2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Wind Estimation for Fixed-wing Aircraft using Command Tracking Approach

Haichao Hong, Mengmeng Wang, Florian Holzapfel, Shengjing Tang

Abstract— **This paper addresses a three-dimensional wind estimation problem for fixed-wing aircraft based on the tracking model predictive static programming which was originally proposed for trajectory tracking control. Without relying on any airflow measurement, the wind velocity is estimated by considering the wind vector as a virtual control input. The wind vector is corrected in real time so that the model output tracks the measured kinematic velocity states in a command-tracking manner. The proposed method features a closed-form solution and a fast indication of wind variations. The effectiveness of the proposed method is illustrated by numerical simulations in the presence of measurement noises.**

I. Introduction

Wind is undoubtedly one of the most important influencing factors for inflight safety, especially during take-off and landing [1]. The provision of accurate and timely wind information would ultimately contribute to the inflight safety concerning both manned and unmanned aerial vehicles. Besides, having the completed wind profile provided can benefit a class of applications, such as trajectory optimization-based dynamic soaring [2]. Therefore, research on wind estimation under such motivations becomes a globally concerted effort [3].

Existing research on this topic has been fruitful. Kalman Filter (KF) and its extensions have been widely employed in various studies. Tian recently proposed two model-aided extended Kalman Filters (EKFs) without using flow angles measurements [4]. The first filter makes use of the flow angle dynamics, while the second one estimates the wind velocity by utilizing random walk processes. Cho developed an EKF-based algorithm for estimation of two-dimensional wind structure [5]. Pachter presented an unscented Kalman Filter to estimate a constant wind with gusts [6]. Lie estimates air dates using two cascaded extended KFs [7]. The converging speed of some filter-based wind estimation approaches may be insufficiently fast in some emergency cases. Much of the existing studies rely on the airflow measurements, especially the airspeed sensor such as pitot tubes [4]–[6], [8], [9]. However, sensor failures cannot be completely avoided. There are cases, in which sensor malfunctions lead to accidents [10]. Therefore, an estimation method without using airflow sensors can be preferable for safety concern.

In the authors' previous work [3], through accident analysis, three key requirements of a suitable wind estimation approach with a good potential were summarized as: 1) an efficient algorithm with fast indication of wind variations; 2) estimation of an unknown three-dimensional (3D) wind structure; 3) minimum usage of airflow measurements including airspeed sensor and flow angles measurements. In [3], the authors also proposed a wind estimation approach based on the generalized model predictive static programming (G-MPSP) [11] which converts the wind estimation problem into a highly efficient static optimization problem without relying on any airflow measurement. The wind vector is considered as the input vector of the system model and estimated in real time by minimizing the difference between the predicted kinematic velocity and the measured value at current time. This is achieved by determining the proper correction to the wind vector using the G-MPSP based on the terminal output error. This approach fulfills the three key requirements. However, the estimate horizon should be judiciously chosen. As only the terminal error of the horizon is taken into consideration, a wider window results in large variations of correction in the horizon, which leads to inaccurate estimates.

In light of the previous research, a wind estimation approach based on an extension of MPSP is proposed in this paper. This extension addresses the command tracking problem, enforcing the state constraints throughout the whole interval [12]. The general idea of the proposed approach is similar to the approach in [3], but it estimates the wind in a command-tracking manner. The wind estimation is also achieved by considering the wind vector as a virtual input of the system model. During a past time horizon, the kinematic velocity states can be measured by using the global navigation satellite system (GNSS), which are regarded as the reference commands. The wind vector, i.e. the system input, is to be determined for tracking the command histories using the prediction model. If the predictions closely match the command histories, the wind velocity estimates can be obtained. The difference between the proposed approach in this paper and the approach in [3] is schematically shown in Fig. 1. If a relatively wider estimation horizon is selected, as the approach in [3] only minimize the terminal output error of the horizon, it does not guarantee that the predicted outputs are close to the measured output for the entire horizon, and thus the estimated values may be inaccurate. By contrast, the proposed approach minimizes the output error at every grid point in the horizon, so that the estimated values are close to the actual values throughout the horizon. This improvement makes it possible that a relatively wider horizon can be selected. This extension inevitably increases

H. Hong is with Beijing Institute of Technology and Technische Universität München. Email: haichao.hong@tum.de
M. Wang is with Beihang University.
F. Holzapfel is with Technische Universität München.
S. Tang is with Beijing Institute of Technology. (Corresponding author. Email: tangsj@bit.edu.cn)

978-1-5386-7891-6/18/$31.00 ©2018 IEEE

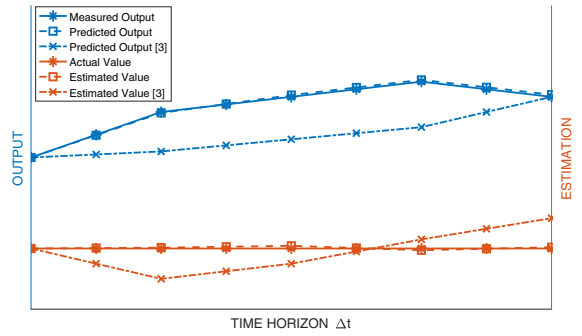

Fig. 1. Schematic comparison between the proposed approach and the approach in [3].

the computational workload as it computes output errors at every grid point instead of only the terminal point. However, as long as the horizon still remains reasonably short, little additional effort needs to be invested. It is noted that the approach in [3] can also generate satisfactory results when the horizon is short enough, and it is easier to be implemented. Therefore, the proposed method is an alternative approach to the method in [3] when a larger horizon is of interest. The proposed method satisfies the three key requirements as well, giving a closed-form solution of the unknown 3D wind velocity without using any airflow measurement.

The remaining parts of this paper are organized as follows. Section II introduces the model that considers the wind velocity as the virtual input to predict the kinematic velocity. Section III presents the generic theory of the parameter estimation method using T-MPSP for wind estimation. In Section IV, the numerical results are presented. The conclusions are drawn in Section V.

II. Aircraft Model

In this section, the point-mass model for states prediction is presented. The objective is to describe the dynamics of the kinematic velocity states in terms of the wind velocity. By propagating the system dynamics, the kinematic velocity can be predicted and thus the difference between the predicted and measured values can be found. The aircraft model is expressed based on the simulation framework proposed in [13]. The rotation of earth is neglected. The translational equations are given in the kinematic frame K as,

$$\begin{bmatrix} \dot{V}_K \\ \dot{\chi}_K \\ \dot{\gamma}_K \end{bmatrix}_K = \frac{1}{m} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{V_K \cos\gamma_K} & 0 \\ 0 & 0 & -\frac{1}{V_K} \end{bmatrix} (\boldsymbol{F}_T)_K , \quad (1)$$

where V_K, χ_K and γ_K denote the kinematic speed, flight-path course angle and flight-path climb angle. m is the mass of aircraft. $(\boldsymbol{F}_T)_K$ is the total force denoted in the K-frame given as

$$(\boldsymbol{F}_T)_K = \boldsymbol{M}_{KA} (\boldsymbol{F}_A)_A + \boldsymbol{M}_{KB} (\boldsymbol{F}_P)_B + \boldsymbol{M}_{KO} (\boldsymbol{F}_G)_O . \quad (2)$$

where $\boldsymbol{F}_A$, $\boldsymbol{F}_P$, $\boldsymbol{F}_G$ are the aerodynamic force specified in the aerodynamic frame A, the propulsion force specified in the body-fixed frame B, and the gravitational force specified in the O-frame, respectively. They are computed as

$$(\boldsymbol{F}_A)_A = \begin{bmatrix} -\bar{q}SC_D \\ \bar{q}SC_Q \\ -\bar{q}SC_L \end{bmatrix} \quad (3)$$

$$\bar{q} = \frac{1}{2}\rho (V_A)^2 \quad (4)$$

$$C_L = C_{L0} + C_{L_\alpha}\alpha \quad (5)$$

$$C_Q = C_{Q_\beta}\beta \quad (6)$$

$$C_D = C_{D0} + k_\alpha C_L + k_\beta C_Q \quad (7)$$

$$(\boldsymbol{F}_P)_B = \begin{bmatrix} \delta_T T_{\max} \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

$$(\boldsymbol{F}_G)_O = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} , \quad (9)$$

where $\bar{q}$, S, ρ and V_A are the dynamic pressure, the reference area, the air density and the aerodynamic speed, respectively. C represents aerodynamic coefficients and derivatives. α and β are the angle of attack and the angle of sideslip. m and g are the mass of aircraft and the acceleration of gravity. k_α and k_β denote the induced factors. δ_T and T_{max} are the thrust lever position and the maximum thrust, respectively. $\boldsymbol{M}_{KA}$, $\boldsymbol{M}_{KB}$ and $\boldsymbol{M}_{KO}$ are transformation matrices given by,

$$\boldsymbol{M}_{KA} = \boldsymbol{M}_{KO}\boldsymbol{M}_{OA} \quad (10)$$

$$\boldsymbol{M}_{KB} = \boldsymbol{M}_{KO}\boldsymbol{M}_{OB} \quad (11)$$

$$\boldsymbol{M}_{KO} = \boldsymbol{M}_y(\gamma_K)\boldsymbol{M}_z(\chi_K) \quad (12)$$

$$\boldsymbol{M}_{OA} = \boldsymbol{M}_z(-\chi_A)\boldsymbol{M}_y(-\gamma_A)\boldsymbol{M}_x(-\mu_A) \quad (13)$$

$$\boldsymbol{M}_{OB} = \boldsymbol{M}_z(-\Psi)\boldsymbol{M}_y(-\Theta)\boldsymbol{M}_x(-\Phi) , \quad (14)$$

where χ_A, γ_A and μ_A are the aerodynamic azimuth angle, climb angle and bank angle, respectively. Ψ, Θ and Φ are the Euler-angles. The definition of basic transformation matrices are given in Appendix.

As the proposed method applies to a past horizon, $(\boldsymbol{F}_P)_B$ is specified. Moreover, it is assumed $(\boldsymbol{F}_G)_O$ constant and known. The attitudes in the past are measured, so $\boldsymbol{M}_{OB}$ is known as well. $\boldsymbol{M}_{KO}$ is function of kinematic velocity states. $\boldsymbol{M}_{OA}$ and $(\boldsymbol{F}_A)_A$ are functions of wind velocity. Therefore, they need to be computed with respect to the aerodynamic velocity. The aerodynamic velocity components can be obtained using the wind triangle as

$$\begin{bmatrix} u_A \\ v_A \\ w_A \end{bmatrix}_O = \begin{bmatrix} V_K \cos\chi_K \cos\gamma_K \\ V_K \sin\chi_K \cos\gamma_K \\ -V_K \sin\gamma_A \end{bmatrix}_O - \begin{bmatrix} u_W \\ v_W \\ w_W \end{bmatrix}_O , \quad (15)$$

where u_W, v_W and w_W are the wind velocity components. They are provided by the algorithm. Then, the angles between O-Frame and A-Frame, and aerodynamic speed can

be calculated as,

$$\chi_A = \arctan\left(\frac{v_A}{u_A}\right) \tag{16}$$

$$\gamma_A = \arctan\left(\frac{-\left(w_A\right)}{\sqrt{\left(u_A\right)^2 + \left(v_A\right)^2}}\right) \tag{17}$$

$$V_A = \sqrt{\left(u_A\right)^2 + \left(v_A\right)^2 + \left(w_A\right)^2}\,. \tag{18}$$

Having the aerodynamic velocity calculated leads to the omission of the air flow measurements including airspeed and flow angles sensors. Hence, they are not required by the estimation approach. The flow angles α, β and μ_A can be thus obtained using coordinate system knowledge as [3],

$$\begin{aligned}\sin\beta = {} & \cos\gamma_A\big[\cos\Phi sin(\chi_A - \Psi) \\ & + sin\Theta sin\Phi\cos(\chi_A - \Psi)\big] \\ & - sin\gamma_A\cos\Theta sin\Phi \end{aligned} \tag{19}$$

$$\begin{aligned}\sin\alpha = {} & \Big\{\cos\gamma_A\big[sin\Theta\cos\Phi\cos(\chi_A - \Psi) \\ & - sin\Phi sin(\chi_A - \Psi)\big] \\ & - sin\gamma_A\cos\Theta\cos\Phi\Big\}/\cos\beta \end{aligned} \tag{20}$$

$$\begin{aligned}\sin\mu_A = {} & \cos\gamma_A(\cos\Theta\sin\Phi\cos\beta + \cos\alpha\sin\Theta\sin\beta \\ & - \cos\Phi\cos\Theta\sin\alpha\sin\beta) \\ & + \sin\gamma_A\cos(\chi_A - \Psi)(\sin\Theta\sin\Phi\cos\beta \\ & - \sin\Theta\cos\Phi\sin\alpha\sin\beta - \cos\Theta\cos\alpha\sin\beta) \\ & + \sin\gamma_A\sin(\chi_A - \Psi)(\cos\Phi\cos\beta \\ & + \sin\Phi\sin\alpha\sin\beta) \end{aligned} \tag{21}$$

Therefore, the dynamics of kinematic velocity expressed as absolute value and flight-path angles can be computed with the kinematic velocity, the wind velocity and the measured information. Taking advantage of this model, the kinematic velocity states can be predicted and can be compared with the measured histories, which will be utilized by the algorithm introduced in the next section.

III. Wind Estimation based on T-MPSP

The proposed wind estimation method is developed based on the T-MPSP [12]. Although the structure of the proposed algorithm finds itself close to the T-MPSP, the proposed algorithm applies to a past time interval, making it a dual application of the T-MPSP, similar to relationship between model predictive control and moving horizon estimation. The general idea of the proposed method is to find the suitable wind vector as input of the aircraft model which leads to a close match between the predicted kinematic velocity states and the measured values. In this section, a generic formulation of the parameter estimation method based on the T-MPSP is presented. The wind estimation can be achieved using this formulation.

A system dynamics are written in a discrete form as,

$$\boldsymbol{x}_{k+1} = F_k\left(\boldsymbol{x}_k,\ \boldsymbol{\nu}_k\right)\,, \tag{22}$$

where $\boldsymbol{x} \in \mathbb{R}^m$ is the state vector, and $\boldsymbol{\nu} \in \mathbb{R}^n$ is the parameter vector to be estimated. F is at least $\mathcal{C}^1$ with respect to $\boldsymbol{x}$ and $\boldsymbol{\nu}$. $k = 1, 2, \ldots, N-1$ are the time steps. This dynamics apply to a past time horizon $[t_c - \Delta t,\ t_c]$, where t_c is the current time. The dynamics of the kinematic velocity described in the previous section in Eq. (1) is considered, so $\boldsymbol{x} = [V_K, \chi_K, \gamma_K]^{\mathrm{T}}$ and $\boldsymbol{\nu} = [u_W, v_W, w_W]^{\mathrm{T}}$.

The output equation is expressed as,

$$\boldsymbol{y}_k = h\left(\boldsymbol{x}_k\right) \tag{23}$$

where $\boldsymbol{y} \in \mathbb{R}^p$ is the output vector. In the wind estimation problem, the output is considered as $\boldsymbol{y} = \boldsymbol{x} = [V_K, \chi_K, \gamma_K]^{\mathrm{T}}$. The state error at step k can be obtained from Eq. (22) as

$$d\boldsymbol{x}_k = \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right] d\boldsymbol{x}_{k-1} + \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{\nu}_{k-1}}\right] d\boldsymbol{\nu}_{k-1}\,, \tag{24}$$

where $k = 2, 3, \ldots, N$. The output error at step k in terms of the state error and parameter error at $(k-1)^{\mathrm{th}}$ step is thus written as,

$$d\boldsymbol{y}_k = \frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\left\{\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right] d\boldsymbol{x}_{k-1} + \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{\nu}_{k-1}}\right] d\boldsymbol{\nu}_{k-1}\right\}. \tag{25}$$

Expanding Eq. (25) for $j = k-1,\ k-2, \ldots, 1$ leads to

$$\begin{aligned} d\boldsymbol{y}_k &= \frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\left\{\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right] d\boldsymbol{x}_{k-1} + \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{\nu}_{k-1}}\right] d\boldsymbol{\nu}_{k-1}\right\} \\ &= \frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\left\{\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right]\left[\frac{\partial F_{k-2}}{\partial \boldsymbol{x}_{k-2}}\right] d\boldsymbol{x}_{k-2}\right. \\ &\quad \left. + \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right]\left[\frac{\partial F_{k-2}}{\partial \boldsymbol{\nu}_{k-2}}\right] d\boldsymbol{\nu}_{k-2} + \left[\frac{\partial F_{k-1}}{\partial \boldsymbol{\nu}_{k-1}}\right] d\boldsymbol{\nu}_{k-1}\right\} \\ &\quad \vdots \\ &= \boldsymbol{A}^k d\boldsymbol{x}_1 + \boldsymbol{B}_1^k d\boldsymbol{\nu}_1 + \boldsymbol{B}_2^k d\boldsymbol{\nu}_2 + \ldots + \boldsymbol{B}_{k-1}^k \boldsymbol{\nu}_{k-1} \\ &= \boldsymbol{A}^k d\boldsymbol{x}_1 + \sum_{j=1}^{k-1} \boldsymbol{B}_j^k d\boldsymbol{\nu}_j\,, \end{aligned} \tag{26}$$

where $k = 2, 3, \ldots, N$, and

$$\boldsymbol{A}^k = \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\right]\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right]\left[\frac{\partial F_{k-2}}{\partial \boldsymbol{x}_{k-2}}\right]\cdots\left[\frac{\partial F_1}{\partial \boldsymbol{x}_1}\right] \tag{27}$$

$$\boldsymbol{B}_j^k = \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\right]\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{x}_{k-1}}\right]\cdots\left[\frac{\partial F_{j+1}}{\partial \boldsymbol{x}_{j+1}}\right]\left[\frac{\partial F_j}{\partial \boldsymbol{\nu}_j}\right], \quad \text{for} \quad j = 1, 2, \ldots, k-2 \tag{28}$$

$$\boldsymbol{B}_{k-1}^k = \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\right]\left[\frac{\partial F_{k-1}}{\partial \boldsymbol{\nu}_{k-1}}\right] \tag{29}$$

It is assumed that the initial condition is specified, so $d\boldsymbol{x}_1 = 0$. Therefore, Eq. (26) is written as,

$$d\boldsymbol{y}_k = \sum_{j=1}^{k-1} \boldsymbol{B}_j^k d\boldsymbol{\nu}_j\,. \tag{30}$$

This equation computes the output error at k^{th} step in terms of the parameter errors at previous steps. The computation of $\boldsymbol{B}_j^k$, for $k = 2, 3, \ldots, N$, $j = 1, 2, \ldots, k-1$ can be intensive.

However, this process can be significantly reduced when $\boldsymbol{B}_j^k$ are computed recursively as follows,

$$\begin{aligned}
\left(\boldsymbol{B}_{k-1}^k\right)^0 &= \boldsymbol{I}_{n\times n} \\
\left(\boldsymbol{B}_j^k\right)^0 &= \left(\boldsymbol{B}_{j+1}^k\right)^0 \left[\frac{\partial F_{j+1}}{\partial \boldsymbol{x}_{j+1}}\right] , j = k-2, k-3, \ldots, 1 \\
\boldsymbol{B}_j^k &= \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{x}_k}\right] \left(\boldsymbol{B}_j^k\right)^0 \left[\frac{\partial F_k}{\partial \boldsymbol{\nu}_k}\right] , j = k-1, k-2, \ldots, 1 ,
\end{aligned} \tag{31}$$

where the superscript 0 denotes intermediate variables. This process needs to be called for every $k = 2, 3, \ldots, N$. Eq. (30) represents the relation between the output errors and the parameter errors. As mentioned before, the parameter errors need to be find out to minimize the output errors between the predicted and the measured outputs. Next, an optimization framework is formulated to exploit Eq. (30) and to obtain the suitable parameter corrections. It is expected to minimize the output error while minimizing the parameter corrections. Therefore, the cost function is selected as,

$$\begin{aligned}
J = &\frac{1}{2}\sum_{k=2}^{N} \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\right)^{\mathrm{T}} \boldsymbol{Q}_k \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \left(\boldsymbol{\nu}_k - \boldsymbol{\nu}_k^p\right)^{\mathrm{T}} \boldsymbol{R}_k \left(\boldsymbol{\nu}_k - \boldsymbol{\nu}_k^p\right) ,
\end{aligned} \tag{32}$$

where $\hat{\boldsymbol{y}}$ is the measured output vector and $\boldsymbol{\nu}^p$ is the uncorrected parameter vector in the previous iteration. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the weighting matrices. Define

$$\begin{aligned}
\Delta \boldsymbol{y}_k &\triangleq \boldsymbol{y}_k - \boldsymbol{y}_k^p \\
\Delta \hat{\boldsymbol{y}}_k &\triangleq \hat{\boldsymbol{y}}_k - \boldsymbol{y}_k^p \\
\Delta \boldsymbol{\nu}_k &\triangleq \boldsymbol{\nu}_k - \boldsymbol{\nu}_k^p
\end{aligned} \tag{33}$$

where $\Delta \boldsymbol{y}$ denotes the change of the predicted output over iterations. $\Delta \hat{\boldsymbol{y}}$ denotes the difference between the measured output and the previous predicted output. $\Delta \boldsymbol{\nu}$ denotes the change of parameter vector over iterations. Substituting Eq. (33) into Eq. (32) leads to

$$\begin{aligned}
J = &\frac{1}{2}\sum_{k=2}^{N} \left(\Delta \boldsymbol{y}_k - \Delta \hat{\boldsymbol{y}}_k\right)^{\mathrm{T}} \boldsymbol{Q}_k \left(\Delta \boldsymbol{y}_k - \Delta \hat{\boldsymbol{y}}_k\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \left(\Delta \boldsymbol{\nu}_k\right)^{\mathrm{T}} \boldsymbol{R}_k \left(\Delta \boldsymbol{\nu}_k\right) .
\end{aligned} \tag{34}$$

The first term of the cost function ensures that the incremental changes of the predicted outputs compensate the output errors in every grid points, while the second term penalizes the changes of the parameter estimates. Considering small error approximations, i.e. $\Delta \boldsymbol{y} \approx d\boldsymbol{y}$ and $\Delta \boldsymbol{\nu} \approx d\boldsymbol{\nu}$, Eq. (34) can be written as,

$$\begin{aligned}
J = &\frac{1}{2}\sum_{k=2}^{N} \left(d\boldsymbol{y}_k - \Delta \hat{\boldsymbol{y}}_k\right)^{\mathrm{T}} \boldsymbol{Q}_k \left(d\boldsymbol{y}_k - \Delta \hat{\boldsymbol{y}}_k\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \left(d\boldsymbol{\nu}_k\right)^{\mathrm{T}} \boldsymbol{R}_k \left(d\boldsymbol{\nu}_k\right) .
\end{aligned} \tag{35}$$

Substituting the relation given in Eq. (30) into the cost function above leads to

$$\begin{aligned}
J = &\frac{1}{2}\sum_{k=2}^{N} \left(\sum_{j=1}^{k-1} \boldsymbol{B}_j^k d\boldsymbol{\nu}_j - \Delta \hat{\boldsymbol{y}}_k\right)^{\mathrm{T}} \boldsymbol{Q}_k \left(\sum_{j=1}^{k-1} \boldsymbol{B}_j^k d\boldsymbol{\nu}_j - \Delta \hat{\boldsymbol{y}}_k\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \left(d\boldsymbol{\nu}_k\right)^{\mathrm{T}} \boldsymbol{R}_k \left(d\boldsymbol{\nu}_k\right) .
\end{aligned} \tag{36}$$

Now that the cost function has been given, it needs to be optimized. Following the optimization theory [14], the necessary conditions for optimality are given with respect to the parameter errors $d\boldsymbol{\nu}$, $\forall\, k = 1, 2, \ldots, N-1$ as,

$$\frac{\partial J}{\partial d\boldsymbol{\nu}_k} = 0, \ \forall\, k = 1, 2, \ldots, N-1 , \tag{37}$$

The necessary conditions lead to a linear system as Eq. (38), where $\boldsymbol{C}_{ij}$ and $\boldsymbol{b}_i$ $\forall\ i = 1, 2, \ldots, N-1$, $j = 1, 2, \ldots, N-1$ are given by

$$\boldsymbol{C}_{ij} = \sum_{l=(j+1)}^{N} \left(\boldsymbol{B}_i^l\right)^{\mathrm{T}} \boldsymbol{Q}_l \boldsymbol{B}_j^l \tag{39}$$

$$\boldsymbol{b}_i = \sum_{l=2}^{N} \left(\boldsymbol{B}_i^l\right)^{\mathrm{T}} \boldsymbol{Q}_l \Delta \boldsymbol{y}_l^d \tag{40}$$

The diagonal elements of the coefficient matrix on the left-hand side are $\boldsymbol{C}_{ij} + \boldsymbol{R}_i$ and the other elements are $\boldsymbol{C}_{ij}$. The solution of the system in Eq. (38) can be obtained as in Eq. (41). Therefore, the updated parameter vector can be expressed as

$$\boldsymbol{\nu}_k = \boldsymbol{\nu}_k^p + d\boldsymbol{\nu}_k , \forall\ k = 1, 2, \ldots, N-1 \tag{42}$$

This procedure needs to be repeated in an iterative manner until the predicted kinematic velocity histories match the

$$\begin{pmatrix} \boldsymbol{C}_{11} + \boldsymbol{R}_1 & \cdots & \boldsymbol{C}_{1(N-1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C}_{(N-1)1} & \cdots & \boldsymbol{C}_{(N-1)(N-1)} + \boldsymbol{R}_{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} d\boldsymbol{\nu}_1 \\ \vdots \\ d\boldsymbol{\nu}_{N-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_{N-1} \end{pmatrix} \tag{38}$$

$$\begin{pmatrix} d\boldsymbol{\nu}_1 \\ \vdots \\ d\boldsymbol{\nu}_{N-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11} + \boldsymbol{R}_1 & \cdots & \boldsymbol{C}_{1(N-1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C}_{(N-1)1} & \cdots & \boldsymbol{C}_{(N-1)(N-1)} + \boldsymbol{R}_{(N-1)(N-1)} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_{N-1} \end{pmatrix} \tag{41}$$

TABLE I

MEASUREMENT NOISES IN SIMULATION

Variable	Noise
Euler-angles, Ψ, Θ, Φ, deg	$\sigma = 0.5\ deg$
Kinematic speed, V_K, m/s	$\sigma = 0.05\ m/s$
Flight-path angles, χ_K, γ_K, deg	$\sigma = 0.05\ deg$

measured histories, such that the wind velocity is estimated. Using the measured state histories, the proposed method can be implemented in a periodical manner to update the wind information in real time.

IV. SIMULATION RESULTS

In this section, the numerical simulation results were presented. The simulation starts with a trainer aircraft flying horizontally at a speed of 100 m/s. The fixed time horizon for estimation is 1 s. The sample time of the measured kinematic velocity states is 0.2 s. The algorithm updates the wind information every 0.2 s. The measurement noises are embodied in Table I. It is to be mentioned that the constant bias in measurement does not affect the estimation, as the estimated is computed based on state deviations, and it is assumed that the measurement bias is not time-varying during the estimation horizon. The wind profile is designed as multiple wind shears along three axises of the NED frame, varying from -10 m/s to 10 m/s. The approach presented in [3] does not perform well using the given horizon and sample time without considering some workarounds, such as assuming that the wind correction is constant during the horizon. Therefore, the comparison is not shown.

A Monte Carlo simulation is carried out, which includes 200 simulation runs. The wind profile is randomly generated every time. An example run is shown in Fig. 2 schematically illustrating the simulation scenario. The blue lines represent the wind estimation histories, while the black dashed lines depict the true wind velocity. The smoothed results obtained by using the moving average method are shown by the red lines for comparison. The simulation results show that the algorithm produced reasonably accurate estimates. Though the smoothing significantly reduced noise in estimates and can be useful in practice, the smoothed results are not discussed in the following, as the raw data of the algorithm outputs is to be shown. One may notice that the estimates in the horizontal plane is significantly higher than in the z_O axis. This is due in part to the couplings between the two horizontal axises, and to the response in the vertical direction that is more sensitive than in the horizontal plane. The error distribution of the Monte Carlo simulation is depicted in Fig. 3. Most of the estimates in the horizontal plane contain errors of less than 1 m/s, while the errors in the vertical direction are primarily within 0.2 m/s. Further details in terms of the root mean square error (RMSEs) and the mean absolute errors can be seen in Fig. 4. The average RMSEs are 1.20 m/s, 1.10 m/s and 0.18 m/s in the x_O, y_O and z_O axis, respectively. Moreover, the average mean absolute errors for all three dimensions are 0.82 m/s, 0.78 m/s and

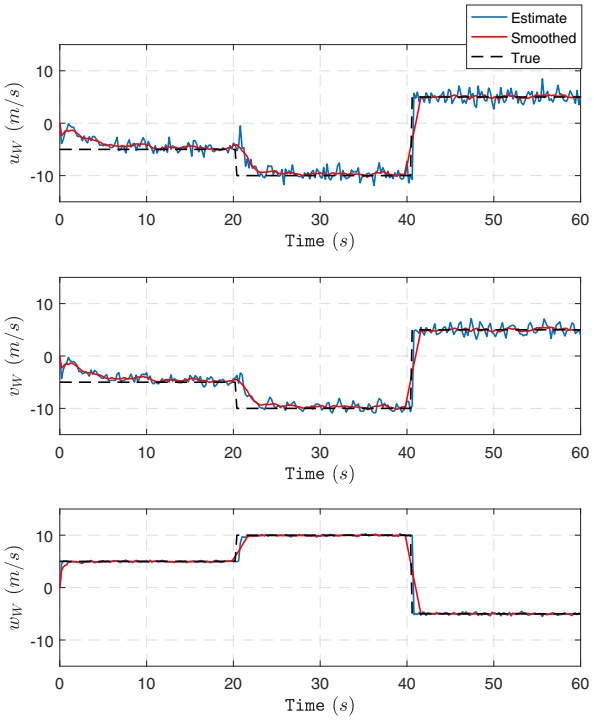

Fig. 2. Example run

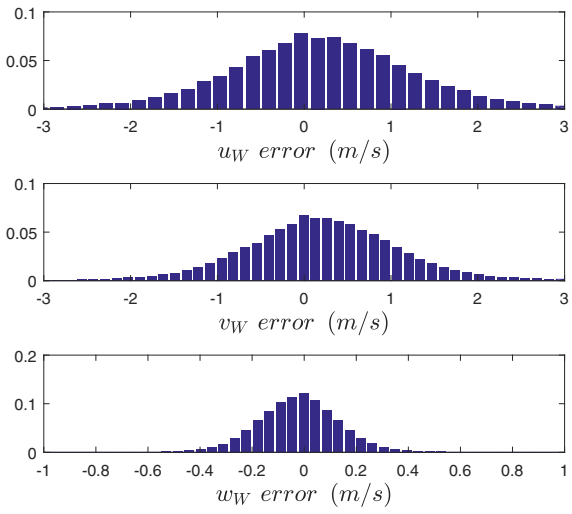

Fig. 3. Error distributions

0.13 m/s accordingly. The measurement noises result in the noises in estimates because of the nature of the proposed method that the deviation of prediction will immediately be reflected in the wind estimation. However, the significance of implementing the proposed method is to indicate the wind variation whenever the measured values do not follow the prediction, which is substantially necessary for emergency cases, as the change of the atmospheric environment is

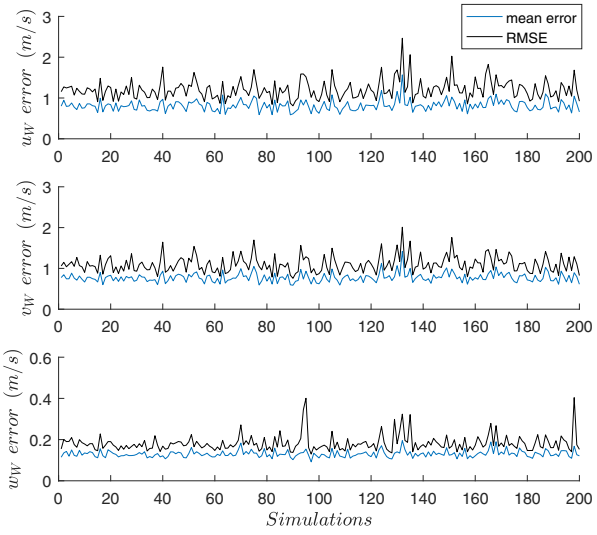

Fig. 4. Details of simulations

of great importance for pilots and systems. The statistics suggest that the algorithm produced satisfactory estimates in terms of accuracy. Besides, since the wind variation in the vertical direction is usually considered as the most hazardous during take-off and landing, this results shows promising potential in practice. In summary, the proposed method is able to produce reasonable wind estimates in the presence measurement noises.

V. Conclusion

This paper addressed the wind estimation problem efficiently using the tracking model predictive static programming (T-MPSP). The wind velocity is considered as the input of the system model, computed to track the measured kinematic velocity states. Airflow measurements are not required by the algorithm and the solution can be obtained in a closed form. The simulation results showed that the wind profile was estimated satisfactorily in the presence of measurement noises. The estimated wind information may contribute to inflight safety and be utilized by other applications accounting for providing necessary environmental details. The proposed command tracking-based parameter estimation approach has the potential to be applied to different problems.

Appendix

The transformation matrices are defined as,

$$\boldsymbol{M}_x(\xi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\xi & \sin\xi \\ 0 & -\sin\xi & \cos\xi \end{pmatrix}$$

$$\boldsymbol{M}_y(\xi) = \begin{pmatrix} \cos\xi & 0 & -\sin\xi \\ 0 & 1 & 0 \\ \sin\xi & 0 & \cos\xi \end{pmatrix}$$

$$\boldsymbol{M}_z(\xi) = \begin{pmatrix} \cos\xi & \sin\xi & 0 \\ -\sin\xi & \cos\xi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where ξ is a given angle.

References

[1] S. S. Mulgund and R. F. Stengel, "Optimal nonlinear estimation for aircraft flight control in wind shear," *Automatica*, vol. 32, no. 1, pp. 3–13, 1996.

[2] J. W. Langelaan, J. Spletzer, C. Montella, and J. Grenestedt, "Wind field estimation for autonomous dynamic soaring," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 16–22.

[3] H. Hong, M. Wang, F. Holzapfel, and S. Tang, "Fast real-time three-dimensional wind estimation for fixed-wing aircraft," *Aerospace Science and Technology*, vol. 69, pp. 674–685, 2017.

[4] P. Tian and H. Chao, "Model aided estimation of angle of attack, sideslip angle, and 3d wind without flow angle measurements," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1844.

[5] A. Cho, J. Kim, S. Lee, and C. Kee, "Wind estimation and airspeed calibration using a uav with a single-antenna gps receiver and pitot tube," *IEEE transactions on aerospace and electronic systems*, vol. 47, no. 1, pp. 109–117, 2011.

[6] M. Pachter, N. Ceccarelli, and P. R. Chandler, "Estimating mavs heading and the wind speed and direction using gps, inertial, and air speed measurements," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[7] F. A. P. Lie and D. Gebre-Egziabher, "Synthetic air data system," *Journal of Aircraft*, vol. 50, no. 4, pp. 1234–1249, 2013.

[8] J. Petrich and K. Subbarao, "On-Board Wind Speed Estimation for UAVs," in *AIAA Guidance, Navigation, and Control Conference*, 2011.

[9] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1016–1030, 2011.

[10] M. Wang, F. Holzapfel, S. Zhang, and F. Zhang, "Backup controller for large transport aircraft with insufficient natural stability," *Journal of Guidance, Control, and Dynamics*, pp. 1–14, , 2016.

[11] A. Maity, H. B. Oza, and R. Padhi, "Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1897–1913, 2014.

[12] P. Kumar and R. Padhi, "Extension of model predictive static programming for reference command tracking," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 855–861, 2014.

[13] F. Fisch, "Development of a framework for the solution of high-fidelity trajectory optimization problems and bilevel optimal control problems," Ph.D. dissertation, Technische Universität München, 2011.

[14] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. Hemisphere, New York, 1975.