Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method

Sun, Junzi; Blom, Henk; Ellerbroek, Joost; Hoekstra, Jacco

Publication date
2018

Document Version
Final published version

Published in
2018 International Conference on Research in Air Transportation

Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method

Junzi Sun[1], Henk A.P. Blom[1,2], Joost Ellerbroek[1], and Jacco M. Hoekstra[1]

[1] Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands
[2] Air Transport Safety Institute, National Aerospace Laboratory, the Netherlands

***Abstract*—This paper focuses on estimating aircraft mass and thrust setting using a recursive Bayesian method called particle filtering. The method is based on a nonlinear state-space system derived from aircraft point-mass performance models. Using solely ADS-B and Mode-S data, flight states such as position, velocity, and wind speed are collected and used for the estimation. An important aspect of particle filtering is noise modeling. Four noise models are proposed in this paper based on the native ADS-B Navigation Accuracy Category (NAC) parameters. Simulations, experiments, and validation, based on a number of flights are carried out to test the theory. As a result, convergence of the estimation can usually be obtained within 30 seconds for any climbing flight. The method proposed in this paper not only provides final estimates, but also defines the limits of noise above which estimation of mass and thrust becomes impossible. When validated with a dataset consisting of the measured true mass and thrust of 50 Cessna Citation II flights, the stochastic recursive Bayesian approach proposed in this paper yields a mean absolute error of 4.6%.**

Keywords - aircraft, state estimation, point-mass model, measurement noise, particle filter, Bayesian estimation

I. Introduction

Estimating aircraft mass based on observed flight trajectory data has long been a topic of interest in ATM research. Aircraft mass acts as not only as an important parameter for many different studies of aircraft performance, but is also a desired piece of knowledge for air traffic controllers in practice. Airlines, however, treat this data as confidential, and access is rarely given to either researchers or air traffic controllers. In practice, this means this data is not accessible nor actively used by the research community.

Earlier studies that addressed this problem were commonly focused on deterministic methods based on the aircraft total energy model (TEM), which is also the core of the BADA performance calculation [1]. Alligier et al. presented a least-squares method [2] and a follow-on machine learning method [3]. Around the same time, Schultz et al. implemented an adaptive estimation method to estimate the aircraft mass [4]. These methods used accurate radar data for estimation.

In our previous studies, we also proposed two different approaches based on ADS-B data. The first study made use of the data from the takeoff phase [5]. The second method used Bayesian inference to construct a posteriori estimation by combining masses computed from different flight phases [6].

From all these studies, a strong link between aircraft mass and thrust setting is evident, and it is not possible to estimate one of the two parameters without some knowledge of the other. Most of the aforementioned studies essentially addressed the estimation process as an optimization problem. Their solutions are obtained using a form of least-squares fitting. Aircraft mass estimated under these conditions is often unrealistic and even outside plausible physical boundaries. This is often due to the uncertainty in the trajectory data, as well as the uncertainty in the system equations. Although the effect of noise is an important aspect in the entire inference process, its relation to mass estimation has not yet been studied comprehensively by previous studies. It is therefore one of the main contributions of this paper.

The problem of mass estimation by a (ground-based) observer can be considered as having to solve an inversed non-linear multi-state system using noisy observation data. To tackle this complex system, we constructed a detailed point-mass flight performance model with ten system states and eight observables in this paper. Then, a tailored, Sample Importance Re-sampling (SIR) particle filter, is introduced to solve these system equations. In addition, four different levels of observation noise models are constructed, based on ADS-B navigation accuracy standards, which are used in the particle filtering.

Both simulated and real flights are used to test the model and method. After that, a set of actual flight data with known initial mass are used for validation. Finally, the paper also offers conclusions on the relationships between estimation and uncertainties based on our model and method.

The remainder of the paper is structured as follows. Section two describes the fundamentals of recursive Bayesian theory, particle filtering, the point-mass flight dynamic system equations, noise models, and the detailed algorithm. Section three presents our experiments using simulated and real flight data. Section four offers the validation results based on real flight data. Finally, a discussion and conclusions are presented in sections five and six.

II. Recursive Bayesian estimation and its application to aircraft state estimation

Aircraft flight dynamics are complex, non-linear, and multivariate. Estimation of parameters such as mass depends on solving this fairly complex inversed flight dynamic system. Due to the high non-linearity and number of system states, deriving the estimation of mass under noisy measurements (and thrust setting when possible) is the main goal of this section.

To this end, we will first give a fundamental introduction to the use of the recursive Bayesian method for generic system state estimation. The Sample Importance Re-sampling (SIR) particle filtering method is introduced for the estimation purposes. Then, the specific system of aircraft flight dynamic equations will be addressed. Combining with the SIR, the solution for the system will be given. A detailed algorithm implementation will also be provided. This section will end with the definition of the observation noise models and other stochastic elements of the SIR.

Recursive Bayesian estimation is a probabilistic method for system states filtering, prediction, and estimation. One of the most widely used applications is the *Particle Filter*, which is technically known as the Sequential Monte Carlo Method.

Starting for a set of initial state conditions, the recursive Bayesian determines the values for the next set of states based on the joint posterior probability of all previous states' measurements. Commonly, for non-linear system with a large number of states, the computation of a closed-form joint posterior probability is not practical.

For this reason, the Monte Carlo method is introduced. It utilizes a very large number of weighted particles to approximate the true probability density functions, where each particle represents a possible set of state values. Through a stochastic update and sampling processes, the original challenge

of updating a high-dimensional joint posterior probability is then transformed to adapting the weights of particles.

A. Recursive Bayesian estimation

First, it is crucial to explain the basics of how recursive Bayesian theory is used for system state estimation.

Denoting $\mathbf{x}_t$ as the system states and $\mathbf{y}_t$ as observables at time t ($t \in \mathbb{N}$), the evolution of the discrete-time state model and observation model of a system can be generalized as follows:

$$\begin{aligned} \mathbf{x}_t &= \boldsymbol{f}(\mathbf{x}_{t-1},\ \boldsymbol{\nu}_{t-1}) \\ \mathbf{y}_t &= \boldsymbol{h}(\mathbf{x}_t,\ \boldsymbol{n}_t) \end{aligned} \tag{1}$$

where $\boldsymbol{f}$ and $\boldsymbol{h}$ represent the state transition function and observation function, $\boldsymbol{\nu}_t$ is the process noise, and $\boldsymbol{n}_t$ is the observation noise. $\boldsymbol{\nu}_t$ and $\boldsymbol{n}_t$ are assumed to be mutually independent sequence of independent and identically distributed variables.

Equation 1 represents the general case of the system and observation function for recursive Bayesian methods. In the particular case of this paper, we assume an additive Gaussian model for both process noise and observation noise. We further assume the noises extends linearly and equation 1 is rewritten as

$$\begin{aligned} \mathbf{x}_t &= \boldsymbol{f}(\mathbf{x}_{t-1}) + \boldsymbol{\nu}_{t-1} \\ \mathbf{y}_t &= \boldsymbol{h}(\mathbf{x}_t) + \boldsymbol{n}_t \end{aligned} \tag{2}$$

Regardless of the form of system and observation functions, the goal of filtering is to compute the probability of system states at any time t based on the observation from time 1 to time t, denoted as $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \tag{3}$$

where the first part $p(\mathbf{y}_t|\mathbf{x}_t)$ is the measuring probability that can be computed based on observation noise $\boldsymbol{n}$. Due to Equation 2, $\mathbf{x}_t$ is a first order Markov process, hence the second part $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ becomes:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\ d\mathbf{x}_{t-1} \tag{4}$$

where the first term $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the state transition probability. It can be computed by the system transaction equation based on the process noise model $\boldsymbol{\nu}$. Combining the above two equations, the recursive form becomes:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\ d\mathbf{x}_{t-1} \tag{5}$$

where the denominator of the fraction is a normalizing factor which does not need to be computed explicitly. The difficulty is to compute the measuring probability and transition probability. In most cases an analyticity solution is not possible. That's where Sequential Monte Carlo (SMC) simulation comes to work.

B. Particle filtering

A particle filter is a recursive Bayesian estimator based on importance sampling that computes the posterior density in a Monte Carlo fashion. It approximates the target distribution $p(x)$ using a large number of samples (particles), drawn from an proposal distribution $q(x)$ and updates it recursively.

To describe the SMC process, at time t, let $\{\mathbf{x}_t^i, \mathrm{w}_t^i\}_{i=i}^N$ be state particles that can represent the posterior density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, where $\{\mathbf{x}_t^i\}$ is the set of states with weights $\{\mathrm{w}_t^i\}$. Henceforth, the posterior density is approximated with the empirical probability density:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \mathrm{w}_t^i \delta(\mathbf{x}_t^i) \tag{6}$$

where $\delta(\cdot)$ is a Dirac delta function centered at $\mathbf{x}_t^i$ and w^i is the normalized weight of a particle which satisfies $\mathrm{w}^i = p(x^i)/q(x^i)$. The most important part of particle filtering is the measurement updating. Sequentially, the particle weight w_t^i is updated in a recursive form. The solution is presented in Equation 7, as derived in [7] :

$$\begin{aligned} \mathbf{x}_t^i &\sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t}) \\ \tilde{\mathrm{w}}_t^i &\propto \mathrm{w}_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t})} \\ \mathrm{w}_t^i &= \frac{\tilde{\mathrm{w}}_t^i}{\sum_{i=1}^N \tilde{\mathrm{w}}_t^i} \end{aligned} \tag{7}$$

At each iteration, the sum of all weights is normalized to one as shown in the last equation. The posterior filtered density is approximated using Equation 6. We can also compute the expected value of the states at each time step using the particle weights:

$$\mathbb{E}[\mathbf{x}_t] = \sum_{i=1}^{N} \mathbf{x}_t^i \mathrm{w}_t^i \tag{8}$$

There are different ways to choose the proposal distribution. A specific particle filter - *Sample Importance Re-sampling* (SIR) - is used for solving the problem of this paper. The SIR particle filter uses the state transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$ as the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \mathbf{y}_k)$. Therefore, the particle update equations in Equation 7 can be simplified as:

$$\begin{aligned} \mathbf{x}_t^i &\sim p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i) \\ \mathrm{w}_t^i &\propto p(\mathbf{y}_t|\mathbf{x}_t^i) \end{aligned} \tag{9}$$

For a SIR particle filter, an additional re-sampling process at each iteration is included. The re-sampling step generates a new set of particles based on the approximated $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$. Weights of all particles are also assigned to $1/N$ after the re-sampling.

This step is essentially a redistribution of particles, which replaces low-weight particles with high-weight particles. The standard re-sampling algorithm is called *residual re-sampling*, which was proposed by Liu et al. in 1998 [8]. Other forms of re-sampling such as *systematic re-sampling* and *stratified re-sampling* are summarized by Douc and Cappe in 2005 [9]. In this paper, residual re-sampling is used.

III. System of point-mass aircraft performance model

A. Aircraft state

In previous section, the general form of the state system and solution using particle filter were given. In this section, the specific system equations based on point-mass aircraft performance are proposed. To illustrate, Figure 1 shows the observable states in all three axes. The left figure shows a horizontal projection of a trajectory, and the right figure a vertical projection. For ease of computation, latitudes, longitudes, and altitudes are converted to a reference Cartesian coordinate system.

Denoting aircraft mass as m, thrust setting coefficient as η, distance flown as $\vec{s}$, altitude as z, ground speed as $\vec{v}_g$, true airspeed as $\vec{v}_a$ (in horizontal projection), vertical rate as (v_z), and wind speed as $\vec{v}_w$, the system state $\mathbf{x}$ vector is:

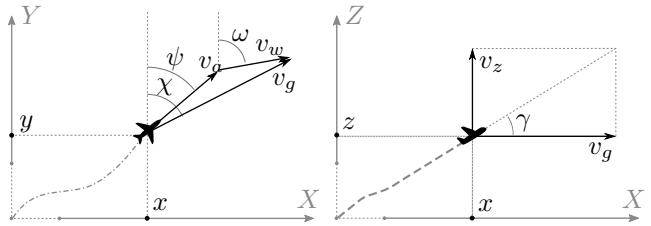

Fig. 1: Observed states in the flight dynamic system

$$\mathbf{x} = (m, \eta, \vec{s}, z, \vec{v}_g, \vec{v}_a, v_z, \vec{v}_w) \tag{10}$$

The vector variables in corresponding x and y axes components are:

$$\begin{aligned} \vec{s} &= (x, y) \\ \vec{v}_g &= (v_{gx}, v_{gy}) \\ \vec{v}_a &= (v_{ax}, v_{ay}) \\ \vec{v}_w &= (v_{wx}, v_{wy}) \end{aligned} \tag{11}$$

Additional angular parameters in Figure 1 are flight path angle γ, ground track χ, aircraft heading ψ, and wind direction ϕ. The measurement vector $\mathbf{y}$ is represented as:

$$\mathbf{y} = (\tilde{\vec{s}}, \tilde{z}, \tilde{\vec{v}}_g, \tilde{v}_z, \tilde{\vec{v}}_w) \tag{12}$$

B. State evolution

Since the goal is to model the system as accurately as possible and shift the uncertainty to the observation noise model, when possible we assume zero process noise for m_t, η_t, $\vec{s}_t$, z_t, and $\vec{v}_{a,t}$. For states that the perfect process model cannot establish accurately (or are unknown), we use an auto-regressive model to describe their process equations. These states are the vertical rate v_z and wind $\vec{v}_w$. The complete state process equations can be described as follows:

$$\begin{aligned} m_t &= m_{t-1} && (13)\\ \eta_t &= \eta_{t-1} && (14)\\ \vec{s}_t &= \vec{s}_{t-1} + (\vec{v}_{a,t-1} + \vec{v}_{w,t-1})\ \mathrm{dt} && (15)\\ z_t &= z_{t-1} + v_{z,t-1}\ \mathrm{dt} && (16)\\ \vec{v}_{a,t} &= \vec{v}_{a,t-1} + \vec{a}_{t-1}\ \mathrm{dt} && (17)\\ v_{z,t} &= \alpha_{vz} v_{z,t-1} + \varepsilon_{vz} && (18)\\ \vec{v}_{w,t} &= \vec{\alpha}_w \vec{v}_{w,t-1} + \vec{\varepsilon}_w && (19) \end{aligned}$$

where $\vec{a}_t$ is the horizontal acceleration. State updates for v_z and $\vec{v}_w$ are expressed using two autoregressive (AR) models with lag $p = 1$. Parameters α_{vz}, ε_{vz}, ε_w, and $\vec{\alpha}_w$ are estimated using the least-squares method. The detailed process and results are shown in Section III-D.

To compute the acceleration ($\vec{a}_t$) at each time step, we need to consider the forces acting on the aircraft. The equations used at each time step are listed as follows:

$$\begin{aligned} \vec{a}_t &= (a_t, \psi_t) && (20)\\ a_t &= \frac{1}{m}(\eta_t T_t - D_t) - g\frac{v_{z,t}}{v_t} && (21)\\ \psi_t &= \arctan_2(v_{ax,t},\ v_{ay,t}) && (22)\\ T_t &= c_1\left(1 - \frac{1}{c_2}z_t + c_3 z_t^2\right) && (23)\\ D_t &= \frac{1}{2}\rho v_t^2 S\left[C_{d0} + k\left(\frac{L_t}{\frac{1}{2}\rho v_t^2 S}\right)^2\right] && (24) \end{aligned}$$

$$\begin{aligned} L_t &= m_t g && (25)\\ v_t &= \sqrt{v_{a,t}^2 + (v_{z,t})^2} && (26) \end{aligned}$$

where L, and D are lift and drag force, and ρ and S are the air density and the aircraft wing surface. From the BADA model [1], T is the maximum thrust profile defined by three coefficients. C_{d0} and k are coefficients for zero drag and induced drag respectively.

During real flights, not all thrust settings can be used for any aircraft mass. The reduction of thrust has a lower limit depending on actual aircraft mass. In the particle filter, such constrains are introduced as the initialization of the particles. The following Equation 27 is adapted from BADA [1] and is used to constrain the relation of mass and thrust setting.

$$\begin{aligned} \eta &\in [\eta_{min}, 1] \\ \eta_{min} &= 1 - 0.20 \cdot \frac{m_{max} - m}{m_{max} - m_{min}} \end{aligned} \tag{27}$$

C. Modeling the aircraft measurement noise

Measurement noise is closely related to sensor errors. For example, GPS errors affect position measurements (related to state $\tilde{x}$) and altimeter errors affect altitude measurements ($\tilde{z}$). Besides sensor errors, the truncation of values in downlinked data (such as latitude, longitude, and altitude in ADS-B position messages) also contributes to the measurement error. ADS-B transponders operate under regulations that define the minimum accuracy of sensors [10]. Different categories of uncertainty indicators are transmitted through ADS-B. In this paper, the *Navigation Accuracy Categories* (NAC) are considered for the construction of observation noise models.

In Table I, the *Navigation Accuracy Category - velocity* (NACv) defines the level of accuracy in terms of horizontal and vertical speed. The NACv indicator is broadcast within the airborne velocity message (Type Code 18). HFOM and VFOM, short for horizontal and vertical figure of merit, indicate the 95% confidence interval which is translated as twice of standard deviation in the observation noise model. They define the σ_{vg}^2, σ_{va}^2, and σ_{vz}^2.

TABLE I: Navigation Accuracy Category / Velocity

NACv	HFOM	VFOM
4	<0.3 m/s	<0.46 m/s
3	<1 m/s	<1.52 m/s
2	<3 m/s	<4.57 m/s
1	<10 m/s	<15.24 m/s
0	>10 m/s or unkown	>15.24 m/s or unkown

Similarly, in Table II, the *Navigation Accuracy Category - position* (NACp) defines the level of accuracy in terms of horizontal and vertical position. For each NACp level, an Estimated Position Error (EPU) and a Vertical Estimated Position Error (VEPU) are defined. Similarly, they indicate the 95% confidence interval of the horizontal and vertical bounds. From these values, σ_x^2, σ_y^2 and σ_z^2, of the observation noise can be obtained.

TABLE II: Navigation Integrity Category

NACp	EPU	VEPU
11	<3 m	<4 m
10	<10 m	<15 m
9	<30 m	<45 m
8	<0.05 NM	n/a
7 - 0	not used in this paper	

Four sets of observation noise models are proposed based on ADS-B specifications, with each corresponding to a difference combination of NACp and NACv, see Table III. These four

TABLE III: Noise models

Noise model	NACp	NACv
$\mathbf{\Sigma}_{n1}$	11	4
$\mathbf{\Sigma}_{n2}$	10	3
$\mathbf{\Sigma}_{n3}$	9	2
$\mathbf{\Sigma}_{n4}$	8	1

models are also the foundation for the experiments in Section V.

The exact values for all four noise models are shown as follows:

$$\begin{aligned}
\mathbf{\Sigma}_{n1} &= \mathrm{diag}(1.5^2, 1.5^2, 2^2, 0.15^2, 0.15^2, 0.23^2, 0.25^2, 0.25^2) \\
\mathbf{\Sigma}_{n2} &= \mathrm{diag}(5^2, 5^2, 7.5^2, 0.5^2, 0.5^2, 0.76^2, 0.75^2, 0.75^2) \\
\mathbf{\Sigma}_{n3} &= \mathrm{diag}(15^2, 15^2, 22.5^2, 1.5^2, 1.5^2, 2.28^2, 2.25^2, 2.25^2) \\
\mathbf{\Sigma}_{n4} &= \mathrm{diag}(48^2, 48^2, 68^2, 5^2, 5^2, 7.62^2, 7.5^2, 7.5^2)
\end{aligned} \tag{28}$$

D. First order autoregressive model for $\vec{v}_w$ and v_z

As previously described in Equations 18 and 19, wind and vertical rate along the climbing flight are modeled as two autoregressive (AR) models, since the underlying process model is unknown and these two time series do exhibit strong correlation between consecutive data points. For simplification, we are using a first-order AR model (AR1). In general the AR1 model (without bias) can be expressed in the following form:

$$\begin{aligned}
\chi_t &= \alpha \chi_{t-1} + \varepsilon \\
\varepsilon &\sim \mathcal{N}(0, \sigma^2)
\end{aligned} \tag{29}$$

where α is the model parameter and ε is a white noise with variance of σ^2. Using real flight data, these parameters can be estimated. For a given flight, α and σ can be estimated in Equation 30 using least-squares.

$$\begin{aligned}
\alpha &= \frac{\sum_{t=1}^{n} \chi_{t-1}\chi_t}{\sum_{t=1}^{n} \chi_{t-1}^2} \\
\sigma^2 &= \mathrm{Var}(\chi_t - \alpha\chi_{t-1})
\end{aligned} \tag{30}$$

The reason that wind can be modeled in this way is that locally it tends to be homogeneously distributed with some degree of variance. There is often a gradual increase of wind magnitude with increasing altitude. The model shall expect the α to remain close to 1.

On the other hand, using AR for the vertical rate offers a different perspective. The vertical rate is often a controlled variable in point mass flight models. Without more information from the aircraft, only an AR model can capture the change of vertical rate from the observer's point of view.

Using the autoregressive models, these state updates can be expressed using system difference equations that are similar to other states in Equation 13. To determine representative values for α and σ^2, ADS-B and Mode-S data was collected for climbing flights in a period of one month ($\pm 10,000$ flights). α and σ^2 were computed for each flight, and the mode (the most frequently occurring value) of each parameter is used for the AR models. Table IV summarizes these values for v_z and $\vec{v}_w$ (used in Equation 18, 19, and 29).

TABLE IV: Parameter summary for AR1 models

	α	σ
v_z	0.9989	0.3687
v_{wx}	1.0005	0.2004
v_{wy}	1.0009	0.2084

E. The observation equations

The observables are distance and speed in horizontal and vertical direction, denoted as $\mathbf{y}$. Recall the measurement function from Equation 2 with additive Gaussian observation noise:

$$\mathbf{y}_t = \boldsymbol{h}(\mathbf{x}_t) + \boldsymbol{n}_t; \quad \boldsymbol{n}_t \sim \mathcal{N}(0, \mathbf{\Sigma}_{\boldsymbol{n}}) \tag{31}$$

The observation function can be written in detail as follows:

$$\tilde{\vec{s}}_t = \vec{s}_t + \vec{n}_{s,t} \tag{32}$$
$$\tilde{z}_t = z_t + n_{z,t} \tag{33}$$
$$\tilde{\vec{v}}_{g,t} = \vec{v}_{a,t} + \vec{v}_{w,t} + \vec{n}_{vg,t} \tag{34}$$
$$\tilde{v}_{z,t} = v_{z,t} + n_{vz,t} \tag{35}$$
$$\tilde{\vec{v}}_{w,t} = \vec{v}_{w,t} + \vec{n}_{vw,t} \tag{36}$$

where the state and measurement vectors at time t are:

$$\begin{aligned}
\mathbf{x}_t &= [m_t, \eta_t, \vec{s}_t, z_t, \vec{v}_{a,t}, v_{z,t}, \vec{v}_{w,t}] \\
&= [m_t, \eta_t, (x_t, y_t), z_t, (v_{ax,t}, v_{ay,t}), v_{z,t}, (v_{wx,t}, v_{wy,t})] \\
\mathbf{y}_t &= [\tilde{\vec{s}}_t, \tilde{z}_t, \tilde{\vec{v}}_{g,t}, \tilde{v}_{z,t}, \tilde{\vec{v}}_{w,t}] \\
&= [(\tilde{x}_t, \tilde{y}_t), \tilde{z}_t, (\tilde{v}_{gx,t}, \tilde{v}_{gy,t}), \tilde{v}_{z,t}, (\tilde{v}_{wx,t}, \tilde{v}_{wy,t})]
\end{aligned} \tag{37}$$

and diagonal covariance matrix $\mathbf{\Sigma}_n$ for noise models:

$$\mathbf{\Sigma}_n = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{vg}^2, \sigma_{vg}^2, \sigma_{vz}^2, \sigma_{vw}^2, \sigma_{vw}^2) \tag{38}$$

Since the noise follows a multivariate normal distribution, the associated importance weight $p(\mathbf{y}_t|\mathbf{x}_t^i)$ can be conveniently computed as follows:

$$\begin{aligned}
\mathrm{w}^i &= p(\mathbf{y}_t|\mathbf{x}_t^i) \\
&\propto exp\left(-\frac{1}{2}\left[\mathbf{y}_t - \boldsymbol{h}(\mathbf{x}_t^i)\right]^T \mathbf{\Sigma}^{-1} \left[\mathbf{y}_t - \boldsymbol{h}(\mathbf{x}_t^i)\right]\right)
\end{aligned} \tag{39}$$

Thus we have the closed form of the particle filter approximation of the conditional probability function of $p(\mathbf{y}_t|\mathbf{x}_t^i)$, which can easily be calculated.

IV. Dedicated particle filter

A. Stochastic kernel

Compared to conventional SIR, in this paper, a modified the SIR particle filter is used. Stochastic kernel noise is applied to the states of interests (m and eta), as well as a hidden state, aircraft heading (ψ). A stochastic kernel is essentially a very small random Gaussian noise added to the the desired state variables of all particles after re-sampling. The stochastic kernels bring tiny variations in these state variables.

For m and η, the kernel helps to prevent the degeneration of particles and maintain a local diversity of values. In the case of aircraft heading ψ, it is essential for the functioning of the particle filter. This is because the current point-mass aircraft performance model lacks a process equation for heading changes. The small kernel allows the particle filter to keep tracking the small change of aircraft heading in some trajectories [1].

$$\begin{aligned}
k_m &\sim \mathcal{N}(0, \sigma_{k,m}^2) \\
k_\eta &\sim \mathcal{N}(0, \sigma_{k,\eta}^2) \\
k_\psi &\sim \mathcal{N}(0, \sigma_{k,\psi}^2)
\end{aligned} \tag{40}$$

[1] Hence experiments in this paper use only the forward parts of climbing flights.

In this paper, the choice of $\sigma_{k,m}$ and $\sigma_{k,\eta}$ are chosen to be 0.4% of the maximum m range ($0.004 \times (m_{mtow} - m_{oew})$) and maximum eta range (0.004×0.15). The choice of $\sigma_{k,\psi}$ is 1 degree.

B. The algorithm

Combining the previously defined system and observation parameters, the complete steps of the modified SIR particle filter is summarized in Algorithm 1 (adapted from the original generic SIR method in [11]).

Algorithm 1 The SIR particle filter: $p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \rightarrow p(\mathbf{x}_t|\mathbf{y}_t)$

let $t := 0$ ▷ initial time step
initialize set of particles $\{\mathbf{x}_0^i, \mathrm{w}_0^i; \quad i = 1, \cdots, N\}$
let $\tilde{\mathbf{x}}_0 := h^{-1}(\mathbf{y}_0)$
let $\mathbf{\Sigma}_n$ be observation noise covariance
for i=1 to N **do**
  draw $m_0^i \sim \mathcal{U}\{m_{oew}, m_{mtow}\}$
  compute $\eta_{min}^i := 1 - 0.2(m_{mtow} - m_0^i)/(m_{mtow} - m_{oew})$
  draw $\eta_0^i \sim \mathcal{U}\{\eta_{min}^i, 1\}$
  draw $(\vec{s}, z_0^i, \vec{v}_{a0}^i, v_{z0}^i, \vec{v}_{w0}^i) \sim \mathcal{N}\{\tilde{\mathbf{x}}_0, \mathbf{\Sigma}_n\}$
  $\mathbf{x}_0^i := (m_0^i, \eta_0^i, \vec{s}_0^i, z_0^i, \vec{v}_{a0}^i, v_{z0}^i, \vec{v}_{w0}^i)$
  $\mathrm{w}_0^i := 1/N$
for i=1 to N **do** ▷ SIR update
  $\mathrm{w}_t^i := \mathrm{w}_{t-1}^i p(\mathbf{y}_t|\mathbf{x}_t^i)$
  $\bar{\mathrm{w}}_t^i := \mathrm{w}_t^i / \sum_{i=1}^N \mathrm{w}_t^i$
for i=1 to N **do** ▷ re-sampling
  draw $\bar{\mathbf{x}}_t^i \sim \sum_{i=1}^N \bar{\mathrm{w}}_t^i \delta(\mathbf{x} - \mathbf{x}_t^i)$
  $\mathrm{w}_t^i := 1/N$
for i=1 to N **do** ▷ state evolution
  draw $\nu_t \sim \mathcal{N}(0, \mathbf{\Sigma}_\varepsilon)$
  $\mathbf{x}_{t+1}^i := f(\bar{\mathbf{x}}_t^i) + \nu_t$
for i=1 to N **do** ▷ apply kernel
  draw $dm^i \sim \mathcal{N}(0, \sigma_{k,m}^2)$ as kernel noise for m
  $m_t^i := m_t^i + dm^i$
  draw $d\eta^i \sim \mathcal{N}(0, \sigma_{k,\eta}^2)$ as kernel noise for η
  $\eta_t^i := \eta_t^i + d\eta^i$
  draw $d\psi^i \sim \mathcal{N}\{0, \sigma_{k,\psi}^2\}$ as kernel noise for heading
  compute $d\vec{v}_a^i$ from $d\psi^i$
  $\vec{v}_{a,t}^i := \vec{v}_{a,t}^i + d\vec{v}_a^i$
let $t := t + 1$, repeat 12

V. Experiments and Results

In this section, three different types of experiments are designed. First, a simulated flight is generated with a known mass and thrust setting. Four rounds of estimations are carried out under four different noise models. The simulation is to ensure the validity of the particle filter based on the proposed system equations. Second, a real flight is chosen and the estimation is undertaken with the same four noise models. This is to ensure the results from on real flight is in-line with the simulation. Finally, a large number simulation experiment is designed to study the uncertainty of the estimation. For all experiments, the number of particles are set to one million. This choice is a balance between accuracy and computational speed.

A. Simulation of a Boeing B737-700 flight

In this experiment, the aircraft mass m is set to be 60000 kg and the thrust setting η is set to be 0.96. The actual measurement noise for the simulated trajectory is $\mathbf{\Sigma}_{n2}$. A zero wind with a constant climb rate for the flight is assumed. Particle filters with four distinct noise models are applied to the trajectory for estimating the mass and thrust settings.

In Figure 2, the convergence of particles under assumed observation noise $\mathbf{\Sigma}_{n2}$ is illustrated as an example. In each convergence plot, the green line represents the true state value, the black dots are simulated observations, and the gray area is bounded by the minimum and maximum state values at each iteration. With the same assumed noise as the true noise, the mass and thrust settings nicely converge to their true values.

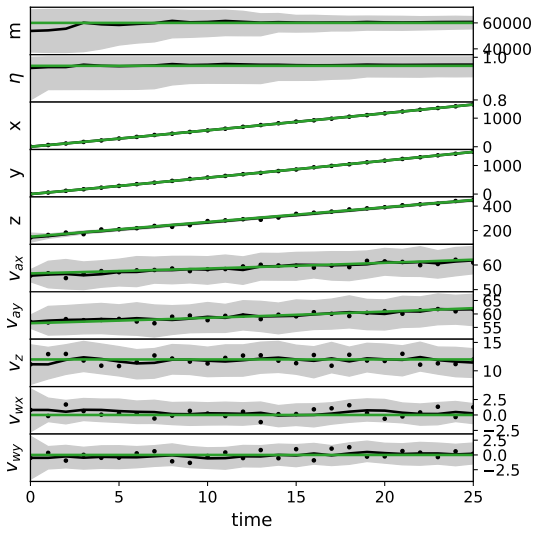

Fig. 2: Particle filter convergence (noise model $\mathbf{\Sigma}_{n2}$)

In Figure 3, all four final distributions of m and η along with their corresponding observation noise levels are shown. In each plot, the left hand side red distribution corresponds to mass, and the right hand side blue distribution represents the thrust setting. It is apparent that with increasing (assumed) observation noise, the uncertainty of the final results increases. The figure also shows that the estimates can become trapped in

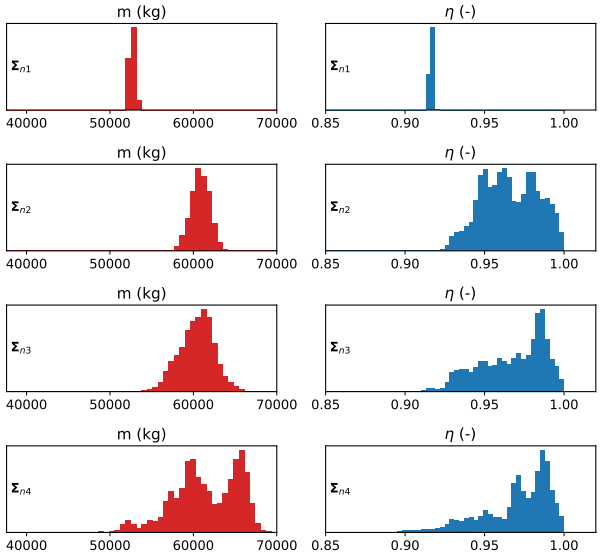

Fig. 3: Final distribution of m and η under different observation noise

a local optimum when the assumed noise is lower than actual noise (in the first plot). When the assumed noise level is much higher than the actual noise, the uncertainty in the final estimate become quite large (in the last plot). These observations are consistent with the fundamental concepts of the particle filter.

B. A real Boeing B737-700 flight example

In this second experiment, one real climbing flight of a Boeing 737-700 is used to demonstrate the SIR particle filter's application on real flights. The trajectory data are gathered from ADS-B and decoded using pyModeS [12]. The wind data are computed as the combination of ADS-B and Enhanced Mode-S data using a gas particle model as detailed in [13].

The trajectory and convergence of the particles when using noise model $\mathbf{\Sigma}_{n3}$ are shown in Figure 4. Similar to the previously simulated case, the black dots represent actual observed data, while the solid lines represent weighted average state values of particles at each iteration. In these plots, not only the mass and thrust setting exhibit convergence, but also the changes in airspeed, vertical rate, and wind are tracked closely by the particle filter with this noise level.

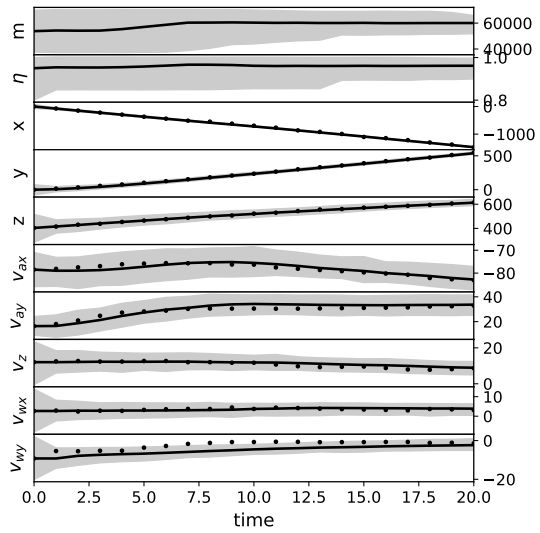

Fig. 4: Convergence of particle filter (noise model $\mathbf{\Sigma}_{n3}$)

Final distributions of estimated m and η under different noise models are shown in Figure 5. The same trend of increasing uncertainty is displayed when compared to the simulated cases. It is also noticeable that the variances in the result from real trajectory is smaller than simulated trajectory. This could be due to the fact that some states are not modeled. In turn, they pose an penalizing effect on the weights of particles.

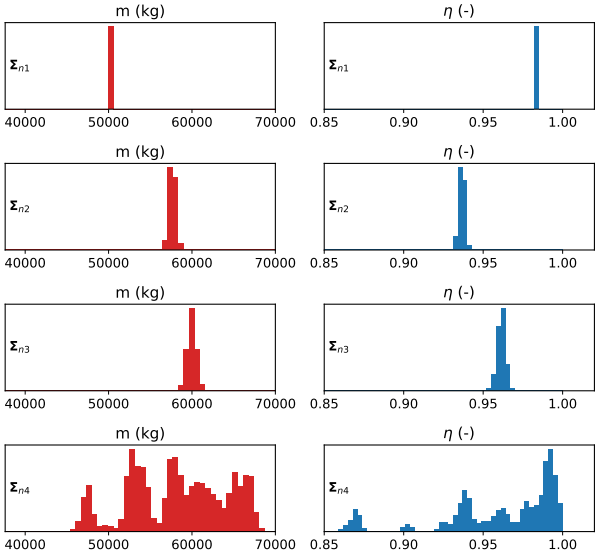

Fig. 5: Final distribution of m and η under different observation noise

Using the simulated and real flights, the applicability of the SIR particle filter for mass and thrust setting estimation is demonstrated. However, previous experiments are based on single sequence of filtering under each noise level. It is not sufficient to study the general uncertainty under these noise models. As such, a large number of filtering rounds will be conducted in the following part of the experiments for this purpose.

C. Accuracy and variation of the particle filter

To examine the accuracy and uncertainty of the particle filter, 500 rounds of estimation under each defined noise model are performed for the simulated flight. The simulated Boeing B737-700 flight is generated with a mass of 60000 kg, a thrust setting of 0.96, and a small simulation noise of $\mathbf{\Sigma}_{n1}/4$ [2].

In Figure 6, the final distribution results grouped by observation noise are shown, which consist of the average of m and η from all rounds. On the left hand side of the figure, results of m are indicated, with the Y-axis ranging from m_{oew} to m_{mtow}. The horizontal black line indicates the actual mass used in the simulations. On the right hand side, the thrust settings are plotted in the same fashion. We can see the particle filter yields an high level of accuracy with the simulated trajectory.

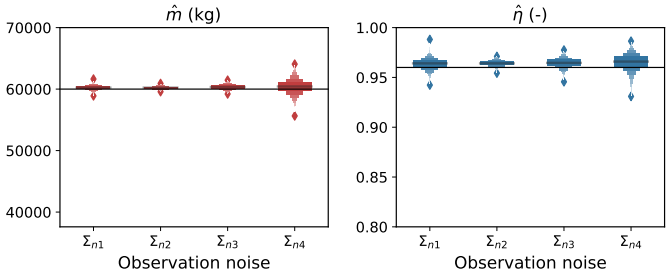

Fig. 6: Estimated mass and thrust setting

By computing the standard deviations of the mass and thrust values from all particles at the end of each run, we are able to understand how results vary under different noise models. In other words, it measures the degree of spreading of particle state values. On the left-hand side of the Figure 7, distributions of particle mass standard deviation are shown. With increasing observation noise, the uncertainty also increases. The right-hand side of the plot does indicate the same trends for the thrust setting but with less severity than the mass. There is an underlying reason for this phenomenon, which is discussed in detail in Section VII-A.

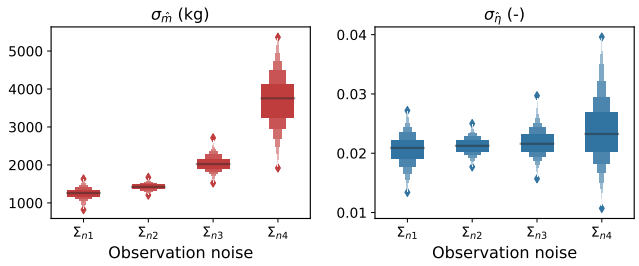

Fig. 7: Standard deviation of particles

To quantify the uncertainty under different noise models, 95% of the confidence range, twice the mean standard deviation, is used as an indicator. This is then translated in the percentage of the aircraft maximum takeoff mass and thrust settings, which are shown in Table V.

TABLE V: Uncertainties related to observation noise

Noise	$2\bar{\sigma}_{\hat{m}}$	% of m_{mtow}	$2\bar{\sigma}_{\hat{\eta}}$	%
Σ_{n1}	2512 kg	3.18%	0.0413	4.13%
Σ_{n2}	2840 kg	3.59%	0.0425	4.25%
Σ_{n3}	4077 kg	5.16%	0.0435	4.35%
Σ_{n4}	7427 kg	9.40%	0.0474	4.74%

[2]This small actual noise ensures all observation noise models are able to track the variations in trajectory.

These results are obtained based on one aircraft type. However, the percentage values from this table can still be an important indicator to quantify the uncertainties of estimations. It can even be used as indicator for the missing uncertainty component for other estimation methods, which were mentioned in the introduction of this paper.

VI. Validation

In this section, a set of 50 real flights with known mass are used to validate the proposed system model and the particle filter. These flights were carried out by a Cessna Citation II aircraft that is operated by TU Delft for student practicals. The mass is obtained accurately by weighing all passengers and measuring the exact amount of fuel on-board prior to the start of each flight.

Although accurate FMS trajectory data is available, ADS-B data was collected and used for the estimation. This is to completely validate our model and method in real use cases. Wind information is derived as the combination of ground speed from ADS-B and airspeed from Mode-S data. In addition, actual NACp amd NACv values are decoded from raw ADS-B data to automatically select the best observation noise model. All these data are also obtained using the open-source pyModeS [12] library.

To obtain a stable estimation, each trajectory is executed with the particle filter for 30 times [3]. The absolute errors ($|\Delta m|/m_{true}$) are computed and illustrated in Figure 8. As a result, the mean absolute error (MAE) is found to be 4.6%, while the median of the absolute error is 3.0%.

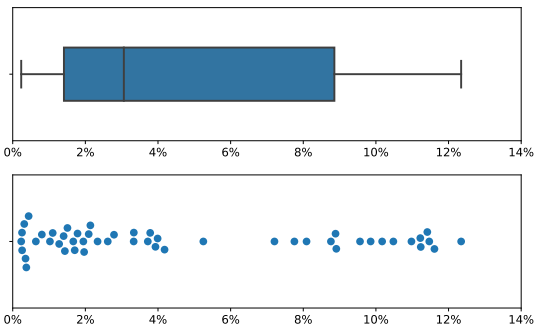

Fig. 8: Distributions of absolute errors

VII. Discussion

In this discussion section, we are going to elaborate on some of the fundamental aspects that concern the aircraft mass estimation. In addition, recommendations on choosing a noise model and practical particle filtering will be explained. Several limitations of the current performance model will also be weighed.

A. The influence of variation in m and eta

From previous experiments, it is noticeable that the thrust ratio (η) does not always converge. But the same trend does not apply to the convergence of mass until a larger noise is used. To understand the reason for this, variation of these two parameters and their influence on the true flight trajectory are studied. In line with this goal, two sets of simple simulations are conducted.

In the first set of simulations, the aircraft mass is fixed at $(m_{oew} + m_{mtow})/2$ and the thrust setting varies from 0.85 to 1. Results are shown on the left two plots of Figure 9. In the second set of simulations, the thrust setting is kept at 0.9 and aircraft mass varies from m_{oew} to m_{mtow}. Results are shown on the right-hand side of Figure 9. Only the horizontal flight distances and speed profiles are illustrated.

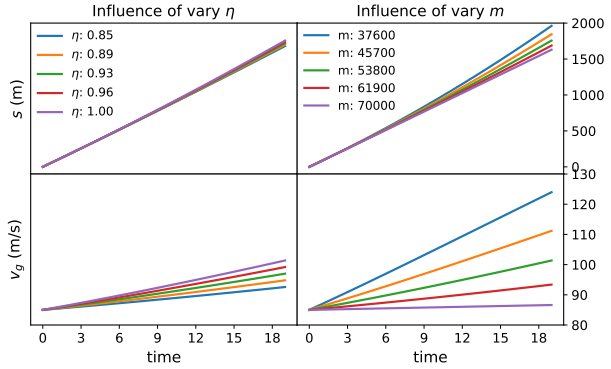

Fig. 9: Varying η versus varying m

The influence of different thrust settings on the flight trajectory is much smaller than the influence of differences in mass. This is shown in both distance traveled and the speed profile. We can intuitively conclude that only under small noise levels, the thrust settings can be estimated. While on the other hand, the estimation of the mass state may tolerate a higher level of noise. Next, noises are added to the simulation. For simplification, only the resulting speed profiles are shown in Figure 10.

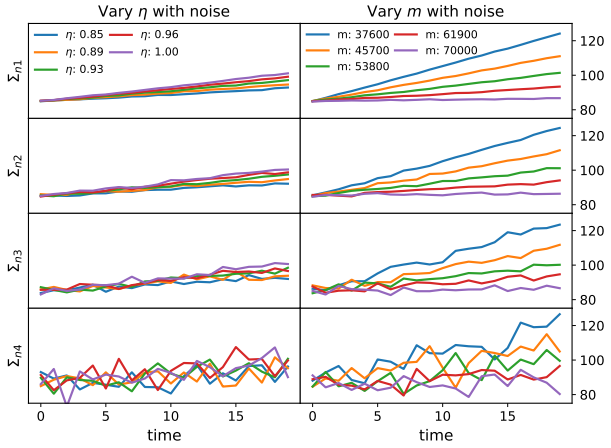

Fig. 10: Simulated speeds with different noise model

The difference in the influence of varying mass and thrust setting becomes quite apparent when noise are present. From the particle filter point of view, in Figure 11, the resulting distributions of m and their η are shown. Here, we've separated the mass in small bins and plotted the distribution of their corresponding thrust settings with the same colors.

From Figure 11, we can observe that the thrust setting η can only be separated from the corresponding mass when the observation noise is smaller than or equal to $\mathbf{\Sigma}_{n2}$. With a larger noise, the convergence of thrust settings becomes impossible. Furthermore, mass can only be estimated confidently when noise is smaller than $\mathbf{\Sigma}_{n4}$. These constrains are not specific to the particle filter method presented in this paper; they are also applicable to any other estimation method.

B. Computation complexity

The particle filter utilizes one million particles in the experiments. However, with ten states, the number of independent particles per state is fairly low, which counts around four

[3]This is similar to using 30 million particles in one round. The implication of this method will be discussed later on.

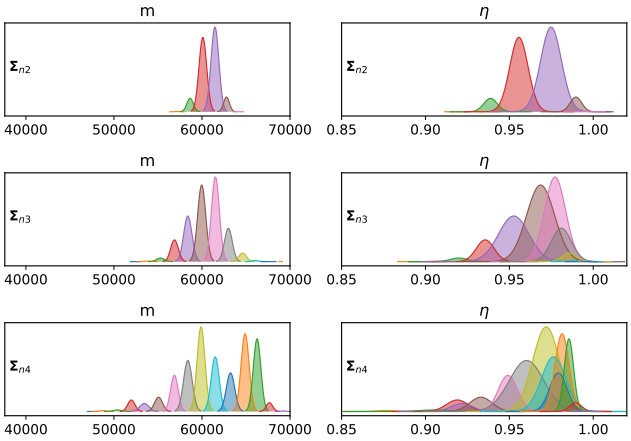

Fig. 11: Resulting distributions (thrust per mass)

particles per state ($1 \cdot 10^6 \approx 4^{10}$). Using such a low number of particles entails that we cannot sample the entire spectrum of all state dimensions effectively. Ideally, 1000^{10} ($1 \cdot 10^{30}$) would be a more appropriate number of particles. This is, however, not computationally possible with current computing power. Moreover it is often not necessary. In this paper, we used the SIR particle filter, which performs an additional re-sampling step at each iteration to keep only the particles with higher importance (weight). This step allows us to significantly reduce the number of particles required.

C. Limitations and recommendations

Although the point-mass model used in this paper is relatively comprehensive, there are limitations which can be improved in future studies.

1) In the current model, the bank angle is left out of the equations. This is an intermediate approach to simplify the system. The consequence is that we are only able to use forward flight segments for the estimations. Introducing this additional parameter could empower the model to cope with the estimation using data from turning trajectories.

2) Similarly, the aerodynamic properties are simplified using the drag polar provided by BADA, with the quasi-constant lift assumption ($L \approx mg$). A future model could also include the angle of attack to better compute the lift and drag.

3) Currently, a naive form of the observation noise is assumed. That is, noise for different observables are assumed to be uncorrelated. Using the previously mentioned VAR method could lead to a better defined covariance matrix in the future.

4) The noise models that are defined in this paper correspond to Version 1 and Version 2 of the ADS-B implementation. For ADS-B Version 0 transponders (equipped on older aircraft types), the noise should be defined based on its Navigation uncertainty Categories (NUC). However, these are less refined than NAC in Version 1 and 2.

5)As a rule of thumb, when it is uncertain, the noise models of $\mathbf{\Sigma}_{n2}$ or $\mathbf{\Sigma}_{n3}$ are generally good starting points. This conclusion is based on our observation that the majority of the ADS-B messages meet the accuracy levels of NACp 9/10 and NACv 3/2.

6) The ISA assumption is used in this paper to compute the maximum thrust. The effect of positive temperature deviation (commonly larger than 10 C°) from ISA generally reduces the maximum thrust of turbofan engines. This may, in turn, impact the estimations. However, for the validation flight used in this paper, the effect of deviation can be safely neglected. This is because the actual temperatures in the month of these flights (March of 2017) are generally similar or lower than ISA temperatures.

VIII. Conclusions

In this paper, steps are taken to construct a comprehensive model and estimation method to derive aircraft mass and thrust setting solely using ADS-B and Mode-S observations.

The complexity of such a problem was discussed at the start of this paper. In summary, the difficulty of such estimations lies in solving an inversed non-linear system that consists of ten states. Together with an unknown noise in the observation data, it was a challenge to make accurate estimations and provide uncertainty quantifications.

Unlike most previous optimization based methods, this paper proposed a recursive Bayesian approach. A specific form of particle filter was designed to tackle this particular estimation challenge. The recursive Bayesian approach not only provides estimates, but also allows for quantifying uncertainties under different noise levels. It was demonstrated in this paper that estimations are possible only under certain noise levels.

By linking the ADS-B native uncertainty reporting system with the observation noise, we were able to construct realistic observation noise models. At the same time, we were able to automatically select the appropriate noise model for each flight using ADS-B data.

With the convincing observation noise models and the aircraft system model, our particle filter was able to estimate aircraft mass and thrust setting when noise constrains were satisfied. Simulated, real, and validation flights were used to test the method proposed in this paper. They were used to approve the method, define the uncertainty, and validate the accuracy of the method. Despite the limitations addressed in the discussion section, our model and method yielded a mean absolute error of 4.6% (median absolute error of 3.0%) of the true mass on the validation dataset. Finally, the convergence speeds were swift. In most of the cases, estimations could be obtained within 30 seconds.

References

[1] A. Nuic, "User manual for the base of aircraft data (bada) revision 3.12," *Atmosphere*, vol. 2014, 2014.

[2] R. Alligier, D. Gianazza, and N. Durand, "Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 45–60, nov 2013.

[3] R. Alligier, D. Gianazza, and N. Durand, "Machine learning and mass estimation methods for ground-based aircraft climb prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3138–3149, 2015.

[4] C. Schultz, D. Thipphavong, and H. Erzberger, "Adaptive trajectory prediction algorithm for climbing flights," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, p. 2, 2012.

[5] J. Sun, J. Ellerbroek, and J. Hoekstra, "Modeling and inferring aircraft takeoff mass from runway ads-b data," in *7th International Conference on Research in Air Transportation*, 2016.

[6] J. Sun, J. Ellerbroek, and J. M. Hoekstra, "Aircraft initial mass estimation using bayesian inference method," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 59–73, 2018.

[7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[8] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[9] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 64–69, IEEE, 2005.

[10] RTCA, "Minimum operational performance standards for 1090mhz extended squitter automatic dependent surveillance-broadcast (ads-b) and traffic information services-broadcast (tis-b)[j]," *RTCA DO-260B with Corrigendum*, vol. 1, no. 1, pp. 1365–1372, 2011.

[11] H. A. P. Blom and E. A. Bloem, "Exact bayesian and particle filtering of stochastic hybrid systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, 2007.

[12] J. Sun, H. Vû, J. Ellerbroek, J. Hoekstra, K. von Hornbostel, T. Robitaille, R. Nobrega, and J. Watterson, "junzis/pymodes v1.2.1," Sept. 2017.

[13] J. Sun, H. Vû, J. Ellerbroek, and J. Hoekstra, "Ground-based wind field construction from mode-s and ads-b data with a novel gas particle model," in *Seventh SESAR Innovation Days*, SESAR, 2017.