Y}_k}$. We would like to solve the following optimization problem

$$\min_{\boldsymbol{\phi}\in\mathcal{H}_{\mathcal{Y}_k}} \operatorname{Tr}\mathbb{E}[\mathbf{x}_k - \boldsymbol{\phi}(\mathcal{Y}_k)][\mathbf{x}_k - \boldsymbol{\phi}(\mathcal{Y}_k)]^\top, \tag{8}$$

where the expectation is taken over the joint distribution $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathcal{Y}_k}$; $\mathcal{H}_{\mathcal{Y}_k}$ contains all possible estimators of $\mathbf{x}_k$ based on the measurement set $\mathcal{Y}_k$; $\boldsymbol{\phi}$ is called an estimator and the one solving (8) is the optimal estimator. One may easily verify by contradiction that the optimal solution to (8) is unique. The optimal estimator of $\mathbf{x}_k$ in this minimum mean square error sense is $\mathbb{E}(\mathbf{x}_k|\mathcal{Y}_k) \in \mathcal{H}_{\mathcal{Y}_k}$. Since $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathcal{Y}_k}$ is not Gaussian due to the multi-mode property of (7), $\mathbb{E}(\mathbf{x}_k|\mathcal{Y}_k)$ cannot be of a linear form because a Gaussian mixture is no longer Gaussian; cf. (2) and (3) where $\hat{\boldsymbol{x}}$ is no longer linear in $\boldsymbol{y}$.

If the actual system dynamics deviates from the nominal model (7), the true joint state-measurement distribution $\mathbb{P}_{\mathbf{x}_k,\mathcal{Y}_k}$ will deviate from the nominal $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathcal{Y}_k}$. As a result, a robust state estimation solution that is insensitive to the model deviations needs to be designed. The distributionally robust counterpart of (8) can be written as

$$\min_{\boldsymbol{\phi}\in\mathcal{H}_{\mathcal{Y}_k}} \max_{\mathbb{P}\in\mathcal{F}_{\mathbf{x}_k,\mathcal{Y}_k}(\boldsymbol{\theta})} \operatorname{Tr}\mathbb{E}[\mathbf{x}_k - \boldsymbol{\phi}(\mathcal{Y}_k)][\mathbf{x}_k - \boldsymbol{\phi}(\mathcal{Y}_k)]^\top, \tag{9}$$

where the expectation is taken over $\mathbb{P}_{\mathbf{x}_k,\mathcal{Y}_k}$ and $\mathcal{F}_{\mathbf{x}_k,\mathcal{Y}_k}(\boldsymbol{\theta})$ is the associated **ambiguity set** consisting of all possible joint distributions $\mathbb{P}_{\mathbf{x}_k,\mathcal{Y}_k}$ that lie in a distributional ball centered at the nominal distribution $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathcal{Y}_k}$ with size parameter vector $\boldsymbol{\theta}$.

Since state estimation problems are real-time problems: the measurements $\mathbf{y}_k$ arrives sequentially and the optimal estimator operates along the discrete time in a recursive way [27], we instead solve a time-incremental [28] problem

$$\min_{\boldsymbol{\phi}\in\mathcal{H}_{\mathbf{y}_k}} \max_{\mathbb{P}\in\mathcal{F}_{\mathbf{x}_k,\mathbf{y}_k|\mathcal{Y}_{k-1}}(\boldsymbol{\theta})} \operatorname{Tr}\mathbb{E}\big\{[\mathbf{x}_k - \boldsymbol{\phi}(\mathbf{y}_k)][\mathbf{x}_k - \boldsymbol{\phi}(\mathbf{y}_k)]^\top|\mathcal{Y}_{k-1}\big\}, \tag{10}$$

where the expectation is taken over $\mathbb{P}_{\mathbf{x}_k,\mathbf{y}_k|\mathcal{Y}_{k-1}}$ and the ambiguity set $\mathcal{F}_{\mathbf{x}_k,\mathbf{y}_k|\mathcal{Y}_{k-1}}(\boldsymbol{\theta})$ is constructed around $\bar{\mathbb{P}}_{\mathbf{x}_k,\mathbf{y}_k|\mathcal{Y}_{k-1}}$, i.e., the nominal conditional joint state-measurement distribution given the previous measurement sequence. Note that in (10), the space of $\boldsymbol{\phi}$ is only defined by $\mathbf{y}_k$ instead of $\mathcal{Y}_k$. To solve (10), we need to first design proper forms for $\mathcal{F}_{\mathbf{x}_k,\mathbf{y}_k|\mathcal{Y}_{k-1}}(\boldsymbol{\theta})$, and then find the explicit optimization equivalent(s) of (10) so that it can be efficiently solved.

Therefore, at each time step k, we are inspired to **first** study a distributionally robust Bayesian estimation problem with multiple nominal models

$$\min_{\boldsymbol{\phi}\in\mathcal{H}_{\mathbf{y}}} \max_{\mathbb{P}\in\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})} \operatorname{Tr}\mathbb{E}[\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})][\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})]^\top \tag{11}$$

subject to the multiple nominal joint state-measurement distributions $\{\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}\}_{j\in[N]}$, the nominal prior model probabilities $\{\bar{\omega}_j\}_{j\in[N]}$, and a properly constructed ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ that is characterized by $\{\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}\}_{j\in[N]}$ and $\{\bar{\omega}_j\}_{j\in[N]}$. The subscript k (i.e., discrete time index) is dropped to avoid notational clutter. **Then**, by identifying the joint distribution of $(\mathbf{x}_k, \mathbf{y}_k)$ conditioned on $\mathcal{Y}_{k-1}$, we can solve (10).

Remark 1: Model (11) is called the distributionally robust counterpart of (1). Note that robust counterparts of the batch and the recursive formulations are not equivalent: The former (9) is called robust filtering without commitment while the latter (10) is called robust filtering under commitment [28]. However, the robust counterpart for the recursive formulation outperforms that for the batch formulation because the latter tends to consume all the prescribed robustness budget at the first several time steps (i.e., as the time proceeds, the robust batch formulation would reduce to the usual non-robust version); see [29, Appendix B]. □

IV. Distributionally Robust Bayesian Estimation Subject to Multiple Nominal Models

In this section, we first design a proper ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ that is compatible with the research aims raised in Subsection I-C, and then solve the distributionally robust Bayesian estimation problem (11). Because we desire the flexibility that responds to specified types of uncertainty, in order to take into account the individual model uncertainties in every nominal candidate model and/or the uncertainties in nominal model weights,[2] a suitable ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ can be constructed in (12), shown at the top of the next page, where $\boldsymbol{\theta} := [\theta_0, \theta_1, \theta_2, \ldots, \theta_N]^\top$, $\mathcal{M}(\mathbb{R}^d)$ denotes all probability distributions on $\mathbb{R}^d$, and $\mathcal{P}(\mathbb{R}^N)$ denotes all N-length discrete distributions; $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ represents the j^{th} nominal joint state-measurement distribution; $\bar{\boldsymbol{\omega}} := (\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_N)^\top$ and $\bar{\omega}_j$ denotes the nominal prior model probability of the j^{th} model; for all $j \in [N]$, $\Delta_j(\cdot,\cdot)$ represents the statistical similarity measure between the possibly true joint state-measurement distribution $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ and its nominal distribution $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$, and $\Delta_0(\cdot,\cdot)$ the statistical similarity measure between the possibly true prior model probability $\boldsymbol{\omega}$ and its nominal value $\bar{\boldsymbol{\omega}}$; θ_0 and θ_j are the radii of the distributional sets defined by $\Delta_0(\cdot,\cdot)$ and $\Delta_j(\cdot,\cdot)$'s, respectively. The ambiguity set in (12) inherently allows the flexibility of responding to specified types of uncertainty. For example, when we set θ_0 to be non-zero and all other θs (from θ_1 to θ_N) to be zeros, we consider only the uncertainties in TPMs (and therefore in model weights), and all candidate models are believed to be

[2] N.B.: Uncertainties in TPMs lead to uncertainties in model weights.

$$\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta}) = \left\{ \mathbb{P}_{\mathbf{x},\mathbf{y}} = \sum_{j=1}^{N} \omega_j \mathbb{P}_{j,\mathbf{x},\mathbf{y}} \;\middle|\; \begin{array}{ll} \mathbb{P}_{j,\mathbf{x},\mathbf{y}} \in \mathcal{M}(\mathbb{R}^n \times \mathbb{R}^m), & \forall j \in [N] \\ \boldsymbol{\omega} \in \mathcal{P}(\mathbb{R}^N) & \\ \Delta_j(\mathbb{P}_{j,\mathbf{x},\mathbf{y}}, \bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}) \le \theta_j, & \forall j \in [N] \\ \Delta_0(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) \le \theta_0 & \end{array} \right\}, \tag{12}$$

exact. For another example, supposing $N = 2$, if we set θ_1 to be non-zero and θ_0 and θ_2 to be zeros, we consider only the model uncertainties in the first nominal candidate model, and the TPM and the second candidate model are believed to be exact. Note that every element in $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ is a mixture distribution, including the least-favorable distribution that solves the inner maximization problem in (11). If $N = 1$, we have $\omega_1 = \bar{\omega}_1 = 1$ and $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ reduces to

$$\mathcal{F}_{\mathbf{x},\mathbf{y}}(\theta) = \left\{ \mathbb{P}_{\mathbf{x},\mathbf{y}} \in \mathcal{M}(\mathbb{R}^n \times \mathbb{R}^m) \,\middle|\, \Delta(\mathbb{P}_{\mathbf{x},\mathbf{y}}, \bar{\mathbb{P}}_{\mathbf{x},\mathbf{y}}) \le \theta \right\},$$

which is a special case discussed in [13].

Next, we find the tractable reformulation(s) of the distributionally robust Bayesian estimation problem (11) subject to the multi-model ambiguity set (12).

Consider the max-min problem induced by (11):

$$\max_{\mathbb{P} \in \mathcal{F}_{\mathbf{x},\mathbf{y}}(\theta)} \min_{\boldsymbol{\phi} \in \mathcal{H}_{\mathbf{y}}} \operatorname{Tr} \mathbb{E}[\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})][\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})]^\top. \tag{13}$$

Lemma 1: The min-max problem (11) and the max-min problem (13) is equivalent. To be specific, the solution solving (13) also solves (11), and vice versa. To clarify further, letting $V(\boldsymbol{\phi}, \mathbb{P}) := \operatorname{Tr} \mathbb{E}[\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})][\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})]^\top$, and supposing $\boldsymbol{\phi}^*$ and $\mathbb{P}^*$ solve the max-min problem (13), then $(\boldsymbol{\phi}^*, \mathbb{P}^*)$ forms a saddle point of the objective function $V(\boldsymbol{\phi}, \mathbb{P})$, i.e.,

$$\min_{\boldsymbol{\phi} \in \mathcal{H}_{\mathbf{y}}} V(\boldsymbol{\phi}, \mathbb{P}^*) = V(\boldsymbol{\phi}^*, \mathbb{P}^*) = \max_{\mathbb{P} \in \mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})} V(\boldsymbol{\phi}^*, \mathbb{P}). \tag{14}$$

Proof: The Gaussianity assumption in [14, Thm. 7] and [30, Thm. 9] can be dropped without changing the statements therein because, for any type of joint distribution $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ (not limited to a Gaussian), the unique optimal state estimator in the minimum mean square error sense is the posterior mean $\mathbb{E}(\mathbf{x}|\mathbf{y})$. As a result, this lemma is immediate following the proofs of [14, Thm. 7] and [30, Thm. 9]. □

The max-min problem (13) is easier to solve than the original min-max problem (11) because for every $\mathbb{P} \in \mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, we can find the unique associated optimal estimator in $\mathcal{H}_{\mathbf{y}}$. For every $j \in [N]$, supposing the nominal joint state-noise distribution is $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{v}_j} := \mathcal{N}_{n+m}\left(\begin{bmatrix} \bar{\boldsymbol{x}}_j \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{M}_j & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}_j \end{bmatrix} \right)$, we have the nominal joint state-measurement distribution

$$\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}} := \mathcal{N}_{n+m}\left(\begin{bmatrix} \bar{\boldsymbol{x}}_j \\ \boldsymbol{H}_j \bar{\boldsymbol{x}}_j \end{bmatrix}, \begin{bmatrix} \boldsymbol{M}_j & \boldsymbol{M}_j \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{M}_j & \boldsymbol{H}_j \boldsymbol{M}_j \boldsymbol{H}_j^\top + \boldsymbol{R}_j \end{bmatrix} \right), \tag{15}$$

which is induced by the linear observation system $\mathbf{y} = \boldsymbol{H}_j \mathbf{x} + \mathbf{v}_j$. Accordingly, given the measurement $\boldsymbol{y}$, we can obtain the nominal state estimate $\overline{\hat{\boldsymbol{x}}_j}$ and the nominal estimation error covariance $\overline{\boldsymbol{P}_j}$; see Appendix A. For every candidate distribution $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ in $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, we assume that its component distributions $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}, \forall j \in [N]$ are also Gaussian. If the possible joint state-noise distribution is $\mathbb{P}_{j,\mathbf{x},\mathbf{v}_j} := \mathcal{N}_{n+m}\left(\begin{bmatrix} \boldsymbol{c}_{j,\boldsymbol{x}} \\ \boldsymbol{c}_{j,\boldsymbol{v}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{bmatrix} \right)$, we can obtain the joint state-measurement distribution

$$\mathbb{P}_{j,\mathbf{x},\mathbf{y}} := \mathcal{N}_{n+m}(\boldsymbol{c}_j, \boldsymbol{\Sigma}_j) \tag{16}$$

where $\boldsymbol{\Sigma}_j := \begin{bmatrix} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} & \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} & \boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{bmatrix}$ and $\boldsymbol{c}_j := \begin{bmatrix} \boldsymbol{c}_{j,\boldsymbol{x}} \\ \boldsymbol{H}_j \boldsymbol{c}_{j,\boldsymbol{x}} + \boldsymbol{c}_{j,\boldsymbol{v}} \end{bmatrix}$. Accordingly, given the measurement $\boldsymbol{y}$, the state estimate $\hat{\boldsymbol{x}}_j$ and the estimation error covariance $\boldsymbol{P}_j$ associated with $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ can be obtained; see Appendix A.

From (2) and (4), for a possible mixture distribution $\mathbb{P}_{\mathbf{x},\mathbf{y}} \in \mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, if $\mathbf{y} = \boldsymbol{y}$ is specified, we have the associated optimal posterior estimate

$$\hat{\boldsymbol{x}} = \sum_{j=1}^{N} \mu_j \cdot \hat{\boldsymbol{x}}_j, \tag{17}$$

and the corresponding estimation error covariance

$$\boldsymbol{P} = \sum_{j=1}^{N} \mu_j \cdot \left\{ \boldsymbol{P}_j + (\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}})^\top \right\}, \tag{18}$$

where $\hat{\boldsymbol{x}}_j$, $\boldsymbol{P}_j$, and μ_j are the optimal posterior estimate of the state $\mathbf{x}$, the posterior error covariance, and the posterior model probability, corresponding to the j^{th} nominal model, respectively. As explained in Subsection II-B, to simplify the problem-solving procedure and without loss of optimality, we work directly on the case where $\boldsymbol{y}$ has been specified; i.e., (6).

Consequently, the explicit optimization equivalent of the distributionally robust Bayesian estimation problem (11) subject to the ambiguity set (12) can be given below.

Proposition 1: Consider the distributionally robust Bayesian estimation problem (11). Suppose that

1) The ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, which is a collection of mixture distributions, is defined in (12);
2) Every candidate distribution inside $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ is a N-component Gaussian mixture, and each Gaussian component is defined as (16), for every $j \in [N]$;
3) The nominal distribution, which is the center of $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, is a N-component Gaussian mixture, and each Gaussian component is defined as (15), for every $j \in [N]$;
4) A measurement $\mathbf{y} = \boldsymbol{y}$ is specified.

Then (11) can be explicitly reformulated into

$$\max_{\boldsymbol{\Theta}} \operatorname{Tr} \sum_{j=1}^{N} \mu_j \cdot \left\{ \boldsymbol{P}_j + (\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}})^\top \right\}, \tag{19}$$

where $\boldsymbol{\Theta} := \{\omega_j, \boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j \in [N]}$; the feasible region of $\boldsymbol{\Theta}$ is implicitly defined by $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ because every $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ in $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ is parameterized by $\boldsymbol{\Theta}$. Note that for every $j \in [N]$, μ_j is defined by ω_j through (3), and $\hat{\boldsymbol{x}}_j$ and $\boldsymbol{P}_j$ are defined by $\boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}$, and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$ (cf. Appendix A), respectively; $\hat{\boldsymbol{x}}$ is defined in (17).

Proof: See Appendix B. □

Compared with the original distributionally robust Bayesian estimation problem (11), which is extremely abstract, the reformulated optimization equivalent (19) is explicit and specific.

Hence, when the ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ is designed, (19) can be explicitly solved over $\boldsymbol{\Theta}$. However, (19) is still a difficult problem because the objective is highly nonlinear in $\boldsymbol{\Theta}$.

For the convenience of later analysis, we give a compact reformulation of (19).

Proposition 2: Problem (19) can be compactly rewritten as

$$\max_{\boldsymbol{\Theta}} -\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu}, \tag{20}$$

where

$$\boldsymbol{A} := \begin{bmatrix} \hat{\boldsymbol{x}}_1^\top \hat{\boldsymbol{x}}_1 & \hat{\boldsymbol{x}}_1^\top \hat{\boldsymbol{x}}_2 & \hat{\boldsymbol{x}}_1^\top \hat{\boldsymbol{x}}_3 & \cdots & \hat{\boldsymbol{x}}_1^\top \hat{\boldsymbol{x}}_N \\ \hat{\boldsymbol{x}}_2^\top \hat{\boldsymbol{x}}_1 & \hat{\boldsymbol{x}}_2^\top \hat{\boldsymbol{x}}_2 & \hat{\boldsymbol{x}}_2^\top \hat{\boldsymbol{x}}_3 & \cdots & \hat{\boldsymbol{x}}_2^\top \hat{\boldsymbol{x}}_N \\ \hat{\boldsymbol{x}}_3^\top \hat{\boldsymbol{x}}_1 & \hat{\boldsymbol{x}}_3^\top \hat{\boldsymbol{x}}_2 & \hat{\boldsymbol{x}}_3^\top \hat{\boldsymbol{x}}_3 & \cdots & \hat{\boldsymbol{x}}_3^\top \hat{\boldsymbol{x}}_N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{x}}_N^\top \hat{\boldsymbol{x}}_1 & \hat{\boldsymbol{x}}_N^\top \hat{\boldsymbol{x}}_2 & \hat{\boldsymbol{x}}_N^\top \hat{\boldsymbol{x}}_3 & \cdots & \hat{\boldsymbol{x}}_N^\top \hat{\boldsymbol{x}}_N \end{bmatrix}, \tag{21}$$

and

$$\boldsymbol{b} := \begin{bmatrix} \operatorname{Tr}[\boldsymbol{P}_1] + \hat{\boldsymbol{x}}_1^\top \hat{\boldsymbol{x}}_1 \\ \operatorname{Tr}[\boldsymbol{P}_2] + \hat{\boldsymbol{x}}_2^\top \hat{\boldsymbol{x}}_2 \\ \vdots \\ \operatorname{Tr}[\boldsymbol{P}_N] + \hat{\boldsymbol{x}}_N^\top \hat{\boldsymbol{x}}_N \end{bmatrix}. \tag{22}$$

Since $\boldsymbol{A} \succeq \mathbf{0}$, the objective of (20) is concave in $\boldsymbol{\mu}$.

Proof: The objective of (19) can be rearranged into $-\operatorname{Tr}[(\sum_{j=1}^N \mu_j \cdot \hat{\boldsymbol{x}}_j)(\sum_{j=1}^N \mu_j \cdot \hat{\boldsymbol{x}}_j)^\top] + \sum_{j=1}^N \mu_j \cdot \operatorname{Tr}[\boldsymbol{P}_j + \hat{\boldsymbol{x}}_j \hat{\boldsymbol{x}}_j^\top]$. By defining $\boldsymbol{A}$, $\boldsymbol{b}$, we have (20). □

The reformulation from (19) to (20) is important for later analysis because the objective of the former is cubic in $\boldsymbol{\mu}$ but that of the latter is quadratic in $\boldsymbol{\mu}$. Cubic programming is difficult because convexity (or concavity) is not naturally implied. However, generally speaking, (20) is still complicated to be solved because $\boldsymbol{\mu}$, $\boldsymbol{A}$, and $\boldsymbol{b}$ are all defined by the decision variables $\boldsymbol{\Theta}$ in a highly nonlinear manner.

In the following, we explicitly solve (19) or (20), whichever is easier under investigated conditions.

A. When $\theta_0 \neq 0$ But $\theta_j = 0, \forall j \in [N]$

We first study a special case where $\theta_j = 0, \forall j = 1, 2, \ldots, N$. Namely, we assume that the nominal model set is correct and able to exactly describe every true dynamics of the actual system. However, the prior model probabilities ω_j's are uncertain due, e.g., to the uncertain model transition probability matrix, to the uncertain initial model probability at the time $k = 0$, or to the method uncertainty (of, e.g., the IMM filter) in approximating posterior state distributions.

When $\theta_j = 0, \forall j \in [N]$, $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ in (12) reduces to

$$\mathcal{F}_{\mathbf{x},\mathbf{y}}(\theta_0) = \left\{ \mathbb{P}_{\mathbf{x},\mathbf{y}} = \textstyle\sum_{j=1}^N \omega_j \bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}} \;\middle|\; \begin{array}{l} \boldsymbol{\omega} \in \mathcal{P}(\mathbb{R}^N) \\ \Delta_0(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) \leq \theta_0 \end{array} \right\}. \tag{23}$$

It means that the nominal model weight vector $\boldsymbol{\omega}$ is uncertain, but the nominal model set is exact. Since only $\boldsymbol{\mu}$ is related to $\boldsymbol{\omega}$, and $\boldsymbol{A}$ and $\boldsymbol{b}$ are not related to $\boldsymbol{\omega}$, in this subsection, we investigate (20) instead of (19). The distributionally robust Bayesian estimation problem (20) can be rewritten as

$$\begin{array}{ll} \max\limits_{\boldsymbol{\omega}} & -\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu} \\ \textit{s.t.} & \left\{ \begin{array}{ll} \sum_{j=1}^N \omega_j = 1, & \\ \omega_j \geq 0, & \forall j \in [N], \\ \Delta_0(\boldsymbol{\omega}, \bar{\boldsymbol{\omega}}) \leq \theta_0, & \end{array} \right. \end{array} \tag{24}$$

where $\boldsymbol{\Theta}$ reduces to $\boldsymbol{\omega}$; the relation between the prior model probability ω_j and the posterior model probability μ_j, for every model j, is established in (3); $\boldsymbol{A}$ and $\boldsymbol{b}$ are defined in (21) and (22), respectively, with $\hat{\boldsymbol{x}}_j$ being replaced with $\overline{\hat{\boldsymbol{x}}_j}$ and $\boldsymbol{P}_j$ being replaced with $\overline{\boldsymbol{P}_j}$ (cf. Appendix A). Note that in this case, $\mathbb{P}_{j,\mathbf{x},\mathbf{y}} = \bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}, \forall j \in [N]$.

Although the objective of (24) is quadratic in $\boldsymbol{\mu}$ and both $\boldsymbol{A}$ and $\boldsymbol{b}$ are fixed, (24) is not a quadratic program because the decision vector is $\boldsymbol{\omega}$ instead of $\boldsymbol{\mu}$, and the relation between $\boldsymbol{\omega}$ and $\boldsymbol{\mu}$ is highly nonlinear; see (3). Since uncertain prior model probabilities lead to uncertain posterior model probabilities, to simplify the problem, we propose to solve (24) over the posterior model probability $\boldsymbol{\mu}$ rather than explicitly over the prior model probability $\boldsymbol{\omega}$.[3] We have

$$\begin{array}{ll} \max\limits_{\boldsymbol{\mu}} & -\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu} \\ \textit{s.t.} & \left\{ \begin{array}{ll} \sum_{j=1}^N \mu_j = 1 & \\ \mu_j \geq 0, & \forall j \in [N], \\ \Delta_0(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}) \leq \theta_0. & \end{array} \right. \end{array} \tag{25}$$

Compared to the model (24), the model (25) is not only easier to be solved but also suitable to capture model uncertainties of the nominal model set because uncertain models give inexact model likelihood evaluations; cf. Step 1.6 in Algorithm 3 in the online supplementary materials. Since uncertainties in the posterior model probabilities may be caused by many factors (e.g., uncertain model transition probability matrix, uncertainties in candidate models), even though the state estimator for jump linear system (7) is robustified by taking into consideration only $\boldsymbol{\mu}$, the robust estimator is still likely to be insensitive to all the causing factors.

To measure the closeness between two discrete distributions $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$ (i.e., to quantify uncertainties in $\boldsymbol{\mu}$), many statistical similarity measures can be used, for example, the popular Kullback–Leibler divergence and Wasserstein distance [30]; justifications and comparisons of using the two exemplified measures can be seen in, e.g., [31]–[36].

We use the Kullback-Leibler (KL) divergence to define $\Delta_0(\cdot,\cdot)$; the case where the Wasserstein distance is employed to define $\Delta_0(\cdot,\cdot)$ is discussed in Appendix C. The problem (25) is particularized to

$$\begin{array}{ll} \max\limits_{\boldsymbol{\mu}} & -\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu} \\ \textit{s.t.} & \left\{ \begin{array}{l} \mathbf{1}^\top \boldsymbol{\mu} = 1 \\ \boldsymbol{\mu}^\top \ln \boldsymbol{\mu} - \boldsymbol{\mu}^\top \ln \bar{\boldsymbol{\mu}} \leq \theta_0, \end{array} \right. \end{array} \tag{26}$$

where $\ln \boldsymbol{\mu} := (\ln \mu_1, \ln \mu_2, \ldots, \ln \mu_N)^\top$. Since the objective is concave and the constraints are convex, the solution of (26) can be readily obtained using the Lagrangian duality theory (particularly, the Karush-Kuhn-Tucker conditions) [37]. For details, see Appendix I in the online supplementary materials. Note that (26) may have multiple optimal solutions. Note also that $\boldsymbol{\mu} \geq \mathbf{0}$ is a redundant constraint because the function $\ln \boldsymbol{\mu}$ implicitly requires it. We computationally prefer the KL divergence for $\Delta_0(\cdot,\cdot)$ because the resulting solution method is computationally more efficient than that under the Wasserstein distance; for details, see Appendix C.

[3] For an extensive discussion, see Appendix H in the online supplementary materials.

B. When $\theta_0 \neq 0$ And $\theta_j \neq 0, \forall j \in [N]$

In this subsection, we discuss the case that both the nominal model set and the prior model probabilities are uncertain; i.e., the generic ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ in (12) is considered.

Proposition 3: Suppose both the model set and the prior model probability $\boldsymbol{\omega}$ are uncertain. If we conduct robustification over the posterior model probability $\boldsymbol{\mu}$ instead of the prior model probability $\boldsymbol{\omega}$, the reformulated distributionally robust Bayesian estimation problem (19) can be written as

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Theta}'} & \operatorname{Tr}\sum_{j=1}^{N}\mu_j\cdot\left\{\boldsymbol{P}_j+(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})^\top\right\} \\ \textit{s.t.} & \left\{\begin{array}{l}\Delta_j(\mathbb{P}_{j,\mathbf{x},\mathbf{y}},\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}})\leq\theta_j,\quad \forall j\in[N]\\ \Delta_0(\boldsymbol{\mu},\bar{\boldsymbol{\mu}})\leq\theta_0,\quad \mathbf{1}^\top\boldsymbol{\mu}=1,\quad \boldsymbol{\mu}\geq\mathbf{0},\end{array}\right. \end{array} \tag{27}$$

where $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{x}}_j$, and $\boldsymbol{P}_j$ are defined in (17) and Appendix A, respectively; $\boldsymbol{\Theta}' := \{\mu_j, \boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$. Recall that $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ is parameterized by $\boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}$, and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$, while $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ is parameterized by $\bar{\boldsymbol{x}}_j, \boldsymbol{M}_j$, and $\boldsymbol{R}_j$; see (16) and (15), respectively.

Proof: This is resulted from the definition of the ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$ in (12). □

Note that $\boldsymbol{\Theta}'$ in (27) is different from $\boldsymbol{\Theta} := \{\omega_j, \boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$ in (19). We optimize (27) over $\boldsymbol{\Theta}'$ instead of $\boldsymbol{\Theta}$ to reduce the complexity of the problem; however, one can also optimize (27) over $\boldsymbol{\Theta}$ for the fidelity of the problem setting; see Appendix H in the online supplementary materials.

Due to the complexity of the definitions of $\hat{\boldsymbol{x}}$ in (17) and $\hat{\boldsymbol{x}}_j$ and $\boldsymbol{P}_j$ in Appendix A, Problem (27) is difficult to solve: there exist strong nonlinearities and involve many decision variables. Hence, we simplify (27) by making the following assumption.

Assumption 1: Suppose $\boldsymbol{c}_{j,\boldsymbol{x}} = \bar{\boldsymbol{x}}_j$, $\boldsymbol{c}_{j,\boldsymbol{v}} = \mathbf{0}$, and

$$\begin{aligned}&\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top+\boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1}\\ &\qquad=\boldsymbol{M}_j\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{M}_j\boldsymbol{H}_j^\top+\boldsymbol{R}_j)^{-1},\end{aligned}$$

for every $j \in [N]$. □

Intuitively, when we conduct optimization in (27) by changing $\{\mu_j, \boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, Assumption 1 requires that the value of $\hat{\boldsymbol{x}}_j$ remains unchanged and equal to its nominal value defined by $\widehat{\bar{\boldsymbol{x}}_j}$ (cf. Appendix A) all the time. However, the value of $\boldsymbol{P}_j$, which is a function of variables $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$, would change. This is one of the practical yet promising tricks to shrink (i.e., limit the size of) the feasible region of (27).

By Assumption 1, the problem (27) can be written as

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Theta}'} & \operatorname{Tr}\sum_{j=1}^{N}\mu_j\cdot\left\{\boldsymbol{P}_j+(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})^\top\right\} \\ \textit{s.t.} & \left\{\begin{array}{ll}\boldsymbol{c}_{j,\boldsymbol{x}}=\bar{\boldsymbol{x}}_j,\quad \boldsymbol{c}_{j,\boldsymbol{v}}=\mathbf{0}, & \forall j\in[N]\\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top=\boldsymbol{K}_j\cdot(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top+\boldsymbol{\Sigma}_{j,\boldsymbol{v}}), & \forall j\in[N]\\ \Delta_j(\mathbb{P}_{j,\mathbf{x},\mathbf{y}},\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}})\leq\theta_j, & \forall j\in[N]\\ \Delta_0(\boldsymbol{\mu},\bar{\boldsymbol{\mu}})\leq\theta_0,\quad \mathbf{1}^\top\boldsymbol{\mu}=1,\quad \boldsymbol{\mu}\geq\mathbf{0},\end{array}\right. \end{array} \tag{28}$$

where

$$\boldsymbol{K}_j := \boldsymbol{M}_j\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{M}_j\boldsymbol{H}_j^\top+\boldsymbol{R}_j)^{-1} \tag{29}$$

is a known matrix. As assumed, for every $j \in [N]$, $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ and $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ are Gaussian. Hence, the distance between $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ and $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ is sufficient to be defined by the first two moments, i.e., their means and covariances. Recall that $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ is parameterized by $\boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}$, and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$, while $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ is parameterized by $\bar{\boldsymbol{x}}_j, \boldsymbol{M}_j$, and $\boldsymbol{R}_j$. Therefore, it is sufficient to only define uncertainty sets for matrices $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$ because $\boldsymbol{c}_{j,\boldsymbol{x}}$ and $\boldsymbol{c}_{j,\boldsymbol{v}}$ have been set to their nominal values. To be specific, the problem (28) can be rewritten as

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Theta}''} & \operatorname{Tr}\sum_{j=1}^{N}\mu_j\cdot\left\{\boldsymbol{P}_j+(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})(\hat{\boldsymbol{x}}_j-\hat{\boldsymbol{x}})^\top\right\} \\ \textit{s.t.} & \left\{\begin{array}{ll}\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top=\boldsymbol{K}_j\cdot(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top+\boldsymbol{\Sigma}_{j,\boldsymbol{v}}), & \forall j\in[N],\\ \Delta_{j,\boldsymbol{x}}(\boldsymbol{\Sigma}_{j,\boldsymbol{x}},\boldsymbol{M}_j)\leq\theta_{j,\boldsymbol{x}}, & \forall j\in[N],\\ \Delta_{j,\boldsymbol{v}}(\boldsymbol{\Sigma}_{j,\boldsymbol{v}},\boldsymbol{R}_j)\leq\theta_{j,\boldsymbol{v}}, & \forall j\in[N],\\ \Delta_0(\boldsymbol{\mu},\bar{\boldsymbol{\mu}})\leq\theta_0,\quad \mathbf{1}^\top\boldsymbol{\mu}=1,\quad \boldsymbol{\mu}\geq\mathbf{0},\\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\succeq\mathbf{0},\boldsymbol{\Sigma}_{j,\boldsymbol{v}}\succeq\mathbf{0}, & \forall j\in[N],\end{array}\right. \end{array} \tag{30}$$

where $\boldsymbol{\Theta}'' := \{\mu_j, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, $\Delta_{j,\boldsymbol{x}}(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{M}_j)$ means a similarity measure between the two matrices $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$ and $\boldsymbol{M}_j$, and $\Delta_{j,\boldsymbol{v}}(\boldsymbol{\Sigma}_{j,\boldsymbol{v}}, \boldsymbol{R}_j)$ denotes a similarity measure between the two matrices $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$ and $\boldsymbol{R}_j$.

The problem (30) can be solved using the coordinate descent method: we can first fix $\boldsymbol{\mu}$ and optimize over $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, and then fix $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$ and optimize over $\boldsymbol{\mu}$. The iterative process proceeds until it converges. Since $\mu_j > 0, \forall j \in [N]$ and the constraints of $\boldsymbol{\mu}$ are not coupled with the constraints of $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, this iterative process can converge only in one round. To be specific, it is sufficient to first fix $\boldsymbol{\mu}$ and optimize over $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, and then fix $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$ and optimize over $\boldsymbol{\mu}$: no more iterations are required.[4]

The sub-problem where $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\}_{j\in[N]}$ and $\{\boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$ are fixed and the optimization is conducted over $\boldsymbol{\mu}$ has been solved in Subsection IV-A. In the following, we focus on solving the sub-problem when $\boldsymbol{\mu}$ is fixed, which is

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Upsilon}} & \sum_{j=1}^{N}\mu_j\cdot\operatorname{Tr}\boldsymbol{P}_j \\ \textit{s.t.} & \left\{\begin{array}{ll}\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top=\boldsymbol{K}_j\cdot(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top+\boldsymbol{\Sigma}_{j,\boldsymbol{v}}), & \forall j\in[N],\\ \Delta_{j,\boldsymbol{x}}(\boldsymbol{\Sigma}_{j,\boldsymbol{x}},\boldsymbol{M}_j)\leq\theta_{j,\boldsymbol{x}},\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\succeq\mathbf{0}, & \forall j\in[N],\\ \Delta_{j,\boldsymbol{v}}(\boldsymbol{\Sigma}_{j,\boldsymbol{v}},\boldsymbol{R}_j)\leq\theta_{j,\boldsymbol{v}},\ \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\succeq\mathbf{0}, & \forall j\in[N],\end{array}\right. \end{array} \tag{31}$$

where $\boldsymbol{\Upsilon} := \{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$. Since $\boldsymbol{\mu} > \mathbf{0}$, for each $j \in [N]$, the problem (31) can be separately solved because both the objective function and the feasible region are separable. Specifically, for all $i, j \in [N]$ and $i \neq j$, $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ are not coupled, so are their respective constraints. As a result, the solution $\boldsymbol{\Upsilon}^* = \{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}^*, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}^*\}_{j\in[N]}$ of (31) also solves

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Sigma}_{j,\boldsymbol{x}},\boldsymbol{\Sigma}_{j,\boldsymbol{v}}} & \operatorname{Tr}\boldsymbol{P}_j \\ \textit{s.t.} & \left\{\begin{array}{l}\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top=\boldsymbol{K}_j\cdot(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top+\boldsymbol{\Sigma}_{j,\boldsymbol{v}}),\\ \Delta_{j,\boldsymbol{x}}(\boldsymbol{\Sigma}_{j,\boldsymbol{x}},\boldsymbol{M}_j)\leq\theta_{j,\boldsymbol{x}},\quad \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\succeq\mathbf{0}\\ \Delta_{j,\boldsymbol{v}}(\boldsymbol{\Sigma}_{j,\boldsymbol{v}},\boldsymbol{R}_j)\leq\theta_{j,\boldsymbol{v}},\quad \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\succeq\mathbf{0}\end{array}\right. \end{array} \tag{32}$$

[4]This can be intuitively seen from the equivalence between (31) and (32): the specific value of $\boldsymbol{\mu}$ does not influence the optimal values of $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$. Hence, the optimization (30) over $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$ can be completely solved in the first-round iteration. Afterward, the optimization over $\boldsymbol{\mu}$ can be readily and completely solved as well.

for every $j \in [N]$, and vice versa. Therefore, it necessitates and suffices to separately solve (32) for every $j \in [N]$.

Recall from Appendix A for the definition of the matrix function:

$$\begin{aligned}&\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}})\\ &\quad := \boldsymbol{\Sigma}_{j,\boldsymbol{x}} - \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1}\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}.\end{aligned}$$

The theorem below plays an important role to further simplify the problem (32).

Theorem 1: The matrix function $\mathrm{Tr}\,\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}})$ is monotonically increasing in both $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$. Namely, for every given $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$, if $\boldsymbol{\Sigma}_1 \succeq \boldsymbol{\Sigma}_2$, we have $\mathrm{Tr}\,\boldsymbol{P}_j(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}) \geq \mathrm{Tr}\,\boldsymbol{P}_j(\boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_{j,\boldsymbol{v}})$. Likewise, for every given $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$, if $\boldsymbol{\Sigma}_1 \succeq \boldsymbol{\Sigma}_2$, we have $\mathrm{Tr}\,\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_1) \geq \mathrm{Tr}\,\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_2)$.

Proof: See Appendix E. □

Theorem 1 equivalently transforms the problem (32) to

$$\begin{aligned}\max_{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}} \quad & \mathrm{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}]\\ \textit{s.t.} \quad & \begin{cases} \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}),\\ \Delta_{j,\boldsymbol{x}}(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{M}_j) \leq \theta_{j,\boldsymbol{x}},\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \succeq \boldsymbol{0},\\ \Delta_{j,\boldsymbol{v}}(\boldsymbol{\Sigma}_{j,\boldsymbol{v}}, \boldsymbol{R}_j) \leq \theta_{j,\boldsymbol{v}},\ \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}. \end{cases}\end{aligned} \tag{33}$$

The proposition below summarizes the solution of the reformulated distributionally robust Bayesian estimation problem (27).

Proposition 4: Suppose both the model set and the prior model probability $\boldsymbol{\omega}$ are uncertain. If we conduct the robustification over the posterior model probability $\boldsymbol{\mu}$ instead of the prior model probability $\boldsymbol{\omega}$ and Assumption 1 is adopted, then the reformulated distributionally robust Bayesian estimation problem (27) can be solved by

1) **Step 1.** Solving (33) for every $j \in [N]$;
2) **Step 2.** Solving (25) with updated $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{\forall j \in [N]}$ from Step 1.[5] □

In the following, as a demonstration, we discuss the particular cases where $\Delta_{j,\boldsymbol{x}}(\cdot,\cdot)$ and $\Delta_{j,\boldsymbol{v}}(\cdot,\cdot)$ are defined by the moments-based similarity measure. The case where $\Delta_{j,\boldsymbol{x}}(\cdot,\cdot)$ and $\Delta_{j,\boldsymbol{v}}(\cdot,\cdot)$ are defined by the Wasserstein distance is discussed in Appendix F.

If we use the moment-based ambiguity sets for $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ as in [13], the problem (33) is particularized into

$$\begin{aligned}\max_{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}} \quad & \mathrm{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}]\\ \textit{s.t.} \quad & \begin{cases} \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}),\\ (1-\theta_{j,\boldsymbol{x}})\boldsymbol{M}_j \preceq \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \preceq (1+\theta_{j,\boldsymbol{x}})\boldsymbol{M}_j,\\ (1-\theta_{j,\boldsymbol{v}})\boldsymbol{R}_j \preceq \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \preceq (1+\theta_{j,\boldsymbol{v}})\boldsymbol{R}_j,\\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \succeq \boldsymbol{0},\ \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}, \end{cases}\end{aligned} \tag{34}$$

where $0 \leq \theta_{j,\boldsymbol{x}} \leq 1$, $0 \leq \theta_{j,\boldsymbol{v}} \leq 1$. The problem (34) is a standard linear positive semi-definite program (SDP), which can be efficiently solved using mature SDP solvers such as MOSEK, GUROBI, YALMIP. However, we show that in a special case, (34) can be analytically solved.

Theorem 2: If $\theta_{j,\boldsymbol{x}} = \theta_{j,\boldsymbol{v}} = \theta_j$, the optimal solution to (34) is analytically given by

$$\begin{cases} \boldsymbol{\Sigma}^*_{j,\boldsymbol{x}} &= (1+\theta_j)\boldsymbol{M}_j,\\ \boldsymbol{\Sigma}^*_{j,\boldsymbol{v}} &= (1+\theta_j)\boldsymbol{R}_j. \end{cases} \tag{35}$$

[5]When $\{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{\forall j \in [N]}$ are updated, $\boldsymbol{A}$ and $\boldsymbol{b}$ in (25) would be updated as well because $\hat{\boldsymbol{x}}_j, \boldsymbol{P}_j, \forall j \in [N]$ therein would be updated.

Proof: See Appendix G. □

V. Distributionally Robust Interactive Multiple Model State Estimator

The overall distributionally robust interactive multiple model (DRIMM) state estimator is summarized in Algorithm 1, which is a robustified, modified version of the standard IMM filter in Algorithm 3 in the online supplementary materials. In Algorithm 1, for clarity, we only highlight the modifications and differences between the proposed distributionally robust IMM filter and the standard IMM filter in Algorithm 3. For other algorithmic statements, see Algorithm 3. Recall from Appendix A that $\boldsymbol{P}_j$ is defined by $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$. In Step 2.1, another modification method for $\boldsymbol{P}_{j,k|k}$ based on Wasserstein distance is available in Appendix F; in Step 2.2, another modification method for $\boldsymbol{\mu}_{k|k}$ based on Wasserstein distance is available in Appendix C. Note that Step 2.1 in Algorithm 1 solves (34) using (35) to modify $\boldsymbol{P}_{j,k|k}$, $\forall j \in [N]$; Line 6 is due to Theorem 2 and the definition of $\boldsymbol{P}_{j,k|k}$ is in Appendix A. Note also that the solution of (26) is given in Algorithm 2 in the online supplementary materials.

Algorithm 1 Distributionally Robust IMM State Estimator

Definition: θ_0, θ_j, for every $j \in [N]$ are the size parameters to define the scale of the ambiguity set (12). N is the number of the nominal models. k denotes the discrete time index. $\boldsymbol{P}_{j,k|k}$ is the posterior state estimation error covariance of the j^{th} nominal model at the time k. $\boldsymbol{\mu}_{k|k}$ is the posterior model probability at the time k.

External: Algorithm 3 (i.e., the standard IMM filter [3], [5]) in the online supplementary materials.

Input: θ_0, θ_j, for every $j \in [N]$.

1: **while** true **do**
2: *// (Step 1) At Time k*
3: Execute Step 1 (i.e., Lines 2 $\sim$ 24) of Algorithm 3
4: *// (Step 2) Robustification; See Proposition 4*
5: *// (Step 2.1) Modify $\boldsymbol{P}_{j,k|k}, \forall j \in [N]$ via (34)*
6: $\boldsymbol{P}_{j,k|k} \leftarrow (1+\theta_j) \cdot \boldsymbol{P}_{j,k|k},\ \forall j \in [N]$
7: *// (Step 2.2) Modify $\boldsymbol{\mu}_{k|k}$ via (25)*
8: Solve (26) to modify $\boldsymbol{\mu}_{k|k}$
9: *// (Step 3) Combined Posterior State Estimate*
10: Execute Step 2 (i.e., Lines 25 $\sim$ 27) of Algorithm 3 to calculate robustified $\hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{P}_{k|k}$, using modified $\boldsymbol{P}_{j,k|k}, \forall j \in [N]$ and $\boldsymbol{\mu}_{k|k}$
11: *// Next Time Step*
12: $k \leftarrow k+1$
13: **end while**

Output: $\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \boldsymbol{\mu}_{k|k}, \forall k$

Remark 2 (Only TPMs Are Uncertain): Uncertain TPMs lead to uncertain model weights. If there do not exist uncertainties in candidate models and only model weights are uncertain, we can robustify the IMM filter through only modifying posterior model weights $\boldsymbol{\mu}_{k|k}$. In this case, Step 2.1 in Algorithm 1 should be ignored and only Step 2.2 should be conducted. □

Remark 3 (Candidate Models Are Uncertain): As long as there exist uncertainties in the designed nominal model set, the DRIMM filter in Algorithm 1 cannot be simplified: Neither Step 2.1 nor Step 2.2 can be ignored. This is because uncertain candidate models introduce uncertainties in evaluating posterior model weights. Specifically, from (3), we can see that when there exist uncertainties in candidate models, the evaluation of model likelihoods cannot be exact, and therefore, the posterior model weights cannot be exact either. However, in practice, when candidate models are uncertain, one may ignore Step 2.2 and just conduct Step 2.1. This leads to a simplification (or approximation) of Algorithm 1, which is an IMM filter using the robustified Kalman filter (rather than the conventional non-robust one) for each candidate model. Recall that the distributionally robust state estimation method for each candidate model has been discussed in [13], [14]. □

Remark 4 (Computational Burdens): Compared with the standard IMM filter in Algorithm 3 in the online supplementary materials, the DRIMM filter in Algorithm 1 has an additional robustification procedure in Step 2. In Step 2.1, there is only a value assignment procedure, and hence, no extra computational burdens are introduced. However, in Step 2.2, an iterative process is involved to numerically solve (26); see Algorithm 2 in the online supplementary materials. Therefore, extra computational resources are required to conduct robustification over $\boldsymbol{\mu}_{k|k}$. □

VI. Experiments

We conduct multi-model target tracking experiments to show the power of the proposed method. All the source data and codes are available online at GitHub: https://github.com/Spratm-Asleaf/DRSE-Jump. The results are obtained by a Lenovo laptop with 16G RAM and 11th Gen Intel(R) Core(TM) i5-11300H CPU @ 3.10GHz. The programming environment is MATLAB 2019B. We implement the following methods for performance comparison.

1) **KF**: the standard Kalman filter with the exactly known model transition trajectory. In practice, the standard Kalman filter is *not applicable* for jump linear systems because we do not know the true model transition trajectory. However, in a simulation experiment, we know the underlying true model transition trajectory so that the standard Kalman filter can be used to provide the theoretically optimal filtering performance.
2) **IMM-T**: the IMM filter using the exactly *true* model transition probability matrix. Note that in a simulation experiment, we know the underlying true model transition probability matrix. But in practice, this method is *not applicable* as the standard Kalman filter is.
3) **IMM-N**: the IMM filter using a user-specified *nominal* model transition probability matrix.
4) **IMM-R**: the IMM filter using the distributionally *robust* Kalman filter in [13] for every operating mode $j \in [N]$. To clarify further, in Algorithm 3 in supplementary materials, Step 1.4 is modified according to the robustified Kalman filter in [13] and all other steps in Algorithm 3 remain unchanged. See also Remark 3.
5) **IMM-B**: the IMM filter using the *Bayesian* method to estimate the true TPM [7]. This method is applicable only when TPM is uncertain.
6) **IMM-M**: the IMM filter using the *maximum* likelihood method to estimate the true TPM [12]. This method is applicable only when TPM is uncertain.
7) **DRIMM**: the distributionally robust IMM filter in Algorithm 1, which shares the same nominal model transition probability matrix with the IMM-N filter. This method is applicable for any types of uncertainty. Particularly, it *can* flexibly respond to specified types of uncertainty.
8) **IMM-RS**: the *risk-sensitive* IMM filter [21]. This method is applicable for any types of uncertainty. However, it *cannot* flexibly respond to specified types of uncertainty.
9) **IMM-C**: the *compensation*-based IMM filter [9]. This method is applicable for any types of uncertainty. However, it *cannot* flexibly respond to specified types of uncertainty.

Since the KF and the IMM-T filters are *not applicable* in practice, given a nominal TPM, a method is promising if it can outperform the IMM-N filter. Note that the IMM-T filter is *not optimal* for jump linear systems due to its approximation nature as elucidated in the introduction.

A. Simulated Experiments

We continue studying the classic one-dimensional target tracking problem in [7], [9], [11], [17], [38] where the target maneuvers with Markov switching accelerations. According to [2], the motion of a target can be independently tracked in each axis; therefore, focusing only on one axis does not lose the generality. The jump Markov linear system is defined by

$$\begin{cases} \begin{bmatrix} \mathrm{p}_{k+1} \\ \mathrm{s}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{p}_k \\ \mathrm{s}_k \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} [\mathrm{a}_{j,k} + \mathrm{w}_k] \\ \mathrm{y}_k = \mathrm{p}_k + \mathrm{v}_k \end{cases} \tag{36}$$

where $\mathrm{p}_k \in \mathbb{R}$ and $\mathrm{s}_k \in \mathbb{R}$ denote the position and velocity of a moving target at time k, respectively; T denotes the sampling time; $\mathrm{a}_{j,k} \in \mathbb{R}$ denotes the possible target maneuvering acceleration at time k which takes the value of

$$\mathrm{a}_{j,k} = \begin{cases} 0, & j = 1, \\ 20, & j = 2, \\ -20, & j = 3; \end{cases} \tag{37}$$

$\mathrm{w}_k \in \mathbb{R}$ and $\mathrm{v}_k \in \mathbb{R}$ denote the acceleration noise and the measurement noise at the time k, respectively. The following settings, as conventionally made in the mentioned state-of-the-art, are taken in this section: $\mathrm{p}_0 \sim \mathcal{N}(80000, 100^2)$, $\mathrm{s}_0 \sim \mathcal{N}(400, 100^2)$, $\mathrm{w}_k \sim \mathcal{N}(0, 2^2)$, $\mathrm{v}_k \sim \mathcal{N}(0, 100^2)$, for every k, and the model's initial probability $\boldsymbol{\mu}_0 = [0.8, 0.1, 0.1]^\top$. In this paper, we set $T = 1$. Throughout the simulated experiments, we use $\boldsymbol{\Pi}_0$ to denote the true model transition probability matrix (TPM) and $\boldsymbol{\Pi}$ the nominal TPM. In addition, the nominal value of maneuvering acceleration is denoted as $\bar{\mathrm{a}}_{j,k}$, which might be different from the true $\mathrm{a}_{j,k}$ in (37).

For each simulation scenario, we conduct 100 independent Monte-Carlo episodes and each episode runs 1000 time steps.

The overall performance evaluation method for each filter is the averaged root-mean-square error (RMSE):

$$\frac{1}{100}\sum_{l=1}^{100}\left[\sqrt{\frac{1}{1000}\sum_{k=1}^{1000}(p_k^{(l)}-\hat{p}_k^{(l)})^2+(s_k^{(l)}-\hat{s}_k^{(l)})^2}\right],$$

where $p_k^{(l)}$ (resp. $s_k^{(l)}$) denotes the true value of position (resp. velocity) at the time k in the l^{th} Monte-Carlo simulation, and $\hat{p}_k^{(l)}$ (resp. $\hat{s}_k^{(l)}$) its estimate. Every method is well-tuned and performs at its best for the given simulation scenario; details can be accessed in the shared source codes and therefore omitted here. For example, in the proposed DRIMM filter (Algorithm 1), we have set $\theta_0 = 0.1$ and $\theta_j = 0.25$ if the j^{th} candidate model is uncertain. The influence of the values of θ's to the filtering performances will be discussed later in Subsection VI-A5.

1) When Only TPM Is Uncertain: First, we suppose that the nominal $\mathbf{\Pi}$ is different from the true $\mathbf{\Pi}_0$. However, the candidate models are exact (i.e., the nominal $\bar{\mathrm{a}}_{j,k}$ is the same as the true $\mathrm{a}_{j,k}$ in (37)). In this case, the IMM-R method is not applicable since it is not designed to handle the uncertainty in $\mathbf{\Pi}$. Note that Step 2.1 in Algorithm 1 should be ignored; see Remark 2.

We first set the true $\mathbf{\Pi}_0$ and the nominal $\mathbf{\Pi}$ as follows:

$$\mathbf{\Pi}_0 := \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \mathbf{\Pi} := \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}.$$

The results are shown in Table I.

TABLE I
WHEN ONLY TPM IS UNCERTAIN

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.77e-06	IMM-T*	81.60	1.99e-05
IMM-N	82.44	1.94e-05	IMM-B	82.36	2.54e-05
IMM-M	82.32	5.32e-05	IMM-C	166.01	2.06e-05
IMM-RS	82.37	8.05e-05	DRIMM	**82.19**	2.14e-04

Avg Time: Average Execution Time at each time step (unit: seconds);
1e-5: 1×10^{-5};
*: Not applicable methods in practice.

Next, we set $\mathbf{\Pi}_0$ and $\mathbf{\Pi}$ as follows:

$$\mathbf{\Pi}_0 := \begin{bmatrix} 0.1 & 0.1 & 0.8 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \mathbf{\Pi} := \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

That is, the first row and the third row of $\mathbf{\Pi}_0$ are the same. However, the nominal $\mathbf{\Pi}$ mistakenly assumes that they are different. The results are given in Table II.

TABLE II
WHEN ONLY TPM IS UNCERTAIN (ANOTHER EXAMPLE)

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.91e-06	IMM-T*	82.48	2.00e-05
IMM-N	84.56	1.98e-05	IMM-B	84.03	2.54e-05
IMM-M	**83.95**	5.41e-05	IMM-C	111.60	2.07e-05
IMM-RS	84.63	8.09e-05	DRIMM	84.47	2.91e-04

See Table I for table notes.

From Tables I and II (and also many other similar experiments), the following main points have to be highlighted.

1) The compensation-based method (i.e., IMM-C) is not satisfactory for the two cases. This is because there does not exist a dominating operating mode for both two cases: i.e., the systems' true operating modes are frequently switching. (However, when the true switching frequency is low, the IMM-C method would be useful; see the real-world target tracking experiments in Appendix VI-B.)
2) In the first case (Table I), the IMM-B method, the IMM-M method, the IMM-RS method, and the proposed DRIMM method are all better than the nominal IMM-N method. This suggests that the four methods are all able to combat uncertainties in the nominal TPM $\mathbf{\Pi}$. In the second case (Table II), the IMM-M and the IMM-B methods perform significantly better than the IMM-RS, the proposed DRIMM, and the nominal IMM-N methods. This is because the IMM-M and the IMM-B methods can adaptively estimate the unknown true TPM so that the uncertainties in the nominal TPM would be reduced. As a result, the filtering performances of the IMM-M and the IMM-B methods can be significantly improved. This result is consistent with the findings in [13], [30]: i.e., adaptive methods which aim to reduce the uncertainties would be potentially better than robust methods that aim to just hedge against uncertainties.
3) Among the robust methods, the proposed DRIMM method performs better than the generic IMM-RS method because the former is able to specifically respond to the uncertainties only in the nominal TPM. In contrast, the IMM-RS method only assumes that there exist model uncertainties; it does not take into account where and how the uncertainties exist. Therefore, the IMM-RS filter tends to be overly conservative.

In summary, when there exist uncertainties only in the nominal TPM, we would suggest that the practitioners should first consider adaptive methods such as the IMM-M and the IMM-B methods because the two methods tend to estimate the unknown true TPM and the uncertainties in the nominal TPM are to be reduced. In addition, the two adaptive methods are computationally more efficient compared to the proposed DRIMM method.

2) When The Model Set Is Incomplete: Second, we consider the case where the true system dynamics has five operating modes: $\mathrm{a}_{1,k} = 0$, $\mathrm{a}_{2,k} = 10$, $\mathrm{a}_{3,k} = -10$, $\mathrm{a}_{4,k} = 20$, and $\mathrm{a}_{5,k} = -20$. But the nominal model set still assumes that there exist three operating modes, as in (37). This is a common situation in target tracking. To be specific, when we do not know the true maneuvering acceleration a_k of the moving target, there exist infinitely many possible values for a_k. However, considering practical constraints such as computational complexity, we may only assume finitely many values for a_k. As a result, the designed nominal model set is potentially incomplete. The true and nominal TPMs are set, respectively, to

$$\mathbf{\Pi}_0 := \begin{bmatrix} 0.6 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.6 \end{bmatrix}, \ \mathbf{\Pi} := \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}.$$

The results are shown in Table III.

TABLE III
THE NOMINAL MODEL SET IS NOT COMPLETE

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.77e-06	IMM-T*	92.24	3.79e-05
IMM-N	101.04	1.94e-05	IMM-B	100.73	2.54e-05
IMM-M	100.00	5.49e-05	IMM-C	100.42	2.09e-05
IMM-RS	98.53	8.02e-05	DRIMM	**95.18**	2.98e-04

See Table I for table notes.

As we can see from Table III, in this case, the adaptive methods (i.e., the IMM-B and IMM-M methods) cannot work significantly better than the nominal IMM-N method because if the nominal model set is incomplete, nominal candidate models are inexact models when the true operating modes are not included in the nominal model set. Hence, there exist uncertainties not only in the TPM but also in nominal candidate models. As a result, robust methods (i.e., the IMM-RS and DRIMM methods) that are able to combat all kinds of uncertainties are promising. Among robust methods, the proposed DRIMM method outperforms the IMM-RS method.

3) When Candidate Models Are Uncertain: Third, we suppose that the candidate models are uncertain (i.e., the nominal $\bar{\mathrm{a}}_{j,k} = [0, 10, -10]$ is different from the true $\mathrm{a}_{j,k} = [0, 20, -20]$ in (37)).

When $\mathbf{\Pi} = \mathbf{\Pi}_0$. Let the diagonal elements of the TPM be 0.8s, and all other non-diagonal elements be 0.1s. In this case, the IMM-B method and the IMM-M are not applicable since they are not designed to handle the uncertainties in candidate models. The results are shown in Table IV.

TABLE IV
CANDIDATE MODELS ARE UNCERTAIN BUT TPM IS EXACT

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.80e-06	IMM-T*	81.60	2.02e-05
IMM-N	95.83	2.08e-05	IMM-C	161.48	2.15e-05
IMM-RS	90.43	8.44e-05	DRIMM	**87.67**	2.85e-04
IMM-R	88.14	2.54e-05			

See Table I for table notes.

As we can see from Table IV, when there exist uncertainties only in the candidate models, the IMM-R filter, the IMM-RS filter, and the DRIMM filter can outperform the nominal IMM-N filter. In addition, the proposed DRIMM filter works better than the IMM-R filter, which validates the claims in Remark 3: That is, uncertain candidate models lead to uncertain model weights, and therefore, robustification should also be conducted over model weights.

When $\mathbf{\Pi} \neq \mathbf{\Pi}_0$. Let the diagonal elements of the true TPM (i.e., $\mathbf{\Pi}_0$) be 0.8s, and all other non-diagonal elements be 0.1s. Let the diagonal elements of the nominal TPM (i.e., $\mathbf{\Pi}$) be 0.6s, and all other non-diagonal elements be 0.2s. The results are shown in Table V.

TABLE V
BOTH CANDIDATE MODELS AND TPM ARE UNCERTAIN

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.84e-06	IMM-T*	81.60	1.98e-05
IMM-N	104.48	1.92e-05	IMM-B	103.99	2.61e-05
IMM-M	102.93	5.35e-05	IMM-C	103.15	2.07e-05
IMM-RS	98.44	7.98e-05	DRIMM	**90.23**	2.99e-04
IMM-R	93.25	2.50e-05			

See Table I for table notes.

As we can see from Table V, when there exist uncertainties in both the candidate models and the nominal TPM, all methods (except the KF and IMM-T methods) can outperform the nominal IMM-N filter because the uncertainties can be adaptively reduced or robustly withstand to some degree. Also, the proposed DRIMM method works better than the IMM-RS method as the latter tends to be overly conservative. In addition, the DRIMM method is better than the IMM-R method because the latter can only partially respond to the uncertainties in the candidate models. In contrast, the DRIMM method is able to work for uncertainties in both the candidate models and the nominal TPM.

4) When There Are No Model Uncertainties: Fourth, we suppose that there are no any model uncertainties (i.e., both the TPM and all candidate models are exact). We let the diagonal elements of the TPM (N.B.: $\mathbf{\Pi}_0 = \mathbf{\Pi}$) be 0.8s, and all other non-diagonal elements be 0.1s. In addition, we let the nominal $\bar{\mathrm{a}}_k$ be equal to the true a_k in (37). The filtering results are shown in Table VI.

TABLE VI
THERE EXIST NO ANY MODEL UNCERTAINTIES

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
KF*	43.62	1.78e-06	IMM-T*	81.60	2.01e-05
IMM-N	**81.60**	1.98e-05	IMM-B	81.61	2.59e-05
IMM-M	81.61	5.37e-05	IMM-C	110.69	2.18e-05
IMM-RS	82.39	8.05e-05	DRIMM	82.45	1.67e-04
IMM-R	82.44	2.51e-05			

See Table I for table notes.

As we can see from Table VI, when there exist no model uncertainties, all modified IMM methods would perform worse than the nominal IMM-N method. This result is consistent with the findings in [13], [30], [39]: There exists a trade-off between the robustness in uncertain conditions and the optimality in perfect conditions. To be specific, robust methods are not optimal in exact conditions but they can withstand uncertainties in inexact conditions; in contrast, the nominal IMM-N method is the best in exact conditions but it has no ability to combat uncertainties in inexact conditions.

5) On The Size of Uncertainty Set (12)*:* In this subsection, we investigate the influence of the values of the size parameters θ's in the uncertainty set (12). As an example and without loss of generality, we suppose that the TPM is exact and only the candidate models are uncertain. To be specific, the nominal $\bar{\mathrm{a}}_{j,k} = [0, 10, -10]$ is different from the true $\mathrm{a}_{j,k} = [0, 20, -20]$. The diagonal elements of the TPM (N.B.: $\mathbf{\Pi}_0 = \mathbf{\Pi}$) are set to 0.8s and the non-diagonal ones

are set to 0.1s. The performances of the DRIMM filter are given in Table VII, against the value of $\theta := \theta_j$ for every j such that the jth model is uncertain (i.e., $j = 2, 3$ in this case). As we can see from Table VII, the size of the ambiguity set (12) can be neither too large nor too small. If the size is too large, the DRIMM filter would be extremely conservative, while if the size is too small, the DRIMM filter cannot offer sufficient robustness. This result is consistent with those in [13], [30], [39]. However, the convincing tuning method for θ's is lacking because the true system state (i.e., training data set) is unknown for a real state estimation problem, and therefore, θ's cannot be rigorously tuned to be optimal. As such, we leave this as an open problem to be solved by the future research. At present, we can only suggest that practitioners can try appropriate values of θ's for their specific problems.

TABLE VII
PERFORMANCES (I.E., RMSE) OF DRIMM AGAINST VALUES OF θ

θ	0.0	0.1	0.2	0.3	0.4
RMSE	95.44	90.49	87.67	86.10	**85.30**
θ	0.5	0.6	0.7	0.8	0.9
RMSE	85.41	85.36	85.51	85.82	86.24

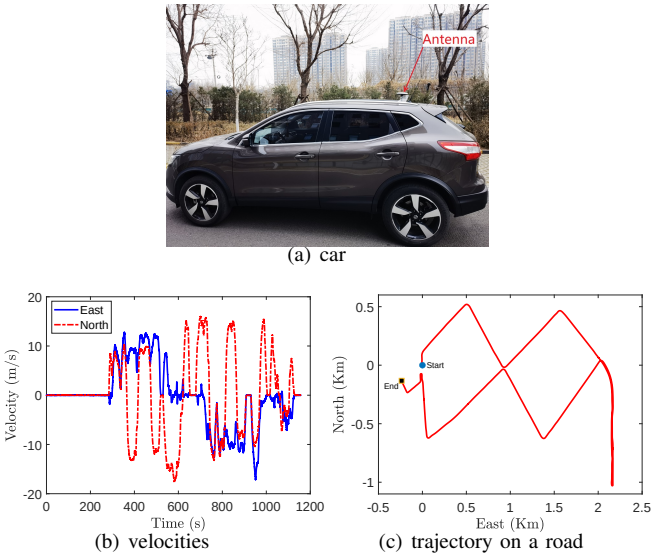
B. Real-World Target Tracking Examples

We track the real-time positions and velocities of a slowly-maneuvering car and a highly-maneuvering quadrotor drone, respectively. The raw position measurements are received by traditional GPS (global positioning system) solutions, while the high-accuracy real-time position and velocity measurements (treated as ground truths) are obtained by RTK (real-time kinematic) solutions [40]. We continue adopting the target tracking framework in (36) where the target's true acceleration at every time k is unknown, and hence, some nominal values are used. The nominal model set is therefore incomplete and uncertain.

1) Track A Slowly-Maneuvering Car: In this experiment, the data are collected by a slowly-maneuvering car that carries a GPS solution and an RTK solution; the car travels on a road in Beijing, China. The commercial model of the used RTK chip is P327 and of the antenna is UA35, both produced by UniStrong Co., Ltd., Beijing, China; see http://en.unistrong.com/. The car, its real-time velocities, and its trajectory are shown in Fig. 1.

Without loss of generality, we study the tracking problem in the east axis (in the east-north-up coordinate). We suppose that there are three nominal values for acceleration in the east axis: $\bar{\mathrm{a}}_{j,k} = [0, 2.5, -2.5]$. The diagonal elements of the nominal TPM are set to 0.8s and the non-diagonal ones are set to 0.1s. The tracking results are shown in Table VIII.

As we can see, in this case, the robust filters (i.e., the IMM-RS, IMM-R, and DRIMM) are not preferable. In contrast, the adaptive filters (i.e., the IMM-B and IMM-M) outperform the nominal IMM-N filter because the uncertainty in the TPM is reduced. In particular, the IMM-C filter significantly works best. This is because the car is a slowly-maneuvering object and seldom maneuvers: the nearly-constant velocity (CV) model where $\bar{\mathrm{a}}_k = 0$ is the dominating model (N.B.: the trajectory consists of piece-wise straight-line segments). To clarify further, small uncertainties exist in the nominal model set and the CV model matches well most of the time.



(a) car
(b) velocities
(c) trajectory on a road

Fig. 1. Car, its real-time velocities, and its trajectory (data credit: UniStrong).

TABLE VIII
TRACKING RESULTS OF THE CAR (SEE TABLE I FOR NOTES)

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
IMM-N	2.30	2.21e-05	IMM-B	2.26	2.86e-05
IMM-M	2.02	5.75e-05	IMM-C	**1.72**	2.28e-05
IMM-RS	2.30	8.22e-05	DRIMM	2.30	5.07e-05
IMM-R	2.29	2.34e-05			

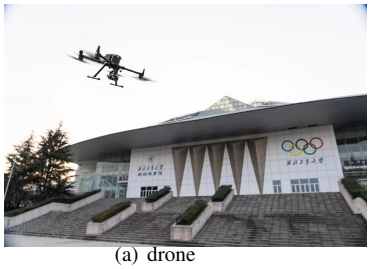
2) Track A Highly-Maneuvering Drone: In this experiment, the data are collected by a highly-maneuvering quadrotor drone that carries a GPS solution and an RTK solution. The commercial model of the drone is Matrice-300-RTK produced by DJI Co., Ltd., Shenzhen, China; see https://www.dji.com/. The drone flies following round trajectories in the air over an open playground, and the flying speed is about 6m/s when collecting data. Parts of the drone's real-time velocities and its trajectory are shown in Fig. 2.

Without loss of generality, we study the tracking problem in the east axis (in the east-north-up coordinate). We suppose that there are three nominal values for acceleration in the east axis: $\bar{\mathrm{a}}_{j,k} = [0, 2.5, -2.5]$. The diagonal elements of the nominal TPM are set to 0.8s and the non-diagonal ones are set to 0.1s. The tracking results are shown in Table IX.

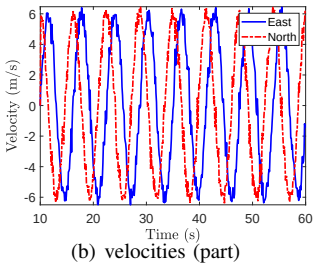
TABLE IX
TRACKING RESULTS OF THE DRONE (SEE TABLE I FOR NOTES)

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
IMM-N	0.99	4.85e-05	IMM-B	1.06	6.25e-05
IMM-M	1.00	1.00e-04	IMM-C	1.90	5.02e-05
IMM-RS	0.99	1.65e-04	DRIMM	**0.86**	1.20e-03
IMM-R	0.89	8.84e-05			

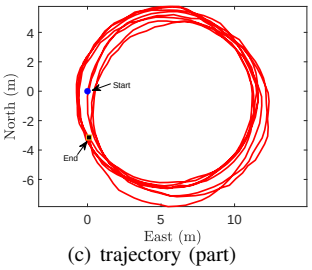
As we can see, in this case, the proposed DRIMM filter outperforms other filters. This is because the drone continu-



(a) drone



(b) velocities (part)



(c) trajectory (part)

Fig. 2. Drone, its real-time velocities, and traveling trajectory, starting from $10s$ and ending at $60s$ (data credit: Competitive Robot Base, Northwestern Polytechnical University, Xi'An, China).

ously maneuvers all the time (i.e., the operating models are frequently switching), and therefore, large uncertainties exist in both the nominal model set and the TPM. Further, we improve the flying speed of the drone from 6m/s to 12m/s. (The maximum flying speed of the used drone is 23m/s; see https://www.dji.com/.) The corresponding tracking results are shown in Table X. The results suggest that, since the modeling uncertainties in both the nominal model set and the TPM are larger, the advantage of the proposed DRIMM filter becomes much more significant. In this case, the IMM-C filter (which performs best in the car-tracking case) even diverges because it is overly confident about the model that has the highest estimated model probability.

TABLE X
TRACKING RESULTS OF THE DRONE (12M/S; SEE TABLE I FOR NOTES)

Filter	RMSE	Avg Time	Filter	RMSE	Avg Time
IMM-N	2.78	4.81e-05	IMM-B	2.68	6.12e-05
IMM-M	2.80	1.26e-04	IMM-C	Diverged	
IMM-RS	2.78	1.72e-04	DRIMM	**2.25**	9.22e-04
IMM-R	2.72	7.30e-05			

3) Closing Remarks: For detailed parameter settings, see shared source codes. The influences of the parameters used in filters are consistent with conclusions in simulated experiments in Section VI (e.g., sizes of ambiguity sets in Subsection VI-A5), and therefore, related discussions are omitted. One may use shared source codes to verify this claim.

VII. CONCLUSIONS

This paper studies the robust state estimation problem for jump Markov linear systems subject to several modeling uncertainties such as 1) the uncertainty of the model transition probability matrix, 2) the uncertainties of nominal candidate models in the nominal model set, and/or 3) the incompleteness of the nominal model set. A unified distributionally robust interactive multiple model (DRIMM) filter is proposed to withstand these model uncertainties. The following main points have to be highlighted.

1) When there exist uncertainties only in TPMs, the IMM filter with Bayesian estimation (i.e., the IMM-B method) and the IMM filter with maximum likelihood estimation (i.e., the IMM-M method) that aim to estimate the unknown true TPMs are recommended. All robust methods should be the last choices. However, the recommendation is conditioned on a sufficiently long time horizon (i.e., sufficient measurements are available so that the TPM can be estimated to a satisfactory accuracy level).[6]
2) When there exists a dominating operating model over time (i.e., the systems' operating modes do not frequently switch), the IMM filter with compensation (i.e., the IMM-C method) is recommended, for example, in slowly-maneuvering target tracking.
3) When there exist uncertainties in nominal candidate models or when the nominal model set is not complete, the proposed DRIMM method is recommended. This is because the DRIMM method is able to flexibly respond to all specified types of uncertainty. Also, it tends to be less conservative than the risk-sensitive IMM filter (i.e., the IMM-RS method).

Unfortunately, the performance of the DRIMM filter depends heavily on the size parameter(s) of the associated ambiguity set, and the convincing tuning method for the size parameter(s) is yet to be found. Therefore, future research following this work is expected to address this open issue.

APPENDIX A
OPTIMAL ESTIMATES AND ERROR COVARIANCES

We have

$$\overline{\hat{\boldsymbol{x}}_j} = \bar{\boldsymbol{x}}_j + \boldsymbol{M}_j\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{M}_j\boldsymbol{H}_j^\top + \boldsymbol{R}_j)^{-1}(\boldsymbol{y} - \boldsymbol{H}_j\bar{\boldsymbol{x}}_j), \quad (38)$$

$$\overline{\boldsymbol{P}_j} = \boldsymbol{M}_j - \boldsymbol{M}_j\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{M}_j\boldsymbol{H}_j^\top + \boldsymbol{R}_j)^{-1}\boldsymbol{H}_j\boldsymbol{M}_j, \quad (39)$$

$$\hat{\boldsymbol{x}}_j = \boldsymbol{c}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1}(\boldsymbol{y} - \boldsymbol{H}_j\boldsymbol{c}_{j,\boldsymbol{x}} - \boldsymbol{c}_{j,\boldsymbol{v}}), \quad (40)$$

and

$$\boldsymbol{P}_j = \boldsymbol{\Sigma}_{j,\boldsymbol{x}} - \boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top(\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}\boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1}\boldsymbol{H}_j\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \quad (41)$$

respectively. For detailed derivations of (38), (39), (40), and (41), refer to the standard Kalman filtering theory.

APPENDIX B
PROOF OF PROPOSITION 1

According to the strong min-max property in (14), (11) is equivalent to (13). Since for every $\mathbb{P} \in \mathcal{F}_{\mathbf{x},\mathbf{y}}(\boldsymbol{\theta})$, the associated optimal estimator $\hat{\mathbf{x}} \in \mathcal{H}_{\mathbf{y}}$ is uniquely determined by $\hat{\mathbf{x}} = \mathbb{E}(\mathbf{x}|\mathbf{y})$, (13) reduces to finding the worst-case conditional distribution $\mathbb{P}^*_{\mathbf{x}|\mathbf{y}=\boldsymbol{y}} \in \mathcal{F}_{\mathbf{x},\mathbf{y}=\boldsymbol{y}}(\boldsymbol{\theta})$ that solves

$$\max_{\mathbb{P}\in\mathcal{F}_{\mathbf{x},\mathbf{y}=\boldsymbol{y}}(\theta)} \operatorname{Tr}\mathbb{E}\Big\{[\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})][\mathbf{x} - \boldsymbol{\phi}(\mathbf{y})]^\top \Big| \mathbf{y} = \boldsymbol{y}\Big\}, \quad (42)$$

[6]One may use shared source codes to verify this claim.

where the ambiguity set $\mathcal{F}_{\mathbf{x},\mathbf{y}=\boldsymbol{y}}(\boldsymbol{\theta})$ contains all conditional distributions of $\mathbf{x}$ given $\mathbf{y}=\boldsymbol{y}$. Since every candidate distribution $\mathbb{P}_{\mathbf{x}|\mathbf{y}=\boldsymbol{y}}$ in $\mathcal{F}_{\mathbf{x},\mathbf{y}=\boldsymbol{y}}(\boldsymbol{\theta})$ is parameterized by $\{\omega_j, \boldsymbol{c}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{c}_{j,\boldsymbol{v}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}\}_{j\in[N]}$, by recalling (3), (40), (41), (17), and (18), problem (42) can be transformed into (19). □

Appendix C
Use Wasserstein Distance Between $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$

Below we discuss the case where the statistical similarity measure between $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$ is defined by the Wasserstein metric. The distributionally robust Bayesian problem (25) is particularized to

$$\max_{\boldsymbol{\mu}} \quad -\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu}$$
$$\text{s.t.} \quad \begin{cases} \min_{\boldsymbol{Q}} \sum_{i=1}^N \sum_{j=1}^N C_{ij} Q_{ij} \le \theta_0, \\ \sum_{j=1}^N Q_{ij} = \bar{\mu}_i, \qquad \forall i \in [N], \\ \sum_{i=1}^N Q_{ij} = \mu_j, \qquad \forall j \in [N], \\ Q_{ij} \ge 0, \qquad \forall i,j \in [N], \end{cases} \tag{43}$$

where C_{ij} measures the difference between the two distributions $\bar{\mathbb{P}}_{i,\mathbf{x},\mathbf{y}}$ and $\bar{\mathbb{P}}_{j,\mathbf{x},\mathbf{y}}$ (which can be defined by the Wasserstein distance, the KL divergence, or whatever suitable), and the matrix $\boldsymbol{Q} := \{Q_{ij}\}_{\forall i,j\in[N]}$ can be regarded as a discrete joint distribution whose marginals are $\boldsymbol{\mu}$ and $\bar{\boldsymbol{\mu}}$. Note that $\sum_{j=1}^N \mu_j = 1 = \sum_{j=1}^N \sum_{i=1}^N Q_{ij}$ is implicitly admitted because $\sum_{i=1}^N \sum_{j=1}^N Q_{ij} = \sum_{i=1}^N \bar{\mu}_i = 1$. Note also that the minimization operator in the first constraint can be dropped because it is redundant for this problem. For two Gaussian distributions $\mathcal{N}_d(\boldsymbol{c}_i, \boldsymbol{\Sigma}_i)$ and $\mathcal{N}_d(\boldsymbol{c}_j, \boldsymbol{\Sigma}_j)$, the Wasserstein distance between them is $C_{ij}^{\mathrm{W}} = \sqrt{\|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2 + \mathrm{Tr}\left[\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j - 2\left(\boldsymbol{\Sigma}_i^{\frac{1}{2}} \boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_i^{\frac{1}{2}}\right)^{\frac{1}{2}}\right]}$ and the KL divergence between them is $C_{ij}^{\mathrm{KL}} = \frac{1}{2}\left[\|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2_{\boldsymbol{\Sigma}_j^{-1}} + \mathrm{Tr}\left[\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_i - \boldsymbol{I}\right] + \ln\det\left(\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_j\right)\right]$.

By plugging $\sum_{i=1}^N Q_{ij} = \mu_j, \forall j \in [N]$ into the objective function, the problem (43) can be explicitly written as

$$\max_{\boldsymbol{Q}} \quad -\sum_{i=1}^N \sum_{j=1}^N \sum_{m=1}^N \sum_{n=1}^N Q_{mi} A_{ij} Q_{nj} + \sum_{i=1}^N \sum_{j=1}^N b_j Q_{ij}$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^N \sum_{j=1}^N C_{ij} Q_{ij} \le \theta_0, \\ \sum_{j=1}^N Q_{ij} = \bar{\mu}_i, \qquad \forall i \in [N], \\ Q_{ij} \ge 0, \qquad \forall i,j \in [N]. \end{cases} \tag{44}$$

The problem (44) is quadratic in the matrix-valued variable $\boldsymbol{Q}$. Therefore, solving (44) is difficult because existing general-purpose commercial solvers cannot be used. As a result, a specifically efficient solution method for (44) or (43) is expected. We propose to use the Frank–Wolfe method [41]. The Frank–Wolfe method is an iterative method. At each iteration, it linearizes the nonlinear objective of an optimization problem and uses the objective-linearized sub-problem to find a feasible direction along which the current solution can be improved. Specifically, the Frank–Wolfe method can construct a sequence $\{\boldsymbol{\mu}^{(r)}\}_{r=0,1,2,\ldots}$ such that it converges to the optimal solution of (43) as $r \to \infty$ where r denotes the iteration count.

Proposition 5: Let $r = 1, 2, \ldots$ be the iteration count. Let $\boldsymbol{c}^{(r)} := -2\boldsymbol{A}\boldsymbol{\mu}^{(r)} + \boldsymbol{b}$ denote the gradient of the objective of (43) at $\boldsymbol{\mu}^{(r)}$. Construct the following iteration process

$$\boldsymbol{\mu}^{(r+1)} = \boldsymbol{\mu}^{(r)} + \beta_r \cdot (\boldsymbol{s}^{(r)} - \boldsymbol{\mu}^{(r)}) \tag{45}$$

where $\boldsymbol{s}^{(r)} := [s_1^{(r)}, s_2^{(r)}, \ldots, s_N^{(r)}]^\top$, $s_j^{(r)} := \sum_{i=1}^N Q_{ij}^{*(r)}$, $\forall j \in [N]$,

$$\boldsymbol{Q}^{*(r)} := \mathrm{argmax}_{\boldsymbol{Q}} \quad \sum_{i=1}^N \sum_{j=1}^N c_j^{(r)} \cdot Q_{ij}$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^N \sum_{j=1}^N C_{ij} Q_{ij} \le \theta_0, \\ \sum_{j=1}^N Q_{ij} = \bar{\mu}_i, \quad \forall i \in [N], \\ Q_{ij} \ge 0, \qquad \forall i,j \in [N], \end{cases} \tag{46}$$

and the step size $\beta_r := \frac{2}{r+1}, \forall r$. Then $\boldsymbol{\mu}^{(r)}$ converges to an optimal solution of (43), as $r \to \infty$. [Note that the solution of (43) might not be unique.]

Proof: See Appendix D. □

Note that the sub-problem in (46) is a N^2-variate linear program that can be solved by the simplex method. However, the sub-problem (46) still introduces higher computational burdens to the solution method under the Wasserstein distance compared to the solution method under the KL divergence. To be specific, the computational complexity of the simplex method for (46) in the worst case is exponential in N^2; no better theoretical results are reported for the generic simplex method although it performs well empirically. However, the root-finding sub-problem in Algorithm 2 in the online supplementary materials is just polynomial in N if Newton's method is employed.

Remark 5: Note that in the Wasserstein case, (26) in Line 8 of Algorithm 1 should be replaced with (43). □

Remark 6: Different statistical similarity measures define different ambiguity sets (e.g., different geometric shapes), and different ambiguity sets further lead to different robust state estimation results [30]. Therefore, in practice, if computational powers allow, one can try all possible ambiguity sets to obtain better performance on specific problems. □

Appendix D
Proof of Proposition 5

This proposition specifies the Frank-Wolfe method [41] for the problem (43). Let $\boldsymbol{s}^{(r)}$ solve the objective-linearized sub-problem of (43)

$$\boldsymbol{s}^{(r)} = \quad \mathrm{argmax}_{\boldsymbol{\mu}} \quad [\boldsymbol{c}^{(r)}]^\top \boldsymbol{\mu}$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^N \sum_{j=1}^N C_{ij} Q_{ij} \le \theta_0, \\ \sum_{j=1}^N Q_{ij} = \bar{\mu}_i, \qquad \forall i \in [N], \\ \sum_{i=1}^N Q_{ij} = \mu_j, \qquad \forall j \in [N], \\ Q_{ij} \ge 0, \qquad \forall i,j \in [N]. \end{cases} \tag{47}$$

By plugging in $\mu_j := \sum_{i=1}^N Q_{ij}, \forall j \in [N]$ into the objective, (47) can be written as (46), which is a linear program that can be efficiently solved by the simplex method. In (45), the vector $\boldsymbol{s}^{(r)} - \boldsymbol{\mu}^{(r)}$ gives a feasible direction along which the current solution $\boldsymbol{\mu}^{(r)}$ can be improved to $\boldsymbol{\mu}^{(r+1)}$. The step size β_r can be $\beta_r := \frac{2}{r+1}, \forall r$ [41]. Since the objective function

of (43) is concave and continuously differentiable, and the feasible region of (43) is a simplex (thus compact and convex) after dropping the minimization operator in the first constraint, according to [41, Theorem 1], this proposition holds. □

APPENDIX E
PROOF OF THEOREM 1

We first prove the first part: for every given $\boldsymbol{\Sigma}_{j,\boldsymbol{v}}$, if $\boldsymbol{\Sigma}_1 \succeq \boldsymbol{\Sigma}_2$, we have $\operatorname{Tr} \boldsymbol{P}_j(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}) \geq \operatorname{Tr} \boldsymbol{P}_j(\boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_{j,\boldsymbol{v}})$. We consider two optimization problems:

$$\begin{array}{ll} \boldsymbol{U}_1 = & \underset{\boldsymbol{U}}{\operatorname{argmax}} \operatorname{Tr} \boldsymbol{U} \\ & \textit{s.t.} \left\{ \begin{array}{l} \left[\begin{array}{cc} \boldsymbol{\Sigma}_1 - \boldsymbol{U} & \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{\Sigma}_1 & \boldsymbol{H}_j \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{array} \right] \succeq \boldsymbol{0}, \\ \boldsymbol{U} \succeq \boldsymbol{0}. \end{array} \right. \end{array} \tag{48}$$

$$\begin{array}{ll} \boldsymbol{U}_2 = & \underset{\boldsymbol{U}}{\operatorname{argmax}} \operatorname{Tr} \boldsymbol{U} \\ & \textit{s.t.} \left\{ \begin{array}{l} \left[\begin{array}{cc} \boldsymbol{\Sigma}_2 - \boldsymbol{U} & \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{\Sigma}_2 & \boldsymbol{H}_j \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{array} \right] \succeq \boldsymbol{0}, \\ \boldsymbol{U} \succeq \boldsymbol{0}. \end{array} \right. \end{array} \tag{49}$$

Since $\boldsymbol{H}_j \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succ \boldsymbol{0}$ and $\boldsymbol{H}_j \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succ \boldsymbol{0}$, by Schur complement, we have $\boldsymbol{U}_1 = \boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top (\boldsymbol{H}_j \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1} \boldsymbol{H}_j \boldsymbol{\Sigma}_1$ and $\boldsymbol{U}_2 = \boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top (\boldsymbol{H}_j \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})^{-1} \boldsymbol{H}_j \boldsymbol{\Sigma}_2$. On the other hand,

$$\begin{aligned} \left[\begin{array}{cc} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{array} \right] &\preceq \left[\begin{array}{cc} \boldsymbol{\Sigma}_2 & \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{\Sigma}_2 & \boldsymbol{H}_j \boldsymbol{\Sigma}_2 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{array} \right] \\ &\preceq \left[\begin{array}{cc} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top \\ \boldsymbol{H}_j \boldsymbol{\Sigma}_1 & \boldsymbol{H}_j \boldsymbol{\Sigma}_1 \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \end{array} \right], \end{aligned}$$

which implies that the feasible region of (48) is larger than that of (49). Therefore, $\boldsymbol{U}_1 \succeq \boldsymbol{U}_2$, and $\boldsymbol{P}_j(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}) - \boldsymbol{P}_j(\boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}) = \boldsymbol{U}_1 - \boldsymbol{U}_2 \succeq \boldsymbol{0}$.

Next, we prove the second part: for every given $\boldsymbol{\Sigma}_{j,\boldsymbol{x}}$, if $\boldsymbol{\Sigma}_1 \succeq \boldsymbol{\Sigma}_2$, we have $\operatorname{Tr} \boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_1) \geq \operatorname{Tr} \boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_2)$. Since $\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_1) = \boldsymbol{\Sigma}_{j,\boldsymbol{x}} - \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_1)^{-1} \boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}}$, $\boldsymbol{P}_j(\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_2) = \boldsymbol{\Sigma}_{j,\boldsymbol{x}} - \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_2)^{-1} \boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}}$, the conclusion is obvious due to basic algebra. □

APPENDIX F
QUANTIFY UNCERTAINTY USING WASSERSTEIN SETS

If we use the Wasserstein ambiguity sets for $\mathbb{P}_{j,\mathbf{x},\mathbf{y}}$ as in [13], the problem (33) is particularized into

$$\begin{array}{ll} \underset{\substack{\boldsymbol{\Sigma}_{j,\boldsymbol{x}} \\ \boldsymbol{\Sigma}_{j,\boldsymbol{v}}}}{\max} & \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}] \\ \textit{s.t.} & \left\{ \begin{array}{l} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}), \\ \sqrt{\operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{M}_j - 2(\boldsymbol{M}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{M}_j^{\frac{1}{2}})^{\frac{1}{2}}]} \leq \theta_{j,\boldsymbol{x}}, \\ \sqrt{\operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{v}} + \boldsymbol{R}_j - 2(\boldsymbol{R}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \boldsymbol{R}_j^{\frac{1}{2}})^{\frac{1}{2}}]} \leq \theta_{j,\boldsymbol{v}}, \\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \succeq \boldsymbol{0}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}. \end{array} \right. \end{array} \tag{50}$$

where $\theta_{j,\boldsymbol{x}} \geq 0$ and $\theta_{j,\boldsymbol{v}} \geq 0$. Problem (50) is a nonlinear positive semi-definite program (SDP). However, it has a linear reformulation, which can be efficiently solved.

Proposition 6: The nonlinear positive SDP (50) can be reformulated into a linear positive SDP

$$\begin{array}{ll} \underset{\substack{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \\ \boldsymbol{V}_{j,\boldsymbol{x}}, \boldsymbol{V}_{j,\boldsymbol{v}}}}{\max} & \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}] \\ \textit{s.t.} & \left\{ \begin{array}{l} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}), \\ \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{M}_j - 2\boldsymbol{V}_{j,\boldsymbol{x}}] \leq \theta_{j,\boldsymbol{x}}^2, \\ \left[\begin{array}{cc} \boldsymbol{M}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{M}_j^{\frac{1}{2}} & \boldsymbol{V}_{j,\boldsymbol{x}} \\ \boldsymbol{V}_{j,\boldsymbol{x}} & \boldsymbol{I} \end{array} \right] \succeq \boldsymbol{0}, \\ \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{v}} + \boldsymbol{R}_j - 2\boldsymbol{V}_{j,\boldsymbol{v}}] \leq \theta_{j,\boldsymbol{v}}^2, \\ \left[\begin{array}{cc} \boldsymbol{R}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \boldsymbol{R}_j^{\frac{1}{2}} & \boldsymbol{V}_{j,\boldsymbol{v}} \\ \boldsymbol{V}_{j,\boldsymbol{v}} & \boldsymbol{I} \end{array} \right] \succeq \boldsymbol{0}, \\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \succeq \boldsymbol{0}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}, \boldsymbol{V}_{j,\boldsymbol{x}} \succeq \boldsymbol{0}, \boldsymbol{V}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}. \end{array} \right. \end{array} \tag{51}$$

Proof: By introducing $\boldsymbol{M}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{M}_j^{\frac{1}{2}} \succeq \boldsymbol{V}_{j,\boldsymbol{x}}^2$ where $\boldsymbol{V}_{j,\boldsymbol{x}} \succeq \boldsymbol{0}$ and using Schur complement, the nonlinear constraint $\sqrt{\operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{M}_j - 2(\boldsymbol{M}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{M}_j^{\frac{1}{2}})^{\frac{1}{2}}]} \leq \theta_{j,\boldsymbol{x}}$ in (50) can be reformulated into a linear equivalent

$$\left\{ \begin{array}{ll} \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{M}_j - 2\boldsymbol{V}_{j,\boldsymbol{x}}] & \leq \theta_{j,\boldsymbol{x}}^2, \\ \left[\begin{array}{cc} \boldsymbol{M}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{M}_j^{\frac{1}{2}} & \boldsymbol{V}_{j,\boldsymbol{x}} \\ \boldsymbol{V}_{j,\boldsymbol{x}} & \boldsymbol{I} \end{array} \right] & \succeq \boldsymbol{0}, \\ \boldsymbol{V}_{j,\boldsymbol{x}} & \succeq \boldsymbol{0}. \end{array} \right.$$

The constraint $\sqrt{\operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{v}} + \boldsymbol{R}_j - 2(\boldsymbol{R}_j^{\frac{1}{2}} \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \boldsymbol{R}_j^{\frac{1}{2}})^{\frac{1}{2}}]} \leq \theta_{j,\boldsymbol{v}}$ in (50) can be handled similarly. This completes the proof. □

Remark 7: Note that in the Wasserstein case, (34) in Line 5 of Algorithm 1 should be replaced with (51). □

APPENDIX G
PROOF OF THEOREM 2

To begin with, we drop the first constraint $\boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})$ and solve the following relaxed problem:

$$\begin{array}{ll} \underset{\boldsymbol{\Sigma}_{j,\boldsymbol{x}}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}}}{\max} & \operatorname{Tr}[\boldsymbol{\Sigma}_{j,\boldsymbol{x}} + \boldsymbol{\Sigma}_{j,\boldsymbol{v}}] \\ \textit{s.t.} & \left\{ \begin{array}{l} (1 - \theta_{j,\boldsymbol{x}}) \boldsymbol{M}_j \preceq \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \preceq (1 + \theta_{j,\boldsymbol{x}}) \boldsymbol{M}_j, \\ (1 - \theta_{j,\boldsymbol{v}}) \boldsymbol{R}_j \preceq \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \preceq (1 + \theta_{j,\boldsymbol{v}}) \boldsymbol{R}_j, \\ \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \succeq \boldsymbol{0}, \boldsymbol{\Sigma}_{j,\boldsymbol{v}} \succeq \boldsymbol{0}. \end{array} \right. \end{array}$$

The solution of this relaxed problem is obviously given in (35). By verifying that the solution (35) also satisfies the constraint $\boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top = \boldsymbol{K}_j \cdot (\boldsymbol{H}_j \boldsymbol{\Sigma}_{j,\boldsymbol{x}} \boldsymbol{H}_j^\top + \boldsymbol{\Sigma}_{j,\boldsymbol{v}})$ where $\boldsymbol{K}_j$ is defined in (29), we complete the proof. □

REFERENCES

[1] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[2] ——, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[3] Y. Bar-Shalom, S. Challa, and H. A. Blom, "IMM estimator versus optimal estimator for hybrid systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 986–991, 2005.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[5] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[6] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[7] V. P. Jilkov and X. R. Li, "Online Bayesian estimation of transition probabilities for markovian jump systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1620–1630, 2004.

[8] U. Orguner and M. Demirekler, "Maximum likelihood estimation of transition probabilities of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5093–5108, 2008.

[9] S. Zhao and F. Liu, "Recursive estimation for Markov jump linear systems with unknown transition probabilities: A compensation approach," *Journal of the Franklin Institute*, vol. 353, no. 7, pp. 1494–1517, 2016.

[10] Y. Ma, S. Zhao, and B. Huang, "Multiple-model state estimation based on variational Bayesian inference," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1679–1685, 2018.

[11] A. Doucet and B. Ristic, "Recursive state estimation for multiple switching models with unknown transition probabilities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 1098–1104, 2002.

[12] U. Orguner and M. Demirekler, "An online sequential algorithm for the estimation of transition probabilities for jump Markov linear systems," *Automatica*, vol. 42, no. 10, pp. 1735–1744, 2006.

[13] S. Wang, Z. Wu, and A. Lim, "Robust state estimation for linear systems under distributional uncertainty," *IEEE Transactions on Signal Processing*, 2021.

[14] S. Wang and Z.-S. Ye, "Distributionally robust state estimation for linear systems subject to uncertainty and outlier," *IEEE Transactions on Signal Processing*, vol. 70, pp. 452–467, 2021.

[15] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for linear dynamic models with dirichlet process mixtures," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 71–84, 2007.

[16] C. Magnant, A. Giremus, E. Grivel, L. Ratton, and B. Joseph, "Bayesian non-parametric methods for dynamic state-noise covariance matrix estimation: Application to target tracking," *Signal Processing*, vol. 127, pp. 135–150, 2016.

[17] R. Guo, M. Shen, D. Huang, X. Yin, and L. Xu, "Recursive estimation of transition probabilities for jump Markov linear systems under minimum Kullback–Leibler divergence criterion," *IET Control Theory & Applications*, vol. 9, no. 17, pp. 2491–2499, 2015.

[18] L. Zhang and E.-K. Boukas, "Mode-dependent $\mathcal{H}_\infty$ filtering for discrete-time Markovian jump linear systems with partly unknown transition probabilities," *Automatica*, vol. 45, no. 6, pp. 1462–1467, 2009.

[19] X. Li, J. Lam, H. Gao, and J. Xiong, "$\mathcal{H}_\infty$ and $\mathcal{H}_2$ filtering for linear systems with uncertain Markov transitions," *Automatica*, vol. 67, pp. 252–266, 2016.

[20] C. E. de Souza, A. Trofino, and K. A. Barbosa, "Mode-independent $\mathcal{H}_\infty$ filters for Markovian jump linear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1837–1841, 2006.

[21] U. Orguner and M. Demirekler, "Risk-sensitive filtering for jump markov linear systems," *Automatica*, vol. 44, no. 1, pp. 109–118, 2008.

[22] S. Zhao and F. Liu, "An adaptive risk-sensitive filtering method for markov jump linear systems with uncertain parameters," *Journal of the Franklin Institute*, vol. 349, no. 6, pp. 2047–2064, 2012.

[23] M. J. Schervish, *Theory of Statistics*. Springer Science & Business Media, 2012.

[24] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[25] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[26] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[27] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces. I. theory," *IEEE Transactions on Automatic Control*, vol. 41, no. 1, pp. 18–33, 1996.

[28] B. C. Levy and R. Nikoukhah, "Robust state space filtering under incremental model perturbations subject to a relative entropy tolerance," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 682–695, 2013.

[29] S. Shafieezadeh Abadeh, V. A. Nguyen, D. Kuhn, and P. M. Mohajerin Esfahani, "Wasserstein distributionally robust kalman filtering," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[30] S. Wang, "Distributionally robust state estimation," Ph.D. dissertation, National University of Singapore, 1 2022.

[31] A. Ben-Tal, D. den Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, pp. 341–357, 2 2013.

[32] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1-2, pp. 115–166, 2018.

[33] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*. Informs, 2019, pp. 130–166.

[34] R. Gao and A. Kleywegt, "Distributionally robust stochastic optimization with wasserstein distance," *Mathematics of Operations Research*, vol. 48, no. 2, pp. 603–655, 2022.

[35] R. Zhu, H. Wei, and X. Bai, "Wasserstein metric based distributionally robust approximate framework for unit commitment," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2991–3001, 2019.

[36] B. Zhou, G. Chen, T. Huang, Q. Song, and Y. Yuan, "Planning pev fast-charging stations using data-driven distributionally robust optimization approach based on ϕ-divergence," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 1, pp. 170–180, 2020.

[37] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Pptimization*. Cambridge University Press, 2004.

[38] G. Wang, "ML estimation of transition probabilities in jump Markov systems via convex optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1492–1502, 2010.

[39] S. Wang, "Distributionally robust state estimation for nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4408–4423, 2022.

[40] P. Teunissen and A. Khodabandeh, "Review and principles of PPP-RTK methods," *Journal of Geodesy*, vol. 89, no. 3, pp. 217–240, 2015.

[41] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *International Conference on Machine Learning*. PMLR, 2013, pp. 427–435.

Dr. Shixiong Wang received the B.Eng. degree in detection, guidance, and control technology, and the M.Eng. degree in systems and control engineering from the School of Electronics and Information, Northwestern Polytechnical University, China, in 2016 and 2018, respectively. He received his Ph.D. degree from the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, in 2022.

He is currently a Postdoctoral Research Associate with the Intelligent Transmission and Processing Laboratory, Imperial College London, London, United Kingdom, from May 2023. He was a Postdoctoral Research Fellow with the Institute of Data Science, National University of Singapore, Singapore, from March 2022 to March 2023.

His research interest includes statistics and optimization theories with applications in signal processing (especially optimal estimation theory), machine learning (especially generalization error theory), and control technology.

Supplementary Materials

APPENDIX H
ROBUSTIFICATION OVER PRIOR MODEL PROBABILITY

In this appendix, we discuss the case where we solve the genuine problem (24) with respect to $\boldsymbol{\omega}$, rather than the simplified case with respect to $\boldsymbol{\mu}$. In consideration of the high computational complexity, the genuine problem (24) with respect to $\boldsymbol{\omega}$ is not investigated in the main body of the paper.

Proposition 7: If the model set is exact and only the prior model probability vector $\boldsymbol{\omega}$ is uncertain [i.e., the special ambiguity set (23) is investigated], the reformulated distributionally robust Bayesian estimation problem (24) can be further reformulated into a tractable quadratic fractional program

$$\begin{aligned} \max_{\boldsymbol{\omega}} \quad & -\frac{\boldsymbol{\omega}^\top(\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}-\boldsymbol{p}\boldsymbol{b}^\top\boldsymbol{C})\boldsymbol{\omega}}{\boldsymbol{\omega}^\top\boldsymbol{p}\boldsymbol{p}^\top\boldsymbol{\omega}} \\ \textit{s.t.} \quad & \begin{cases} \sum_{j=1}^{N}\omega_j = 1, \\ \omega_j \geq 0, \qquad\qquad \forall j\in[N], \\ \Delta_0(\boldsymbol{\omega},\bar{\boldsymbol{\omega}}) \leq \theta_0, \end{cases} \end{aligned} \tag{52}$$

where $\boldsymbol{p} := [p_1(\boldsymbol{y}), p_2(\boldsymbol{y}), \ldots, p_N(\boldsymbol{y})]^\top$ denotes the likelihoods of the candidate models given the measurement $\boldsymbol{y}$ and $\boldsymbol{C} := \mathrm{diag}(\boldsymbol{p})$ is a diagonal matrix whose diagonal entries are elements of $\boldsymbol{p}$.

Proof: From (3), for every $j \in [N]$, we have $\mu_j = \frac{\omega_j p_j(\boldsymbol{y})}{\sum_{j=1}^{N}\omega_j p_j(\boldsymbol{y})} = \frac{\omega_j p_j(\boldsymbol{y})}{\boldsymbol{\omega}^\top\boldsymbol{p}}$, i.e., $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]^\top = \frac{\boldsymbol{C}\boldsymbol{\omega}}{\boldsymbol{\omega}^\top\boldsymbol{p}}$. Therefore, the problem (24) can be explicitly written as

$$\begin{aligned} \max_{\boldsymbol{\omega}} \quad & -\left(\frac{\boldsymbol{C}\boldsymbol{\omega}}{\boldsymbol{\omega}^\top\boldsymbol{p}}\right)^\top \boldsymbol{A}\left(\frac{\boldsymbol{C}\boldsymbol{\omega}}{\boldsymbol{\omega}^\top\boldsymbol{p}}\right) + \boldsymbol{b}^\top\left(\frac{\boldsymbol{C}\boldsymbol{\omega}}{\boldsymbol{\omega}^\top\boldsymbol{p}}\right) \\ \textit{s.t.} \quad & \begin{cases} \sum_{j=1}^{N}\omega_j = 1, \\ \omega_j \geq 0, \qquad\qquad \forall j\in[N], \\ \Delta_0(\boldsymbol{\omega},\bar{\boldsymbol{\omega}}) \leq \theta_0, \end{cases} \end{aligned} \tag{53}$$

which can be rearranged into the quadratic fractional program (52). □

The problem (52) can be written in a compact form

$$\max_{\boldsymbol{\omega}\in\Omega} \frac{f_1(\boldsymbol{\omega})}{f_2(\boldsymbol{\omega})}, \tag{54}$$

where $f_1(\boldsymbol{\omega}) := -\boldsymbol{\omega}^\top(\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}-\boldsymbol{p}\boldsymbol{b}^\top\boldsymbol{C})\boldsymbol{\omega}$ denotes the numerator of the objective of (52), $f_2(\boldsymbol{\omega}) := \boldsymbol{\omega}^\top\boldsymbol{p}\boldsymbol{p}^\top\boldsymbol{\omega}$ the denominator of the objective of (52), and Ω the feasible region of (52). One may verify that although $f_2(\boldsymbol{\omega})$ is convex, $f_1(\boldsymbol{\omega})$ is neither concave nor convex. However, $f_1(\boldsymbol{\omega}) \geq 0$ can be guaranteed because the objective of (19) is non-negative, as are those of (24) and (52). Complete (approximated) solutions to the problem (54) can be found in, e.g., [S1],[7] [S2],[8] where involved indefinite quadratic programs can be solved by the method in, e.g., [S3].[9] Numerically solving (54) is time-consuming due to the indefiniteness of $f_1(\boldsymbol{\omega})$. Therefore, in this paper, we do not proceed further for (54). Instead, we find a simplified alternative to the original problem (24) with respect to $\boldsymbol{\mu}$. Interested readers may implement solution methods in, e.g., [S3], to solve (54) themselves.

APPENDIX I
SOLUTION TO (26)

The Lagrangian of (26) is

$$\begin{aligned} \min_{\lambda_0\geq 0,\lambda_1}\max_{\boldsymbol{\mu}} \quad & -\boldsymbol{\mu}^\top\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}^\top\boldsymbol{\mu} + \lambda_1\cdot(1-\mathbf{1}^\top\boldsymbol{\mu}) + \\ & \lambda_0\cdot(\theta_0 - \boldsymbol{\mu}^\top\ln\boldsymbol{\mu} + \boldsymbol{\mu}^\top\ln\bar{\boldsymbol{\mu}}). \end{aligned} \tag{55}$$

For every $\lambda_0 \geq 0$ and λ_1, the maximum $\boldsymbol{\mu}$ satisfies the first-order optimality condition:

$$-2\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} - \lambda_1\cdot\mathbf{1} + \lambda_0\cdot(-\ln\boldsymbol{\mu} - \mathbf{1} + \ln\bar{\boldsymbol{\mu}}) = \mathbf{0}, \tag{56}$$

[7][S1] W. Dinkelbach, "On nonlinear fractional programming," Management Science, vol. 13, no. 7, pp. 492–498, 1967.

[8][S2] A. T. Phillips, Quadratic Fractional Programming: Dinkelbach Method.Boston, MA: Springer US, 2001, pp. 2107–2110. [Online]. Available: https://doi.org/10.1007/0-306-48332-7_406.

[9][S3] A. Phillips and J. Rosen, "Guaranteed ϵ-approximate solution for indefinite quadratic global minimization," Naval Research Logistics (NRL), vol. 37, no. 4, pp. 499–514, 1990.

which transforms (55) to

$$\min_{\lambda_0\geq 0,\lambda_1} \quad \lambda_0\theta_0+\lambda_1+\boldsymbol{\mu}^\top \boldsymbol{A}\boldsymbol{\mu}+\lambda_0\mathbf{1}^\top\boldsymbol{\mu}. \tag{57}$$

Since (26) is a convex program and $\bar{\boldsymbol{u}}$ is a relative interior point in the feasible set, there does not exist duality gap between (26) and (57). Since (57) is convex, any first-order gradient-based method, e.g., projected gradient descent, is applicable to solve it. Let the objective of (57) be denoted as $f(\boldsymbol{\lambda})$. From (56), we have $-2\boldsymbol{A}\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_0}=\ln\boldsymbol{\mu}+\mathbf{1}-\ln\bar{\boldsymbol{\mu}}+\lambda_0\frac{1}{\boldsymbol{\mu}}\odot\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_0}$, and $-2\boldsymbol{A}\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_1}=\mathbf{1}+\lambda_0\frac{1}{\boldsymbol{\mu}}\odot\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_1}$, where $\frac{1}{\boldsymbol{\mu}}$ means element-wise fraction, and $\odot$ denotes the Hadamard product (i.e., the element-wise product). The gradient of the objective of (57) with respect to λ_0 and λ_1 are given by

$$\begin{aligned}\frac{\partial f(\boldsymbol{\lambda})}{\partial\lambda_0} &= \theta_0+2\boldsymbol{\mu}^\top\boldsymbol{A}\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_0}+\mathbf{1}^\top\boldsymbol{\mu}+\lambda_0\mathbf{1}^\top\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_0}\\ &= \theta_0-\boldsymbol{\mu}^\top\ln\boldsymbol{\mu}+\boldsymbol{\mu}^\top\ln\bar{\boldsymbol{\mu}},\end{aligned} \tag{58}$$

and

$$\frac{\partial f(\boldsymbol{\lambda})}{\partial\lambda_1}=1+2\boldsymbol{\mu}^\top\boldsymbol{A}\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_0}+\lambda_0\mathbf{1}^\top\frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\lambda_1}=1-\mathbf{1}^\top\boldsymbol{\mu}. \tag{59}$$

respectively. Hence, when the optimality of (57) reaches, i.e., when the gradients with respect to λ_0 and λ_1 vanish, we have $1=\sum_{j=1}^N\mu_j$ and $\theta_0=\sum_{j=1}^N\mu_j\cdot\ln\frac{\mu_j}{\bar{\mu}_j}$. Specifically, it means $\boldsymbol{\mu}$ is indeed a distribution summed to unit and all the robustness budget θ_0 has been used. In summary, the solution to (26) is summarized in Algorithm 2. Since (26) is a convex program, every iteration improves the objective.

Algorithm 2 Solution to (26)

Definition: S as maximum allowed iteration steps and s the current iteration step; α as step size; ϵ as numerical precision threshold; $\mathrm{abs}(\cdot)$ returns absolute value.

Remark: Since (57) is convex, any initial values for $\lambda_0\geq 0$ and λ_1 are acceptable. If early stopping is applied (i.e., S is not sufficiently large for time-saving purpose), a normalization procedure is necessary to guarantee $1=\sum_j\mu_j$.

Input: $S,\alpha,\epsilon,\lambda_0,\lambda_1$

$s\leftarrow 0$;
while true **do**
  // Update $\boldsymbol{\mu}$
  Solve N-variable nonlinear root-finding sub-problem (56) to obtain $\boldsymbol{\mu}^{(s)}$ with current λ_0 and λ_1 (see Remark 8)
  // Gradient Descent to Update λ_0 *and* λ_1
  $\lambda_0\leftarrow\lambda_0-\alpha\cdot\frac{\partial f(\boldsymbol{\lambda})}{\partial\lambda_0}$ *// See (58)*
  $\lambda_1\leftarrow\lambda_1-\alpha\cdot\frac{\partial f(\boldsymbol{\lambda})}{\partial\lambda_1}$ *// See (59)*
  // Projection
  if $\lambda_0<0$ **then** $\lambda_0\leftarrow 0$
  end if
  // Next Iteration
  $s\leftarrow s+1$
  // Stopping Rule
  if $s>S$ **or** $\mathrm{abs}(\frac{\partial f(\boldsymbol{\lambda})}{\partial\lambda_1})<\epsilon$ **then**
    if $1\neq\sum_i\mu_i^{(s)}$ **then** *// Early Stopping Applied*
      $\mu_i^{(s)}\leftarrow\mu_i^{(s)}/\sum_j\mu_j^{(s)},\quad\forall i\in[N],$
    end if
    break while
  end if
end while

Output: $\boldsymbol{\mu}^{(s)}$

Remark 8: We discuss the N-variate root-finding problem $-2\boldsymbol{A}\boldsymbol{\mu}+\boldsymbol{b}-\lambda_1\cdot\mathbf{1}+\lambda_0\cdot(-\ln\boldsymbol{\mu}-\mathbf{1}+\ln\bar{\boldsymbol{\mu}})=\mathbf{0}$ on $\boldsymbol{\mu}\geq\mathbf{0}$. Let $\boldsymbol{g}(\boldsymbol{\mu}):=-2\boldsymbol{A}\boldsymbol{\mu}+\boldsymbol{b}-\lambda_1\cdot\mathbf{1}+\lambda_0\cdot(-\ln\boldsymbol{\mu}-\mathbf{1}+\ln\bar{\boldsymbol{\mu}})$. One may verify that $\mathrm{d}\boldsymbol{g}(\boldsymbol{\mu})/\mathrm{d}\boldsymbol{\mu}\prec\mathbf{0}$ (i.e., $\boldsymbol{g}$ is a monotonically decreasing function in $\boldsymbol{\mu}$), $\boldsymbol{g}(\mathbf{0})\to\boldsymbol{\infty}$, and $\boldsymbol{g}(\boldsymbol{\infty})\to-\boldsymbol{\infty}$. Therefore, at least one root of $\boldsymbol{g}(\boldsymbol{\mu})=\mathbf{0}$ exists and Newton's method can be used to find it. □

Remark 9: If the 2-norm constraint $\|\boldsymbol{\mu}-\bar{\boldsymbol{\mu}}\|_2\leq\theta_0$ is used to replace the KL divergence constraint, then the root-finding procedure would be significantly simplified. Therefore, in practice, to save computational time, one may choose the 2-norm constraint $(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^\top(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})\leq\theta_0^2$. Another choice to reduce the computational complexity is to use the Frank-Wolfe method (i.e., linearization of the objective function) as in Proposition 5. □

APPENDIX J
THE STANDARD IMM FILTER

The implementation details of the interactive multiple model (IMM) method is given in Algorithm 3. The results in Step 2 (see Line 25) are due to (2) and (4) where $\mu_{j,k|k-1}$ and $\mu_{j,k|k}$ are prior and posterior model probabilities of the j^{th} model, respectively. The prior model probability, model likelihood, and posterior model probability of the j^{th} model are calculated in Step 1.5 (see Line 18), Step 1.6 (see Line 20), and Step 1.7 (see Line 22), respectively. See [3], [5] (in the reference list of the main body of the paper) for more information.

Algorithm 3 Interactive Multiple Model Algorithm [3], [5]

Definition: Let $\hat{\boldsymbol{x}}_{j,k|k-1}$ denote the prior state estimate provided by the j^{th} model and $\boldsymbol{P}_{j,k|k-1}$ the corresponding state estimation error covariance. Let $\hat{\boldsymbol{x}}_{j,k|k}$ denote the posterior state estimate provided by the j^{th} model and $\boldsymbol{P}_{j,k|k}$ the corresponding state estimation error covariance; Let $\hat{\boldsymbol{x}}_{k|k}$ denote the combined posterior state estimate of the N models and $\boldsymbol{P}_{k|k}$ the corresponding state estimation error covariance; Let $\mu_{j,k|k-1}$ and $\mu_{j,k|k}$ be the prior and posterior model probability of the j^{th} model at the time k, respectively; Let $\{\pi_{ij}\}_{i,j=1,2,\ldots,N}$ be the model transition probability matrix.
Initialization: $\forall j \in [N]$, initialize $\mu_{j,0|0}$, $\hat{\boldsymbol{x}}_{j,0|0}$, and $\boldsymbol{P}_{j,0|0}$.
Remark: In literature, prior and posterior state estimate are also known as predicted and updated state estimate, respectively.

Input: $\boldsymbol{y}_k$, $k = 1, 2, 3, \ldots$
1: **while** true **do**
2: *// (Step 1) At Time k*
3: **for** $j = 1 : N$ **do**
4: *// (Step 1.1) Transition Probability From i^{th} Model at Time $k-1$ To j^{th} Model at Time k*
5: $\mu_{ij,k|k-1} = \frac{\pi_{ij} \cdot \mu_{i,k-1|k-1}}{\sum_{i=1}^{N} \pi_{ij} \cdot \mu_{i,k-1|k-1}}$
6: *// (Step 1.2) Initialize the j^{th} Filter*
7: $\hat{\boldsymbol{x}}^0_{j,k-1|k-1} = \sum_{i=1}^{N} \mu_{ij,k|k-1} \cdot \hat{\boldsymbol{x}}_{i,k-1|k-1}$
8: $\boldsymbol{P}^0_{j,k-1|k-1} = \sum_{i=1}^{N} \mu_{ij,k|k-1} \cdot \Big\{ \boldsymbol{P}_{i,k-1|k-1} + (\hat{\boldsymbol{x}}_{i,k-1|k-1} - \hat{\boldsymbol{x}}^0_{j,k-1|k-1})(\hat{\boldsymbol{x}}_{i,k-1|k-1} - \hat{\boldsymbol{x}}^0_{j,k-1|k-1})^\top \Big\}$
9: *// (Step 1.3) Prior Estimation of the j^{th} Filter (i.e., Time Update)*
10: $\hat{\boldsymbol{x}}_{j,k|k-1} = \boldsymbol{F}_{j,k-1} \hat{\boldsymbol{x}}^0_{j,k-1|k-1}$
11: $\boldsymbol{P}_{j,k|k-1} = \boldsymbol{F}_{j,k-1} \boldsymbol{P}^0_{j,k-1|k-1} \boldsymbol{F}^\top_{j,k-1} + \boldsymbol{G}_{j,k-1} \boldsymbol{Q}_{j,k-1} \boldsymbol{G}^\top_{j,k-1}$
12: *// (Step 1.4) Posterior Estimation of the j^{th} Filter (i.e., Measurement Update)*
13: $\boldsymbol{r}_{j,k} = \boldsymbol{y}_k - \boldsymbol{H}_{j,k} \hat{\boldsymbol{x}}_{j,k|k-1}$ *// Innovation*
14: $\boldsymbol{S}_{j,k} = \boldsymbol{H}_{j,k} \boldsymbol{P}_{j,k|k-1} \boldsymbol{H}^\top_{j,k} + \boldsymbol{R}_{j,k}$ *// Innovation Covariance*
15: $\boldsymbol{K}_{j,k} = \boldsymbol{P}_{j,k|k-1} \boldsymbol{H}^\top_{j,k} \boldsymbol{S}^{-1}_{j,k}$ *// Filter Gain*
16: $\hat{\boldsymbol{x}}_{j,k|k} = \hat{\boldsymbol{x}}_{j,k|k-1} + \boldsymbol{K}_{j,k} \cdot \boldsymbol{r}_{j,k} = \hat{\boldsymbol{x}}_{j,k|k-1} + \boldsymbol{P}_{j,k|k-1} \boldsymbol{H}^\top_{j,k} \boldsymbol{S}^{-1}_{j,k} \cdot \big[\boldsymbol{y}(k) - \boldsymbol{H}_{j,k} \hat{\boldsymbol{x}}_{j,k|k-1}\big]$
17: $\boldsymbol{P}_{j,k|k} = \boldsymbol{P}_{j,k|k-1} - \boldsymbol{P}_{j,k|k-1} \boldsymbol{H}^\top_{j,k} \boldsymbol{S}^{-1}_{j,k} \boldsymbol{H}^\top_{j,k} \boldsymbol{P}_{j,k|k-1}$
18: *// (Step 1.5) Prior Probability of the j^{th} Model*
19: $\mu_{j,k|k-1} = \sum_{i=1}^{N} \pi_{ij} \cdot \mu_{i,k-1|k-1}$
20: *// (Step 1.6) Likelihood of the j^{th} Model*
21: $\lambda_{j,k} = \mathcal{N}_n(\boldsymbol{r}_{j,k}; \boldsymbol{0}, \boldsymbol{S}_{j,k})$
22: *// (Step 1.7) Posterior Probability of the j^{th} Model*
23: $\mu_{j,k|k} = \frac{\mu_{j,k|k-1} \cdot \lambda_{j,k}}{\sum_{i=1}^{N} \mu_{j,k|k-1} \cdot \lambda_{j,k}}$
24: **end for**
25: *// (Step 2) Combined Posterior State Estimate*
26: $\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^{N} \mu_{j,k|k} \cdot \hat{\boldsymbol{x}}_{j,k|k}$
27: $\boldsymbol{P}_{k|k} = \sum_{j=1}^{N} \mu_{j,k|k} \cdot \big\{ \boldsymbol{P}_{j,k|k} + (\hat{\boldsymbol{x}}_{j,k|k} - \hat{\boldsymbol{x}}_{k|k})(\hat{\boldsymbol{x}}_{j,k|k} - \hat{\boldsymbol{x}}_{k|k})^\top \big\}$
28: *// (Step 3) Next Time Step*
29: $k \leftarrow k + 1$
30: **end while**
Output: $\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}, \mu_{j,k|k}$