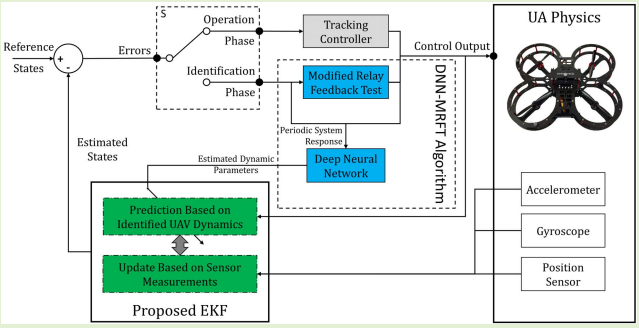
Real-Time Adaptive Dynamics Based State Estimation Scheme for Unmanned Aircrafts

Mohamad Wahbah, Mohamad Chehadeh, *Member, IEEE*, Mahmoud Hamandi, Lakmal Seneviratne, and Yahya Zweiri, *Member, IEEE*

***Abstract*—In this paper we present a state estimation scheme for Unmanned Aircrafts (UAs) utilizing dynamics based models and multi-sensor data fusion. Employing the UA dynamics in estimation can substantially enhance the estimator performance, but obtaining accurate dynamics parameters for each UA is computationally costly and complex. To eliminate these issues, we propose two decoupled Extended Kalman Filters (EKFs), namely the Rotational Decoupled Extended Kalman Filter (RDEKF) and the Translational Decoupled Extended Kalman Filter (TDEKF). The dynamics parameters in these filters are identified in real-time using the Deep Neural Network and the Modified Relay Feedback Test (DNN-MRFT) approach. This approach doesn't demand prior knowledge of the UA physical parameters, requiring only an Inertial Measurement Unit (IMU) and a positioning system for model classification. Our estimation scheme provides position, velocity and attitude estimates, in addition to smooth lag-free inertial acceleration estimates. We show experimentally the advantages of our approach on trajectory tracking problems that uses low rate position sensors. We also demonstrate how utilizing the estimated acceleration in feedback control can reduce the tracking error of an optimally tuned system by 43%. Moreover, the proposed estimator produces smooth estimates that leads to a reduction of controller action by 6.6%, when compared to kinematic based estimators. We compare the achieved results against other methods that require full prior knowledge of the UA parameters or the noise models, and show advantages in performance and real-time capability.**



***Index Terms*—Acceleration feedback, state estimation, DNN-MRFT, dynamic model, Kalman filter, unmanned aircraft (UA).**

NOMENCLATURE

$\mathcal{W}$	Inertial Frame formed from the basis $\{\boldsymbol{w_x}, \boldsymbol{w_y}, \boldsymbol{w_z}\}$.
$\mathcal{B}$	Body Fixed Frame formed from the basis $\{\boldsymbol{b_x}, \boldsymbol{b_y}, \boldsymbol{b_z}\}$.
$\mathcal{S}$	Sensor Fixed Frame formed from the basis $\{\boldsymbol{s_x}, \boldsymbol{s_y}, \boldsymbol{s_z}\}$.
${}^{\mathcal{F}}\boldsymbol{v}$	A vector $\boldsymbol{v}$ described in frame $\mathcal{F}$. v_i corresponds to the i^{th} element of ${}^{\mathcal{F}}\boldsymbol{v}$.
${}^{\mathcal{F}}\bar{\boldsymbol{v}}$	A 2D vector projection of vector $\boldsymbol{v}$ on the basis forming the x-y plane of frame $\mathcal{F}$.
${}^{\mathcal{W}}_{\mathcal{B}}R$	An orthonormal rotation matrix that describes the orientation of frame $\mathcal{B}$ with respect to frame $\mathcal{W}$, ${}^{\mathcal{B}}_{\mathcal{W}}R = {}^{\mathcal{W}}_{\mathcal{B}}R^{-1} = {}^{\mathcal{W}}_{\mathcal{B}}R^{T}$.
$\mathcal{I}$	Diagonal matrix consisting of the UAs' inertia around the $\mathcal{B}$ principal axes, $= diag(\mathcal{I}_x, \mathcal{I}_y, \mathcal{I}_z)$.
I_n	Square diagonal identity matrix of size $n \times n$.
$\mathcal{N}(0, R)$	Zero mean normally distributed noise with variance R.
$\mathcal{O}$	Observability matrix.
$\boldsymbol{\alpha}$	Rotational acceleration of the UA.
${}^{\mathcal{B}}\bar{\boldsymbol{\alpha}}_{\boldsymbol{t}}$	Inertia normalized thrust generated moment vector, $= \bar{\boldsymbol{M}} \cdot [\, \mathcal{I}_x^{-1}\ \mathcal{I}_y^{-1}\ ^{T}$.
χ	$= T_{\lambda_z}^{-1} T_{prop}^{-1} K_{eq_z}$.
Γ_p	Diagonal matrix of rotational profile drag coefficients in the $\mathcal{B}$ frame, $= diag(\gamma_{p,x}, \gamma_{p,y}, \gamma_{p,z})$.
$\bar{\Gamma}$	Diagonal matrix of total rotational drag coefficients in the $\mathcal{B}$ frame, $= diag(\gamma_x, \gamma_y)$.
Λ_p	Diagonal matrix of translational profile drag coefficients in the $\mathcal{B}$ frame, $= diag(\lambda_{p,x}, \lambda_{p,y}, \lambda_{p,z})$.
Λ	Diagonal matrix of total translational drag coefficients in the $\mathcal{B}$ frame, $= diag(\lambda_x, \lambda_y, \lambda_z)$.

Manuscript received 31 March 2022; revised 10 May 2022; accepted 15 May 2022. Date of publication 21 June 2022; date of current version 14 July 2022. This work was supported by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS and Award CIRA-2020-082. The associate editor coordinating the review of this article and approving it for publication was Dr. Yulong Huang. *(Mohamad Wahbah and Mohamad Chehadeh contributed equally to this work.) (Corresponding author: Mohamad Chehadeh.)*

Mohamad Wahbah, Mohamad Chehadeh, Mahmoud Hamandi, and Yahya Zweiri are with the Khalifa University Center for Autonomous Robotic Systems (KUCARS), Department of Aerospace Engineering, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: Mohamad.chehadeh@ku.ac.ae).

Lakmal Seneviratne is with the Department of Mechanical Engineering, KUCARS, Abu Dhabi, United Arab Emirates.

Digital Object Identifier 10.1109/JSEN.2022.3183187

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

μ_i Rotational velocity of the i^{th} rotor.
$\boldsymbol{\omega}$ Rotational velocity of the UA.
τ Time delay quantity. τ_{b_x}, τ_{b_y}, and τ_z are associated with closed loop delays of the rotational dynamics around $\boldsymbol{b_x}$, around $\boldsymbol{b_y}$, and along $\boldsymbol{b_z}$ respectively.
θ Angle of rotation around $\boldsymbol{w_x}$, referred to as the roll of the UA.
ϕ Angle of rotation around $\boldsymbol{w_y}$, referred to as the pitch of the UA.
ψ Angle of rotation around $\boldsymbol{w_z}$, referred to as the yaw of the UA.
$\boldsymbol{a}$ Linear acceleration of the UA.
$\boldsymbol{a}_{bs}$ Body specific accelerations of the UA.
a_T Mass normalized generated thrust, $= \frac{^{\mathcal{B}}\boldsymbol{F_t}\cdot\boldsymbol{b_z}}{m}$.
$\boldsymbol{g}$ Gravity field vector with $^{\mathcal{W}}\boldsymbol{g} = [0, 0, a_g]^T$, and where a_g is the gravitational acceleration constant.
k_T Thrust coefficient of the rotor.
k_C Motor command to rotational velocity coefficient.
m Mass of the UA.
$\boldsymbol{p}$ Position of the UA.
$\boldsymbol{q}$ Quaternion, $= \{q_w, q_x, q_y, q_z\}$.
$\boldsymbol{q}_r$ Estimated orientation quaternion.
$\boldsymbol{r_i}$ Position of the center of the i^{th} motor relative to the UA's center of mass.
u_i Motor command sent to the i^{th} electronic speed controllers.
u_M Modified Relay Feedback Test output.
u_T Collective motor commands sent to all electronic speed controllers, generating thrust force along $\boldsymbol{b_z}$.
$\bar{\boldsymbol{u}}_{\boldsymbol{M}}$ Differential motor commands sent to opposing electronic speed controllers causing a moment around $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$ respectively, $= [u_{b_x}\ u_{b_y}]^T$.
$\bar{\boldsymbol{u}}_{\boldsymbol{M_{bias}}}$ Estimated bias in the differential motor commands sent to opposing electronic speed controllers.
$u_{T_{bias}}$ Collective motor command bias.
$\boldsymbol{v}$ Linear velocity of the UA.
$\boldsymbol{v}_i^h$ Velocity of the i^{th} rotor hub.
$\boldsymbol{x_R}$ State vector of the Rotational Decoupled Extended Kalman Filter (RDEKF).
$\boldsymbol{x_T}$ State vector of the Translational Decoupled Extended Kalman Filter (TDEKF).
$\mathbf{x}^{\mathbf{f}}$ Vector of filtered states.
$\mathbf{x}^{\mathbf{g}}$ Vector of ground truth states.
$\boldsymbol{F}$ Total force of applied to UA.
$\boldsymbol{F_d}$ Force applied due to drag.
$\boldsymbol{F_g}$ Force due to gravity.
$\boldsymbol{F_i}$ Thrust force of the i^{th} motor.
$\boldsymbol{F_m}$ Rotor motion inflow drag force due to rigid body translation.
$\boldsymbol{F_t}$ Total generated thrust due to the rotors' rotations.
$\bar{\boldsymbol{K}}_{\boldsymbol{eq}}$ Vector of equivalent gain for rotational dynamics around $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$ accounting for propulsion gain and inertia, $= [K_{eq_{b_x}}\ K_{eq_{b_y}}]^T$.
$K_{eq_{b_z}}$ Equivalent gain of the vertical motion dynamics, accounting for propulsion gain and inertia.
K_x, K_y Loop gain associated with lateral loop dynamics.
$^{\mathcal{B}}\boldsymbol{M}$ Total moment applied to UA.
$^{\mathcal{B}}\boldsymbol{M}_d$ Moment due to profile drag and rotor motion inflow.
$^{\mathcal{B}}\boldsymbol{M}_f$ Moment due to blade flapping.
$^{\mathcal{B}}\boldsymbol{M}_g$ Moment due to the gyroscopic effect.
$^{\mathcal{B}}\boldsymbol{M}_t$ Moment generated due to rotors' rotation.
T_Γ Diagonal matrix of the rotational time constants, $= diag(T_{\gamma_x}, T_{\gamma_y})$.
T_Λ Diagonal matrix of the translational time constants, $= diag(T_{\lambda_x}, T_{\lambda_y}, T_{\lambda_z})$.
$\bar{T}_{prop}$ Diagonal matrix associated with the propulsion system dynamics causing moment around $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$, $= diag(T_{prop_{b_x}}, T_{prop_{b_y}})$.
T_{prop} Time constant associated with the propulsion system dynamics.
β Modified Relay Feedback Test phase parameter.
Ψ Phase obtained from β.
ζ Tunable data generation parameters.
h Modified Relay Feedback Test amplitude.
D Subspace containing the time parameters of the processes defined in the DNN-MRFT, $d_i \in D$.
$\check{D}$ Discretized subspace from D adhering to the relative sensitivity criteria of the DNN-MRFT, $\check{d}_i \in \check{D}$.
J^* Relative sensitivity specifications.
M Map that generates data s_i from process parameters d_i, M^{-1} defines its inverse.
S Data feature space used for training, $s_i \in S$.

I. Introduction

UNMANNED Aircrafts (UA)s have seen increasing popularity in the last decade due to their versatility and high maneuverability compared to other robotic platforms. Due to their increased workspace compared to Unmanned Ground Vehicles (UGV)s and manipulators, UAs became an attractive option for a myriad of industries, including defense, agriculture, and entertainment. In the majority of these applications, the vehicle's autonomy is imperative to achieve the allocated tasks.

While autonomy requires a UA to be endowed with a combination of abilities, such as mission planning, trajectory tracking, optimal control, etc.; state estimation using on-board sensors is usually essential to be able to achieve the majority of these abilities, and thus achieve platform autonomy. As such, in what follows we will present a novel UA state estimator, that estimates decoupled translational and rotational pose of the platform. Moreover, the estimator incorporates the platform dynamics to be able to estimate in real-time the platform accelerations, while relying on off-the-shelf sensors.

A. Relevant Work

Most UAs used nowadays have an on-board Inertial Measurement Unit (IMU), and a source of position sensing, such as Real-Time Kinematics (RTK) systems [1], Motion Capture (MoCap) [2], Radio Frequency (RF) [3], [4], electromagnetic [5], and Ultra Wide Bandwidth (UWB) [6]), in addition to on-board measurements (e.g. visual odometry [7], [8]). IMUs usually consist of three axes gyroscopes, accelerometers, and magnetometers. The literature is mature with a variety of kinematic estimators that estimate the UA attitude from IMU measurements [9]–[11]. The shortcoming of these filters is that they consider quasi-stationary flight for certain assumptions on stability and performance to hold. These attitude estimators are usually augmented with position measurements to achieve several advantages such as higher update rates of position, velocity estimation, IMU biases correction, or providing a phase lead in position and velocity estimates.

Another class of filters use kinetics, in addition to the kinematic equations of motion, and as such can be referred to as dynamics based filters. Such filters utilize model knowledge to estimate additional states, or physical properties, and to provide smoother estimates. A large variety of such filters exist in the literature, with each group following a different approach. A notable contribution made by [12] explains how an accelerometer measures body specific accelerations only. These accelerations are caused by forces that affect the housing of the accelerometer, but not the sensor itself. In other words, the accelerometer measures the generated thrust, the induced drag, and other forces acting on the UA body, but not the gravity. The work of [13] utilized this fact to develop a tunable deterministic observer that requires the prior knowledge of translational drag to estimate the UA's drag. This drag estimate is later used in the UA's velocity estimation. In [14], the authors developed an Extended Kalman Filter (EKF) that estimates attitude angles, inertial lateral velocities, and gyroscope biases. The developed EKF utilizes the UA's under-actuation constraints to reduce drifts in velocity; however, their method assumes smooth flights (i.e. almost constant propeller rotational velocity) and requires a prior knowledge of the nominal rotor drag coefficient and the UA mass. The authors of [15] improved on [14] by achieving a real-time estimate of the lateral drag parameters, where these parameters were shown to be observable in [15]. The designed filters in [13]–[15] provided estimates of the lateral velocity, and the attitude only.

In [16], the work of [15] was extended to provide low-drift state estimates in three dimensions by incorporating the input to the motor command. While their method is promising, it required extensive pre-flight calibrations of motor's command to motor's thrust. Such mapping disregards motor dynamics, which would result in poor estimate of high-frequency components. Work in [17] showed that the inertia of the UA along with its motor's lift and torque constants can be estimated in real-time by an EKF using on-board measurements, without the need of a test bench. However, the corresponding model states were only observable under certain maneuvers that require manual operation by a pilot, or hand-tuning of the flight controller. Another dynamic filter was presented in [17], which required only prior knowledge of the drone mass, and the diagonal distances between the motors; however, this filter disregarded drag and motor dynamics, which contribute greatly to the platform motion, and correspondingly the state estimation. In the recent work of [18], a UKF was developed to estimate concurrently the lateral and vertical drag terms, the inertia of the UA, the inertia of the blades, the motor torque constant, the mass of the UA, along with its attitude and velocity. While their approach is promising, it requires the use of motor speed measurement sensors, which are not available in most commercial UAs. Similar to [17], the approach in [18] required the UA to preform aggressive maneuvers in order to estimate the UA inertial and drag parameters.

While the above methods estimated different dynamics of the corresponding platforms, none of them provides estimates for inertial accelerations, which can be quite useful for high performance feedback control as found in [19], [20]. The authors in [19] perform agressive maneuvers by utilizing inertial acceleration readings from a high-end IMU fused with thrust estimates from motor's rotational speed measurements. To mitigate the need for motor's rotational speed measurements, [20] proposed an adaptive notch filtering technique that finds the platform acceleration and propeller induced vibration directly from IMU measurements. While these two methods provide promising methods to accurately filter IMU measurements in real-time, both methods require accurate measurements of the platform's propellers' rotational speed. As such, it is not straightforward to apply them to different UAs, as many commercial platforms do not provide such measurements.

To conclude, there is a gap in the current UA state estimation literature in having a dynamics based estimator that can adapt in real-time for rotational and translational drag parameters, while providing smooth and lag-free inertial acceleration estimates. Estimators that can estimate model parameters online without requiring expensive sensors or additional measurement sources would enable mass deployment of high performance, and low-cost UAs. As such, the main motivation of this article is to provide an in-flight high-performance dynamics based estimators that does not require prior knowledge of the dynamics and uses sensory setup found on typical commercial UAs.

B. Contributions

Motivated by the above, this paper aims at presenting a novel UA dynamics based state estimator, that can be tuned online without any previous knowledge of the platform dynamics. The proposed estimation scheme, illustrated in Fig. 1, requires only an IMU and a position sensor, which is the standard configuration in mainstream autonomous UAs. The model based prediction and sensor fusion are implemented in two novel filters, which we refer to as the Rotational Decoupled Extended Kalman Filter (RDEKF) and the Translational Decoupled Extended Kalman Filter (TDEKF). Accordingly, the contributions of this paper can be summarized as follows:

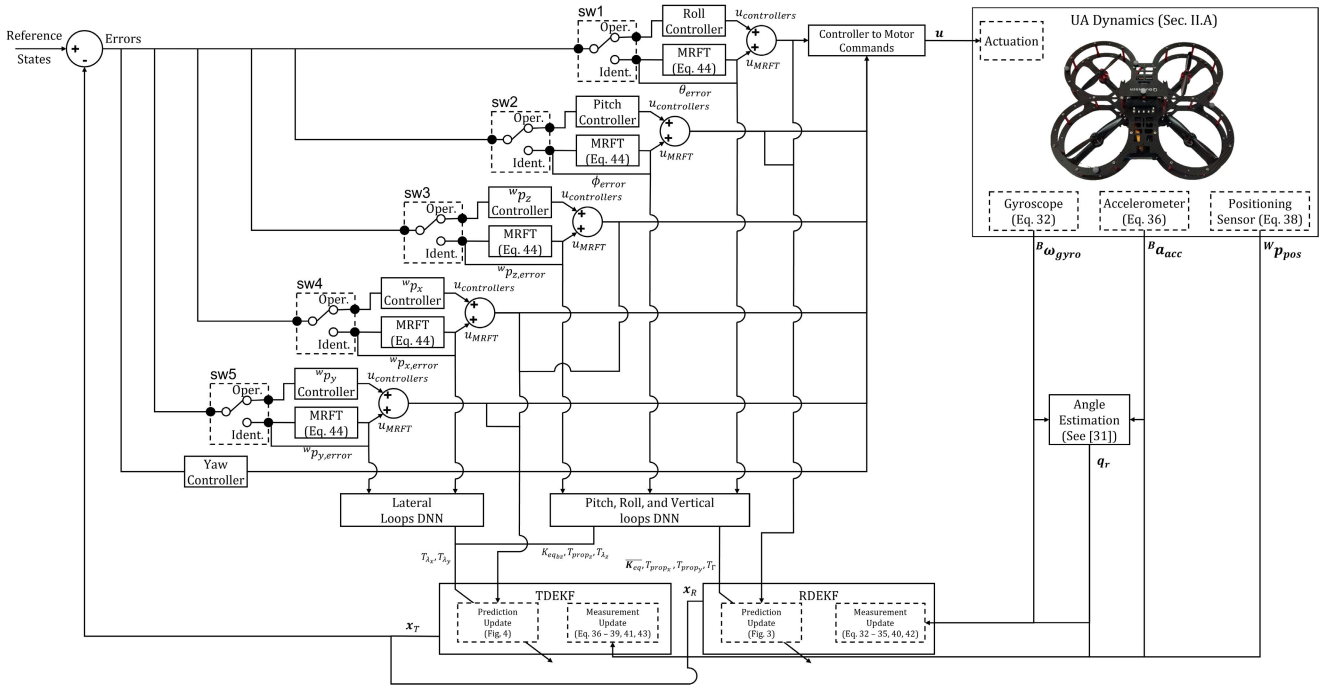


Fig. 1. Structure of the proposed estimation scheme. When a switch "sw x" is at position "Ident"., the identification phase is activated for that loop, and the Deep Neural Network Modified Relay Feedback Test (DNN-MRFT) framework is used to identify the model parameters of that loop. After the model parameters has been updated, the switch flips to position "Oper." so that the operation phase starts and the estimates are used to control the UA. Note that the controller outputs are used as inputs for the RDEKF and TDEKF prediction steps. The model parameters are identified and adapted in real-time in-flight.

1) Propose a general filtration approach to estimate a UA's translational and angular pose, while taking the platform and actuator dynamics into account.
2) The proposed framework is shown to be able to provide a smooth and lag-free estimate of the platform's rotational speeds and inertial accelerations, without requiring motor speed measurements as in [18] and [19].
3) As dynamics based filters for a nonlinear system are challenging to tune, this paper proposes the use of an AI method (the DNN-MRFT framework [21]) to estimate online, and in real-time, the platform's dynamics required by the proposed filter. This alleviates the need for expensive pre-flight calibrations as required by [16].
4) The proposed filter and tuning approach are validated in an extensive experimental campaign that shows the advantage of the proposed approach in estimating the platform's angular speeds and inertial acceleration and the ensuing platform's pose. In addition, the experimental campaign clearly showcases the use of the DNN-MRFT framework for the tuning of the proposed filter. A video that summarizes the conducted experiments of this paper is available in [22].

Our experimental campaign shows that the proposed RDEKF and TDEKF state estimates can be used to perform multiple flight trajectories. During this campaign, we demonstrate how the RDEKF and TDEKF provide significantly smoother and lag-free estimates of the rotational speeds and inertial accelerations, achieving an average Root Mean Square Error (RMSE) of 0.1532 m/s^2 and 0.0459 rad/s respectively, as seen in Section V-B. These results represent a significant improvement compared to raw gyro and accelerometer measurements. Hence, it follows that the use of RDEKF and TDEKF resulted in better reference tracking achieving 43% reduction in ramp reference tracking, as well as reducing the controller action by approximately 6.6% when tested on a figure-eight trajectory. In addition, we execute a high speed figure-eight maneuver using low rate position measurements of 10 Hz only. Such maneuvers are usually infeasible without high rate measurements, yet by using the proposed filters, the maneuver was executed successfully and with good tracking performance. Such experiment illustrates how our approach can enable accurate tracking using low frequency position measurement systems, such as on-board cameras, or RTK systems. The full closed-form model and the linearization steps are included in this document, and can be found in a detailed workbook in [23].

C. Structure of the Article

This article is organized as follows. In Section II we summarize the nonlinear dynamic model of the UA, and derive the corresponding semi-decoupled linear kinetics and nonlinear kinematics model. Then in Section III we present the design of the RDEKF and TDEKF based on the developed model. In Section IV we briefly describe how the dynamic model parameters can be tuned using the DNN-MRFT approach. The proposed approach is validated through experimental results, shown in Section V. Finally, Section VI concludes the article.

II. System Model

Kalman filtering utilizes the knowledge of the model to predict the states of the system. As such, using a prediction model that closely matches the actual UA physical plant will lead to better noise filtering, and estimation. Detailed models can be achieved by considering all the dynamics that affect the UA, which would result in accurate estimation of the evolved states over the prediction horizon. However, such detailed models are expensive to obtain (i.e. require extensive lab and flight tests), in addition to being computationally expensive to estimate, and thus difficult to use in a real-time estimator. A better approach is to use lumped models with parameters that describe accurately the evolution of the states in the prediction horizon, while still being possible to compute in real-time. In this section, we first explore the non-linear UA model found in literature. We then present our proposed linearized SISO models that greatly simplify the full model computation, while retaining estimation accuracy. Finally, we introduce the linearized models used by RDEKF and TDEKF and show how the parameters identified by DNN-MRFT can be used in the estimation models.

A. Nonlinear Model

It is necessary to model all forces acting on the UA body to relate the measurements obtained by the inertial sensors to our proposed model. As such, we aim first to develop a nonlinear model that captures said forces, then introduce the necessary assumptions and linearizations to achieve real-time performance, and to be able to use DNN-MRFT for parameter identification. The reference frames used in this work are defined in the Nomenclature and are shown in Fig. 2 for illustration.

For a body motion that abides to Newtonian dynamics, the summation of forces is:

$$m\boldsymbol{a} = \sum \boldsymbol{F} \tag{1}$$

where $\boldsymbol{F}$ is any force applied to the body. For a UA, this can be expanded into:

$$m\boldsymbol{a} = \boldsymbol{F_t} - \boldsymbol{F_d} - \boldsymbol{F_g} \tag{2}$$

The total generated thrust $\boldsymbol{F_t}$ is due to the rotors' rotations, and can be modeled as [24]:

$${}^{\mathcal{B}}\boldsymbol{F_t} = k_T \sum_{i=1}^{4} \mu_i^2 \boldsymbol{b_z} \tag{3}$$

where a rotor's rotation is related to the Electronic Speed Controller (ESC) input command by:

$$\mu_i = k_C u_i \tag{4}$$

where u_i is the motor command sent to the i^{th} motor. Note that in this article, we will also use u_i with $i \in \{b_x, b_y, z\}$ to conveniently describe differential, or collective motor thrust commands that correspond to the outputs of the individual inner control loops, for rotations around b_x and b_y, and collective thrust.

The induced translational drag forces $\boldsymbol{F_d}$ consist of two parts as seen from $\mathcal{B}$ [24], [25]. The first part of the drag is lateral

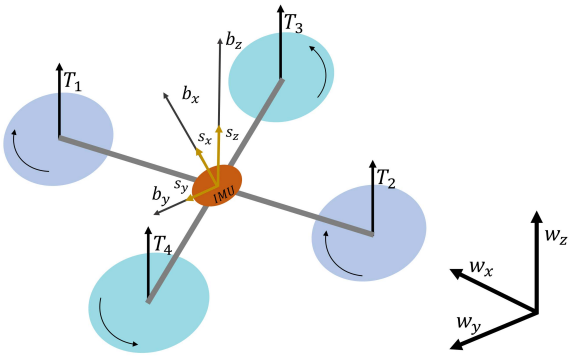


Fig. 2. Notation and reference frames used in this work. Without loss of generality, $\mathcal{B}$ is defined assuming "X" quadrotor configuration.

(i.e. co-planar with $\boldsymbol{b_x} \times \boldsymbol{b_y}$) and is mainly attributed to blade flapping, and profile drag on the rotors and the UA's body. The second part of the drag, which is the prominent one, is along $\boldsymbol{b_z}$ and is mainly caused by the change in inflow angle, and the profile drag on the rotors and the UA body. The change in inflow angle changes the thrust coefficient k_T, and was approximated in [24] for the i^{th} motor to be:

$$k_{T_i} = k_{T_0} + \Delta k_{T_i} \tag{5}$$

where the change increment, Δk_{T_i}, is given by:

$$\Delta k_{T_i} = \frac{c_r}{\mu_i} {}^{\mathcal{B}}\boldsymbol{v}_i^h \tag{6}$$

where c_r is a constant of the rotor's physical properties, and where ${}^{\mathcal{B}}\boldsymbol{v}_i^h$ is given by:

$${}^{\mathcal{B}}\boldsymbol{v}_i^h = {}^{\mathcal{B}}\boldsymbol{v} + {}^{\mathcal{B}}\boldsymbol{\omega} \times \boldsymbol{r_i} \tag{7}$$

Thus, rotor motion inflow not only damps translations along $\boldsymbol{b_z}$, but also rotational motion by producing opposing moments. It follows that $\boldsymbol{F_d}$ can be modeled as [24], [25]:

$${}^{\mathcal{B}}\boldsymbol{F_d} = ({}^{\mathcal{B}}\boldsymbol{F_m} \cdot \boldsymbol{b_z})\boldsymbol{b_z} + \Lambda_p {}^{\mathcal{B}}\boldsymbol{v} \tag{8}$$

Finally, the gravity force is given by:

$${}^{\mathcal{B}}\boldsymbol{F_g} = m {}^{\mathcal{B}}\boldsymbol{g} \tag{9}$$

On the other hand, the rotation dynamics of the UA are given by:

$$\mathcal{I}\frac{d}{dt}({}^{\mathcal{B}}\boldsymbol{\omega}) = {}^{\mathcal{B}}\boldsymbol{M}_t - {}^{\mathcal{B}}\boldsymbol{M}_g - {}^{\mathcal{B}}\boldsymbol{M}_d - {}^{\mathcal{B}}\boldsymbol{M}_f \tag{10}$$

The moment vector ${}^{\mathcal{B}}\boldsymbol{M}_t$, generated by the rotors due to rotation is [26]:

$${}^{\mathcal{B}}\boldsymbol{M}_t = \sum_{i=1}^{4} \left(\boldsymbol{r_i} \times \boldsymbol{F_i} - d(i) k_m \mu_i^2 \boldsymbol{b_z}\right) \tag{11}$$

where $d(i)$ is a function of the motor direction, that returns one for counter-clock wise rotations, and negative one for clock wise rotations, and k_m is a positive coefficient. The gyroscopic moments are given by [25]:

$${}^{\mathcal{B}}\boldsymbol{M}_g = {}^{\mathcal{B}}\boldsymbol{\omega} \times \mathcal{I}\, {}^{\mathcal{B}}\boldsymbol{\omega} \tag{12}$$

TABLE I
MAPPING OF NONLINEAR MODEL PARAMETERS TO THEIR REPRESENTING LUMPED LINEARIZED MODEL PARAMETERS. THESE LUMPED PARAMETERS ARE OBSERVABLE THROUGH DNN-MRFT REAL-TIME IDENTIFICATION ALGORITHM. THE USE OF THE OBTAINED LUMPED PARAMETERS IN THE PREDICTION STEP OF TDEKF AND RDEKF IS SHOWN IN FIG. 4 AND FIG. 3 RESPECTIVELY

Nonlinear model parameter(s)	Representing lumped linearized model parameter	Comments
$\frac{k_T k_C}{m}$ in Eqs. (2), (3), (4)	$K_{eq_{b_z}}$ in Eq. (15)	Used in TDEKF
${}^{\mathcal{B}}\boldsymbol{F_m} \cdot \boldsymbol{b_z}$ and $\lambda_{p,z}$ in Eq. (8)	T_{λ_z} in Eq. (15)	Used in TDEKF
$\lambda_{p,x}$, $\lambda_{p,y}$ in Eq. (8), and k_f, μ_i with $i \in \{1,2,3,4\}$ in Eq. (14)	T_{λ_i} with $i \in \{x, y\}$ in Eq. (17)	Used in TDEKF
k_T in Eq. (3), k_C in Eq. (4), $\mathcal{I}$ in Eq. (10), and $\boldsymbol{r_i}$ in Eq. (11)	$\bar{\boldsymbol{K}}_{\boldsymbol{eq}} = [K_{eq_{b_x}} \ K_{eq_{b_y}}]^T$ in Eq. (16)	Used in RDEKF
${}^{\mathcal{B}}\boldsymbol{M}_m$ and $\boldsymbol{\Gamma_p}$ in Eq. (13)	T_{γ_x} and T_{γ_y} in Eq. (16)	Used in RDEKF

and the moments caused by the profile drag and motion inflow are given by [24]:

$$ {}^{\mathcal{B}}\boldsymbol{M}_d = {}^{\mathcal{B}}\boldsymbol{M}_m + \Gamma_p {}^{\mathcal{B}}\boldsymbol{w} \tag{13} $$

where ${}^{\mathcal{B}}\boldsymbol{M}_m$ is a vector of the moments due to the motion inflow caused by angular rotation, as presented in Eq. (6). Lastly, the moments caused by blade flapping are adopted from [18]:

$$ {}^{\mathcal{B}}\boldsymbol{M}_f = \sum_{i=1}^{4} k_f \ \mu_i {}^{\mathcal{B}}\boldsymbol{v}_i^h \times \boldsymbol{b_z} \tag{14} $$

where k_f is a positive coefficient.

B. Linearized and Loosely Coupled Models

Designing a dynamics based estimator based on the form of nonlinear coupled equations presented in Eqs.(2) and (10) poses two challenges. The first challenge is associated with the computational and tuning complexity of a high dimensional strongly coupled EKF estimator. The second challenge is related to the identification or estimation of the physical parameters of the UA. We solve the first challenge by loosening the coupling between the rotational and translational dynamics, and thus ignoring gyroscopic moments presented in Eq. (12). It is argued in [27] that the effect of the gyroscopic moments can be neglected for the range of the physical parameters of the common multirotor UA sizes and designs. The only coupling that remains is due to kinematics, which cannot be ignored. The second challenge is solved by performing accurate real-time identification of lumped system parameters through DNN-MRFT. We refer to them as lumped parameters due to the fact that a single parameter may be used to capture a few physical phenomena that have the same effect from the dynamics perspective. For example, drag from motion inflow and profile drag are described by one lumped parameter. Table II summarizes the mapping of nonlinear model parameters to their representing lumped linearized model parameters. Linear drag models were found to be good approximates of the underlying physics even at translational speeds of few meters per second [21], [24], [28], [29] which justifies the linearization of nonlinear drag terms.

The model structures used for identification were proposed by [27] and confirmed experimentally in [21], [28], [30]. The vertical motion dynamics (i.e. the movement along $\boldsymbol{b_z}$), are given by [27]:

$$ \begin{aligned} &T_{prop_{b_z}} T_{\lambda_z} {}^{\mathcal{B}}\dddot{p}_z(t) + (T_{prop_{b_z}} + T_{\lambda_z}) {}^{\mathcal{B}}\ddot{p}_z(t) + {}^{\mathcal{B}}\dot{p}_z(t) \\ &\quad = K_{eq_{b_z}} u_T(t - \tau_z) \end{aligned} \tag{15} $$

with $K_{eq_{b_z}} = K_{prop_{b_z}} K_z$. Attitude dynamics corresponding to rotations around $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$ have the same model structure of vertical motion dynamics, and can be modeled as [27]:

$$ \begin{aligned} &T_{prop_i} T_{\gamma_j} {}^{\mathcal{B}}\ddot{\omega}_j(t) + (T_{prop_i} + T_{\gamma_j}) {}^{\mathcal{B}}\dot{\omega}_j(t) + {}^{\mathcal{B}}\omega_j(t) \\ &\quad = K_{eq_i} u_i(t - \tau_i) \end{aligned} \tag{16} $$

with $i \in \{b_x, b_y\}$, $j \in \{x, y\}$, and $\bar{\boldsymbol{K}}_{\boldsymbol{eq}} = [K_{eq_{b_x}} \ K_{eq_{b_y}}]^T = [K_{prop_{b_x}} K_{b_x} \ K_{prop_{b_y}} K_{b_y}]^T$.

The time delay terms appearing in Eqs. (15) and (16) are used to capture propulsion, processing, communication and measurement delays. The time delay is distributed among the forward (e.g. delay in propulsion) and feedback (e.g. delay in measurement) paths of the control loop. The DNN-MRFT is an input-output identification method, hence it cannot distinguish these delays. For the simplicity of computation we assume that all delays are present in the forward path.

Lateral motion dynamics, i.e. motion along $\boldsymbol{w_x}$ or $\boldsymbol{w_y}$, cascade attitude dynamics, due to underactuation, and are modeled as follows [30]:

$$ \begin{aligned} T_{\lambda_x} {}^{\mathcal{W}}\ddot{p}_x(t) + {}^{\mathcal{W}}\dot{p}_x(t) &= K_x \phi(t) \\ T_{\lambda_y} {}^{\mathcal{W}}\ddot{p}_y(t) + {}^{\mathcal{W}}\dot{p}_y(t) &= K_y \theta(t) \end{aligned} \tag{17} $$

Quaternions are used to properly handle rotational speed integration, to obtain UA attitude as needed in Eq. (17), through the following formulas:

$$ \begin{aligned} \dot{\boldsymbol{q}} &= \frac{1}{2} \boldsymbol{q} \otimes \left[0, \quad {}^{\mathcal{B}}\omega_x, \quad {}^{\mathcal{B}}\omega_y, \quad {}^{\mathcal{B}}\omega_z\right] \\ \boldsymbol{q}_t &= \boldsymbol{q}_{t-1} \oplus \dot{\boldsymbol{q}} \Delta t \end{aligned} $$

where $\boldsymbol{q}_t$ is the new estimate of the orientation, $\boldsymbol{q}_{t-1}$ is the previous estimate, and Δt is the integration time step. It should be noted that this work uses the Hamilton quaternion convention.

Note that we have omitted the yaw moment as estimation of the yaw dynamics is not of interest in this work, where the focus is devoted to fast transient dynamics that are present in aggressive maneuvers. Moreover, control and estimation for yaw dynamics is simpler due to the presence of full state

measurements and the lower relative degree of the system [30]. Therefore, in this work, we are using a kinematic estimator for the yaw states based on a modified version of the implementation of [31] that allows absolute yaw measurements and yaw body rates to be fused.

III. Decoupled Dynamics Based Estimation Scheme

Due to the nonlinear properties brought by the coupled kinematics and the quaternion integration, a nonlinear estimator needs to be used. Extended Kalman Filters solve this issue by linearizing the prediction and the measurement models to provide corrected estimates. In this section, we go over the construction of the two proposed Extended Kalman Filters, and their respective prediction and measurement steps.

Let us consider the state-space model for an arbitrary nonlinear system with the states $\boldsymbol{x}(t)$, an input $\boldsymbol{u}(t)$, and additive process and measurement noises:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \mathcal{N}(0, R_{proc}) \tag{18}$$

$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \mathcal{N}(0, R_{meas}) \tag{19}$$

The process noise variance is R_{proc}, and the measurement noise variance is R_{meas}. Equation (18) describes the change in the system caused by its current state, and the input it is subjected to, while Eq. (19) shows how the measurements change as the states and input change.

To be able to use the EKF, let us linearize the state equation f and the output equation h around the current states $\boldsymbol{x}$ at time t, and re-write the generalized model equations (18,19) to obtain the following linear time varying (LTV) system:

$$\dot{\boldsymbol{x}}(t) = A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) + \mathcal{N}(0, R_{proc}) \tag{20}$$

$$\boldsymbol{y}(t) = C(t)\boldsymbol{x}(t) + D(t)\boldsymbol{u}(t) + \mathcal{N}(0, R_{meas}) \tag{21}$$

where $A(t)$ is the state matrix, $B(t)$ is the input matrix, $C(t)$ is the output matrix, and $D(t)$ is the feedforward matrix. The forms of these matrices for the RDEKF and TDEKF are detailed in the appendix.

Finally, the system above is discretized to be able to apply the prediction and correction steps. The prediction step shown below estimates the system states at the next iteration from the current system states and input:

$$\tilde{\boldsymbol{x}}_{n+1} = F_n \boldsymbol{x}_n + G_n \boldsymbol{u}_n \tag{22}$$

where:

$$F_n = e^{A(t)\Delta t} \tag{23}$$

$$G_n = \int_0^{\Delta t} e^{A(t)\Delta t}\, dt\ B(t) \tag{24}$$

with Δt being the filter time step.

On the other hand, the correction step shown below corrects the estimate of the system states to reduce the error between the measured and estimated states

$$\hat{\boldsymbol{x}}_n = \tilde{\boldsymbol{x}}_n + K_n(z_n - H\tilde{\boldsymbol{x}}_n) \tag{25}$$

where K_n is the Kalman Gain, z_n is the state measurement, and where we assumed that $H = C$ is a constant matrix independent of the system states and correction step, and that $D = \mathbf{0}$.

A. Estimator Design

Let's donate $\boldsymbol{x_R}$ as the states vector of the RDEKF, and $\boldsymbol{x_T}$ as the state vector of the TDEKF. The vector $\boldsymbol{x_R} \in \mathbb{R}^{12\times 1}$ is defined as follows:

$$\boldsymbol{x_R} = \begin{bmatrix} \boldsymbol{q} & {}^{\mathcal{B}}\bar{\boldsymbol{\omega}} & {}^{\mathcal{B}}\bar{\boldsymbol{\alpha}} & {}^{\mathcal{B}}\bar{\boldsymbol{\alpha}}_{\boldsymbol{t}} & \bar{\boldsymbol{u}}_{\boldsymbol{M_{bias}}} \end{bmatrix}^T \tag{26}$$

The process bias $\bar{\boldsymbol{u}}_{\boldsymbol{M_{bias}}}$ is used to offset rotations caused by motor mismatch, and weight imbalance. The states in $\boldsymbol{x_R}$, apart from $\boldsymbol{q}$, are all defined as 2D vectors, *i.e.,*the projection of the corresponding vector on $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$ respectively. The angle ψ is assumed to be measured externally, and $\boldsymbol{q}$ is updated accordingly.

The TDEKF estimates the states vector, $\boldsymbol{x_T} \in \mathbb{R}^{11\times 1}$ defined as follows:

$$\boldsymbol{x_T} = \begin{bmatrix} {}^{\mathcal{W}}\boldsymbol{p} & {}^{\mathcal{W}}\boldsymbol{v} & {}^{\mathcal{B}}\boldsymbol{a}_{bs} & a_T & u_{T_{bias}} \end{bmatrix}^T \tag{27}$$

The process bias $u_{T_{bias}}$ compensates for the slow drift in thrust command caused by battery voltage drop. These two states are modeled as scalar quantities rather than a vector, as the generated thrust is always aligned with $\boldsymbol{b_z}$. The rest of the estimates in $\boldsymbol{x_T}$ are three-dimensional quantities.

B. Prediction Model

Eq. (28) and Eq. (29) shows respectively the nonlinear dynamic system of the RDEKF and TDEKF, while the details corresponding to the measurement model of the two systems are discussed in the next subsection. The two system models are also illustrated in Figs. 3 and 4 for clarity.

Let us denote by $f(\boldsymbol{x}_R(t), \bar{\boldsymbol{u}}_M, {}^{\mathcal{B}}\omega_z)$ as the function describing the nonlinear dynamics of the RDEKF shown below:

$$\begin{aligned}
\dot{\boldsymbol{q}} &= \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0, & {}^{\mathcal{B}}\omega_x, & {}^{\mathcal{B}}\omega_y, & {}^{\mathcal{B}}\omega_z \end{bmatrix} \\
\dot{\bar{\boldsymbol{\omega}}} &= \bar{\boldsymbol{\alpha}} \\
\dot{\bar{\boldsymbol{\alpha}}} &= \bar{T}_{prop}^{-1}[T_\Gamma^{-1}\bar{\boldsymbol{K}}_{eq}^T(\bar{\boldsymbol{u}}_M - \bar{\boldsymbol{u}}_{M_{bias}}) - \bar{\boldsymbol{\alpha}}_t] - \bar{\Gamma}^{-1}\bar{\boldsymbol{\alpha}} \\
\dot{\bar{\boldsymbol{\alpha}}}_t &= \bar{T}_{prop}^{-1}[T_\Gamma^{-1}\bar{\boldsymbol{K}}_{eq}^T(\bar{\boldsymbol{u}}_M - \bar{\boldsymbol{u}}_{M_{bias}}) - \bar{\boldsymbol{\alpha}}_t] \\
\dot{\bar{\boldsymbol{u}}}_{M_{bias}} &= 0
\end{aligned} \tag{28}$$

Then, let us denote by $f(\boldsymbol{x}_T(t), u_T(t), {}^{\mathcal{W}}_{\mathcal{B}}R)$ as the function describing the nonlinear dynamics of the TDEKF shown below:

$$\begin{aligned}
\dot{\boldsymbol{p}} &= \boldsymbol{v} \\
\dot{\boldsymbol{v}} &= {}^{\mathcal{W}}_{\mathcal{B}}R\boldsymbol{a}_{bs} \\
\dot{\boldsymbol{a}}_{bs} &= \Lambda^{-1}\boldsymbol{a}_{bs} - a_T T_{prop}^{-1}\boldsymbol{w}_z + \chi(u_T - u_{T_{bias}})\boldsymbol{w}_z + {}^{\mathcal{W}}_{\mathcal{B}}R^T\boldsymbol{g} \\
\dot{a}_T &= -T_{prop}^{-1}a_T + \chi(u_T - u_{T_{bias}}) \\
\dot{u}_{T_{bias}} &= 0
\end{aligned} \tag{29}$$

In the case of the RDEKF, $\boldsymbol{u}(t)$ consists of $u_{b_x}(t)$ and $u_{b_y}(t)$, while for the TDEKF $\boldsymbol{u}(t)$ is $u_T(t)$, and the gravity vector $\boldsymbol{g}$. The gravity is treated as an input for representation sake; while the filter is running it is assumed to be constant.

The above nonlinear systems can be linearized by computing the Jacobian of the corresponding f functions; the resultant LTV system is reported in the appendix for clarity.

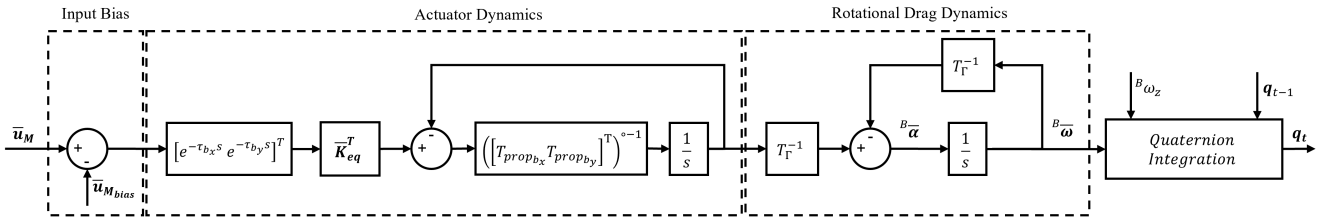


Fig. 3. Estimator prediction model used in the RDEKF.

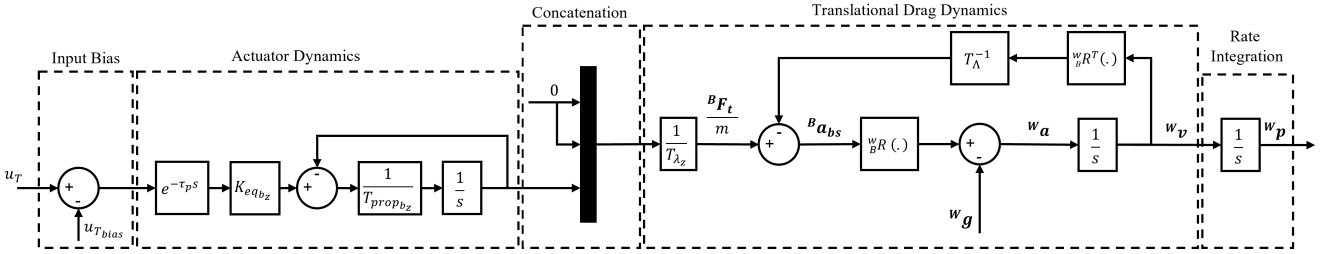


Fig. 4. Estimator prediction model used in the TDEKF.

C. Measurement Model

The proposed RDEKF and TDEKF fuses multiple state measurements to correct the predicted states as detailed below. The TDEKF is updated with acceleration measurements from a 3-axis accelerometer, in addition to position measurements. The position measurements could be acquired from either intrinsic measurements (such as GPS), or extrinsic measurements (such as a MoCap system). The RDEKF is updated with angular velocity measurements from a 3-axis gyroscope, in addition to orientation measurements. For the orientation measurements, we utilized a modification of the orientation estimation approach presented in [31] to provide angle estimates, which are used by the TDEKF and RDEKF as measurement sources. The gains in the orientation estimation filter are scheduled to depend greatly on the gyroscope measurements during aggressive maneuvers.

In addition to the above measurements, the RDEKF and TDEKF are designed to accommodate for propeller thrust measurements; these measurements are usually derived from the measurement of the propeller rotational speeds. Since these measurements are not available in all UAs, in what follows we will show how to incorporate it in the correction model, and later we will show that the proposed filter can still work even when such measurements are not available.

Finally, since the inertial sensors sampling rates are usually much higher from that of a position sensor, the multi-rate sampling approach described in [32] is used.

To incorporate the measurements in the filters, we need to provide measurement models that relate the measured quantities, to the filter estimates. The gyroscope measurements model adopted in this work is:

$$^{\mathcal{B}}\boldsymbol{\omega}_{gyro} = ^{\mathcal{B}}\boldsymbol{\omega} + \mathcal{N}(0, R_{gyro}) \tag{30}$$

where R_{gyro} is the additive noise variance. The gyroscope directly measures the rotational velocity of the UA, thus its observation matrix is:

$$H_{gyro} = \begin{bmatrix} 0_{2\times4} & I_2 & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \end{bmatrix} \tag{31}$$

The orientation measurements can be modeled to have an additive measurement noise with variance R_{ang}:

$$\boldsymbol{q}_r = \boldsymbol{q} + \mathcal{N}(0, R_{ang}) \tag{32}$$

The orientation provided by the corresponding measurement directly relates to the quaternion estimate from the filter, thus the orientation observation matrix is trivial:

$$H_r = \begin{bmatrix} I_4 & 0_{4\times2} & 0_{4\times2} & 0_{4\times2} & 0_{4\times2} \end{bmatrix} \tag{33}$$

The accelerometer measurement model used in the TDEKF is:

$$^{\mathcal{B}}\boldsymbol{a}_{acc} = \boldsymbol{a}_{bs} + \mathcal{N}(0, R_{acc}) \tag{34}$$

where R_{acc} is the measurement noise variance. The accelerometer gives a direct measurement of $\boldsymbol{a}_{bs}$, hence, the accelerometer observation matrix has a trivial derivation:

$$H_{acc} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times1} & 0_{3\times1} \end{bmatrix} \tag{35}$$

It should be noted that the accelerometer is assumed to be perfectly aligned with the UA's body. Hence, it provides measurements of the UA acceleration, without the need for any transform. Additionally, the accelerometer is assumed to be calibrated using six-point tumble method [33], and that any corresponding calibration bias has been eliminated.

The position measurement model depends on the sensor used. In the experimental part of this work, we use a MoCap system. We also simulated other positioning systems by reducing the update rate of the MoCap system to match typical arrangements usually found in deployed UAs; e.g. an RTK receiver with centimeter level accuracy, or a UWB localization system. The measurement model for the aforementioned systems is:

$$\boldsymbol{p}_{pos} = ^{\mathcal{W}}\boldsymbol{p} + \mathcal{N}(0, R_{pos}) \tag{36}$$

where R_{pos} is the noise variance, and the observation matrix is

$$H_{pos} = \begin{bmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \end{bmatrix} \tag{37}$$

Finally, the measurement model for the inertia normalized thrust generated moment $\bar{\boldsymbol{\alpha}}_t$ and the mass normalized generated thrust a_T are shown below:

$$\bar{\boldsymbol{\alpha}}_{t,u_{meas}} = \bar{\boldsymbol{\alpha}}_t + \mathcal{N}(0, R_{\bar{\boldsymbol{\alpha}}_t}) \tag{38}$$

$$a_{T,u_{meas}} = a_T + \mathcal{N}(0, R_{a_T}) \tag{39}$$

where u_{meas} is the measured propeller thrust, and $R_{\bar{\boldsymbol{\alpha}}_t}$ and R_{a_T} are respectively the noise variance of the measurement of $\bar{\boldsymbol{\alpha}}_t$ and a_T. The corresponding observation matrices are as follows:

$$H_{\bar{\boldsymbol{\alpha}}_t} = \begin{bmatrix} 0_{2\times4} & 0_{2\times2} & 0_{2\times2} & I_2 & 0_{2\times2} \end{bmatrix} \tag{40}$$

$$H_{a_T} = \begin{bmatrix} 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & I_1 & 0 \end{bmatrix} \tag{41}$$

D. Observability of TDEKF and RDEKF

A system is said to be observable if, for any two sets of states and inputs, the corresponding measurements are distinct [34]. As such, if a system is observable, and given distinct observations and known inputs, we can deduce the underlying state vectors. On the other hand, a system is said to be detectable if, all unobservable system states are stable.

The full observability analysis of the TDEKF and RDEKF systems are reported in the appendix for clarity, and the main findings are reported in what follows.

As it seems, in the case where a_T and $\bar{\boldsymbol{\alpha}}_t$ are measurable (*i.e.,*the platform's propeller rotational speeds are not measurable), the system is fully observable, with the corresponding observability matrices having rank equal to the dimensionality of the system *i.e.,*$\text{rank}(\mathcal{O}_{T_{a_T}}(t)) = 10$ and $\text{rank}(\mathcal{O}_{T_{\bar{\boldsymbol{\alpha}}_t}}(t)) = 12$ for all t.

On the other hand, if a_T and $\bar{\boldsymbol{\alpha}}_t$ are not measurable, the corresponding system is not observable. A canonical decomposition of the TDEKF system shows that in this case, the linear addition of a_T and $u_{T_{bias}}$ is observable while their linear difference is not observable. Moreover, the eigenvalue corresponding to the unobservable state is zero. As such, the unobservable state is not stable and the system is not detectable. A similar analysis, with similar conclusions, can be made in the case of the RDEKF and the corresponding linear combinations of $\bar{\boldsymbol{\alpha}}_t$ and $\bar{\boldsymbol{u}}_{M_{bias}}$.

During our experimental campaign, we observed that unobserved state (linear difference of a_T and $u_{T_{bias}}$ in the case of the TDEKF, and $\bar{\boldsymbol{\alpha}}_t$ and $\bar{\boldsymbol{u}}_{M_{bias}}$ in the case of the RDEKF) simply integrate any bias present in the system. As our filter is tuned online, such biases remain small, and the corresponding value of the unobserved state remain manageable if the system is run for a short period of time. However, and to avoid any unexpected behavior, when propeller rotational speeds are not measurable, we assume $u_{T_{bias}} = 0$ and $\bar{\boldsymbol{u}}_{M_{bias}} = \mathbf{0}$, and we incorporate these assumptions in the measurement model. Note that the resulting system is observable even in the absence of propeller's rotational speed measurements.

IV. Identification of Model Parameters Through DNN-MRFT

The DNN-MRFT approach was suggested in [21] to perform real-time identification and near-optimal tuning of UA control loops. The objective of the DNN is to find the unknown dynamic parameters of the unmanned aircraft (UA) based on input data generated by the MRFT controller [35]. MRFT excites a stable periodic motion. Measured periodic system output is then fed to a deep neural network (DNN) which classifies the unknown process and provides the corresponding system parameters. In this section, we present the DNN-MRFT approach and its specific implementation when applied to find the dynamic parameters in the RDEKF and TDEKF prediction models.

A. Problem Formulation

Let $\tau \in [\tau_{min}, \tau_{max}]$, $T_{prop} \in [T_{prop,min}, T_{prop,max}]$, $T_\lambda \in [T_{\lambda,min}, T_{\lambda,max}]$, and $K_{eq} \in [K_{eq,min}, K_{eq,max}]$, with $D = \tau \times T_{prop} \times T_\lambda \times K_{eq}$ denoting the domain of the unknown parameters characterizing a linear system $G(s, d_i)$, (The Laplace variable s will be dropped for brevity) where $d_i \in D$. Let us define the features space $S \subset \mathbb{R}^{n\times m}$, where n denotes the number of elements in each feature vector, and m denotes the number of feature vectors. Then the map $(d_i, \zeta) \mapsto s_i = M(d_i, \zeta)$, where $s_i \in S$ and ζ denotes tunable data generation parameters, defines a data generation function that acts on the process $G(d_i)$. Our goal now is to find an inverse map, $s_i \mapsto d_i = M^{-1}(s_i)$, to infer the unknown process parameters from experimentally generated data s_i.

B. Data Generation Through MRFT

The data generation function M we use is the MRFT given by [35]:

$$u_M(t) = \begin{cases} h & : e(t) \geq b_1 \ \vee \ (e(t) > -b_2 \ \wedge \ u_M(t-) = \ \ h) \\ -h & : e(t) \leq -b_2 \ \vee \ (e(t) < b_1 \ \wedge \ u_M(t-) = -h) \end{cases} \tag{42}$$

where $b_1 = -\beta e_{min}$ and $b_2 = \beta e_{max}$, and $u_M(-t)$, e_{max}, and e_{min} are the previous command, maximum error, and minimum error, respectively. MRFT parameter β is a tunable data generation parameter.

The use of MRFT for data generation has multiple advantages over other methods. Firstly, it is a closed-loop method making the system stable and robust to external disturbances during the data generation phase (refer to [30] for a proof of periodic stability). Secondly, it exploits the gain scale property of linear systems, thus eliminating the need for including K_{eq} in the unknown parameters' domain D, which greatly reduces the numerical complexity of the problem. We use the error input to MRFT $e(t)$ and the MRFT output $u_M(t)$ as data vectors for training; hence $S \in \mathbb{R}^{n\times2}$. The parameter n defines the length of the data vector, and it is selected to fit the largest steady-state oscillation period from all members of $\breve{D}$. A suitable value of the MRFT parameter $\beta \in [-1, 1]$ is found through the process of the design of optimal non-parametric

TABLE II
RANGE OF MODEL PARAMETERS, OPTIMAL VALUE OF MRFT PARAMETER β^*, AND THE TEST SET REPORTED MAXIMUM AND AVERAGE VALUES OF THE RELATIVE SENSITIVITY FUNCTION SHOWN IN EQ. (44) FOR EVERY LINEARIZED DYNAMIC MODEL USED. NOTE THAT β^* VALUE FOR THE UNDERACTUATED LATERAL LOOPS DEPENDS ON ATTITUDE IDENTIFICATION, AND IT IS USUALLY CLOSE TO ZERO

Dynamics	Range of model parameters	β^*	Average J	Maximum J
Attitude	$T_{prop_{b_x}}, T_{prop_{b_y}} \in [0.015, 0.3];\ T_{\gamma_x}, T_{\gamma_y} \in [0.2, 2];\ \tau_{b_x}, \tau_{b_y} \in [0.0005, 0.15]$	-0.73	0.53%	3.51%
Altitude	$T_{prop_z} \in [0.015, 0.3];\ T_{\lambda_z} \in [0.2, 2];\ \tau_z \in [0.0005, 0.15]$	-0.71	0.3%	5.03%
Lateral	$T_{\lambda_x}, T_{\lambda_y} \in [0.2, 6];\ \tau_x, \tau_y \in [0.0005, 0.15]$	Varies	-0.19%	4.91%

tuning rules [35], resulting in the optimal data generation parameter β^*. For simplicity, we refer to $M(d_i, \beta^*)$ by $M(d_i)$.

A summary of the DNN-MRFT tunable parameters used in this work for attitude, altitude and lateral loops is given in Table II.

C. Process Classification Based on DNN

We use a DNN to approximate the mapping from MRFT generated data to the unknown process dynamics. To allow real-time computation of the controller parameters, and to speed up the DNN training process, we approximate the map M^{-1} over the set $\check{D}$ to obtain $\check{M}^{-1} : \check{S} \to \check{D}$. This changes the DNN task from regression over M^{-1} to classification over $\check{M}^{-1}$. The optimal controllers can be found offline for every output class in $\check{D}$, and arranged in the form of a look-up table to be used in-flight. To discretize the process parameters domain we use the following discretization function:

$$\Delta(D, J^*) = \{\check{D} : J(\check{d}_i, \check{d}_j) < J^* \wedge \{\check{d}_i, \check{d}_j\} \in \check{D}\} \quad (43)$$

where $J(d_i, d_j)$ is defined as the relative sensitivity function [36]:

$$J(d_i, d_j) = \frac{Q(C(d_j), G(d_i)) - Q(C(d_i), G(d_i))}{Q(C(d_i), G(d_i))} \times 100\% \quad (44)$$

where $C(d)$ is a controller that is optimal for the process with model parameter d, and $Q(C, G)$ is the Integral Square Error (ISE) performance index of applying a controller C on process G. The constant, J^* in Eq. (43), defines a controller performance specification that prevents under-discretization such that $\forall d_i \in D\ \exists\ \check{d}_i \in \check{D} : J(\check{d}_i, d_i) < J^*$. This condition, however, does not prevent over-discretization of the parameters space. We use $J^* = 10\%$ in this work which was found to provide satisfactory performance in [21]. Note that when $i = j$ in Eq. (44), the relative sensitivity evaluates to $J(d_i, d_i) = 0$.

Due to over-discretization, the addition of noise from sensors, and nonlinearities, the map $\check{M}$ is surjective, i.e. it is not invertible. This non-invertibility condition may lead to instability of the closed loop controllers due to the inevitable misclassification. Thus, we explicitly include the relative sensitivity measure $J(\check{d}_i, \check{d}_T)$ defined in Eq. (44) to penalize misclassifications. The parameters $\check{d}_T$ denote the labeled output process parameters, as the ground truth is available in simulation where the DNN training is performed. The Softmax function used in training is modified to be as follows:

$$p_i = \frac{e^{\kappa_{iT} \cdot a_i}}{\sum_{j=1}^{N} e^{\kappa_{jT} \cdot a_j}} \quad (45)$$

where p_i is the Softmax output probability, a_i is the input logit, and $\kappa_{iT} = 1 + J(\check{d}_i, \check{d}_T)$ is a biased function of the relative sensitivity. As a loss function for training the DNN, the modified formulation in Eq. (45) is utilized alongside the cross-entropy function $L = -\sum_{i=1}^{N} y_i \log(p_i)$, where y is a one-hot encoded vector that indicates the ground truth class T. The partial derivative of L with respect to the logit a_i is computed as:

$$\frac{\partial L}{\partial a_i} = \kappa_{iT} \times (p_i - y_i) \quad (46)$$

Such modification penalizes misclassifications in proportion to the degradation of the closed-loop performance compared to the optimal. A full derivation of the modified Softmax function can be found in [21].

The training set $\mathcal{T} \subset \check{S} \times \check{D}$ and the validation set $\mathcal{V} \subset \check{S} \times \check{D}$ are both used for DNN training phase, and the testing set $\mathcal{U} \subset D \setminus \check{D}$, used for benchmarking, is sampled from outside $\check{D}$ to avoid over-fitting. To improve robustness qualities of the DNN approximation of $\check{M}^{-1}$, we add a randomly sampled constant bias to MRFT, and a Gaussian noise to the training, validation, and testing samples.

The application of DNN-MRFT approach to the underactuated lateral loops (i.e. outer loops) is dependent on the attitude loops tuning and identification results, and hence the outer loop map is defined as $(\check{d}_{o,i}, \check{d}_i) \mapsto \check{s}_{o,i} = \check{M}_o(\check{d}_{o,i}, \check{d}_i)$ with $\check{d}_{o,i} \in \check{D}_o$, and $\check{d}_i \in \check{D}$. Hence, identification of the attitude loops needs to be carried first to estimate $\check{d}_i$, and then $\check{d}_{o,i}$ can be estimated. A simpler inverse map, $\check{M}_{o,d_i}(\check{d}_{o,i}$, can be estimated which keeps the DNN structure the same for all control loops at the expense of having $|\check{D}|$ outer DNNs, i.e. one DNN for every $\check{d}_i \in \check{D}$. The process of cascading DNN-MRFT is detailed in our previous work [30]. As a result, the set of DNNs we used are as follows: first, a DNN to for altitude with 208 output classes (we have widened the range of time delay compared to attitude to accommodate for network delays), a DNN for attitude loops with 48 output classes, and a set of 48 DNNs for lateral loops with an average of 15 output classes for each. ADAM algorithm was used for training, which takes a few minutes per DNN. The DNN inference time takes a few milliseconds in modern embedded processors, which enables the real-time performance.

The DNN architecture used for all inverse map approximations is the same, as it showed to provide the best classification accuracy while doing a sweep of the hyperparameters of different DNN architectures. The used DNN consists of multiple fully connected layers, with ReLU activation functions

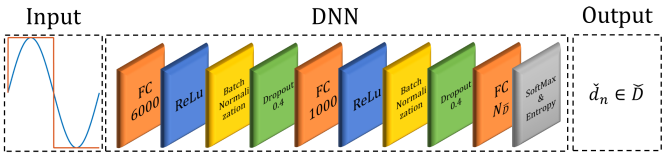


Fig. 5. Used deep neural network architecture. The inputs are the observed variable and the MRFT controller action, while the output is the identified class $\check{d}_n$. The last fully connected layer has a size of the number of discretized processes in $\check{D}$.

in between. Drop-out and batch normalization layers were added to the outputs of the activation functions to reduce over-fitting. The drop-out rate used was 40%. The modified Softmax function described earlier was used in the last layer. The DNN architecture used can be seen in Fig. 5. The testing set results are summarized in Table II.

D. DNN-MRFT Application Considerations for RDEKF and TDEKF

It is important to ensure that the self-excited oscillations do not excite the nonlinearities of the physical system. For example, consider the nonlinearity of the proposed TDEKF process presented in Fig. 4. Keeping $\boldsymbol{w_z}$ aligned with $\boldsymbol{b_z}$ during identification ensures negligible contribution of the rotation matrix nonlinearity, i.e. ${}^{\mathcal{W}}_{\mathcal{B}}R \approx I_3$. In consequence, the identification of the drag parameters T_{λ_x} and T_{λ_y}, which are defined for movements along $\boldsymbol{b_x}$ and $\boldsymbol{b_y}$ respectively, requires the self-excited oscillations to induce lateral movements while keeping $\boldsymbol{w_z}$ aligned with $\boldsymbol{b_z}$. Obviously, this is not possible due to the under-actuated nature of multirotor platforms, and the thrust vector must change its orientation to generate accelerations along $\boldsymbol{w_x}$ or $\boldsymbol{w_y}$. Consequently, during the DNN-MRFT phase, the UA will change its orientation slightly to produce oscillations during lateral processes identifications. For the purpose of this work, we assume that λ_z has no effect during that phase and the side projected area of the UA remains the same during MRFT; note that this is possible since ${}^{\mathcal{B}}v_i \gg {}^{\mathcal{B}}v_z$, and ${}^{\mathcal{B}}v_i \approx {}^{\mathcal{W}}v_i$ for $i \in \{x, y\}$.

The proposed approach can be summarized in Fig. 1. The figure depicts the UA physical system, the sensors measuring its state, the actuation system, and more importantly, the proposed estimation scheme. In our approach, we assume that the UA is in hover state, and all the switches sw are in position $Oper.$. The switches will then move to position $Ident.$ consecutively. This will trigger the DNN-MRFT algorithm as described, causing oscillations and identifying the system. This phase lasts just a few seconds for each loop, and guarantees stability of the loop under test [30]. After all loops have been identified, and the two estimators' dynamic parameters have been updated, the switch is flipped to position $Oper.$. In this switch position, the estimator will collect the control action and predict the states of the UA, while providing corrections as new measurements arrive.

V. Experimental Results

In this section, we provide experimental data that showcases the performance of the proposed estimator. The testing setup consisted of the UA, a MoCap system, and a workstation. The UA used was the Quanser QDrone platform (seen in Fig. 6) with an on-board Intel Atom x7-Z8750 processor. All the control and estimation algorithms were executed in real-time on-board of the UA. Thus, the whole estimation scheme, including TDEKF and RDEKF, require less than 1 ms to be evaluated in the operation phase of the QDrone flight. In the identification phase, the identification DNNs are loaded on-board and require a few milliseconds for prediction.

Fig. 6. The QDrone platform used in the experiments.

On-board also is a BMI160 IMU, and an Intel AC 8620 Wifi module. The IMU's gyroscope has noise root mean square (RMS) of 0.07 $^\circ/s$, a noise spectral density of 0.007 $^\circ/s/\sqrt{Hz}$, and a non-linearity of 0.1 $\%FS$. The accelerometer has an RMS noise of 1.8 mg, a noise spectral density of 300 $\mu g/\sqrt{Hz}$, and a non-linearity of 0.5 $\%FS$. For MoCap system positioning, we used the Optitrack Prime 17 cameras which transmit images to the workstation that is equipped with a software to resolve the pose of QDrone, and then streams this pose to QDrone in real-time over WiFi. The workstation also provided the Graphical User Interface (GUI) needed for interacting with the UA. The codes were developed in MATLAB Simulink environment, and a C++ code was generated using QUARC from Quanser. The full estimation scheme for QDrone is released as an open-source software, and can be found in the detailed workbook in [23].

This section is split into four parts. First, we demonstrate the use of DNN-MRFT to identify the parameters of the SISO models presented in Section II-B. Then we examine the proposed estimation scheme and the identified model by studying the effect of down-sampling position measurements on trajectory tracking of a figure-eight maneuver. To highlight the importance of providing acceleration as an estimate, a ramp test is carried out to show how incorporating filtered accelerations can enhance controller tracking. Finally, we showcase another advantage of our estimation scheme, in the form of controller action reduction, which reduces noise in operation, and increases the flight time of the battery. The aforementioned tests assumes a calibrated accelerometer, i.e. ${}^{\mathcal{S}}_{\mathcal{B}}R = I_3$ with proper scaling adjustment of the accelerometer gains. The calibration was carried out in accordance with the method outlined in [33]. A video recording of all the experiments in this section is available in [22].

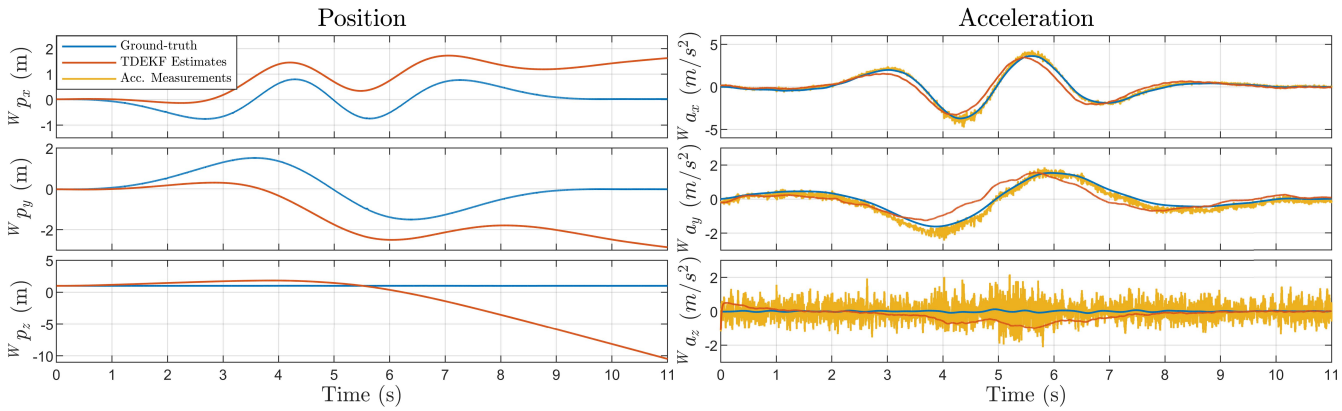


Fig. 7. Proposed translational filter open-loop performance.

TABLE III
IDENTIFIED QDRONE PARAMETERS USED IN THE RDEKF AND TDEKF PREDICTION MODELS. THE ACCURACY OF DNN-MRFT BASED IDENTIFICATION WAS VALIDATED IN OUR PREVIOUS WORK [21] AND [30]

Model Parameter	Estimated Value
τ_{b_x}	0.0009 s
$K_{eq_{b_x}}$	72.6454
$T_{prop_{b_x}}$	0.064 s
T_{γ_x}	0.2494 s
τ_{b_y}	0.0009 s
$K_{eq_{b_y}}$	75.846
$T_{prop_{b_y}}$	0.064 s
T_{γ_y}	0.2494 s
τ_z	0.053 s
K_{eq_z}	0.2949
T_{prop}	0.0177 s
T_{λ_x}	1.1629 s
T_{λ_y}	1.1629 s
T_{λ_z}	0.5793 s

A. Parameters Identification Using the DNN-MRFT and Prediction Performance

We first utilize the DNN-MRFT framework to identify the UA model parameters, as described in Section II-B. At hover, we successively trigger the identification phase, as shown in Fig. 1. The identified parameters obtained are shown in Table III. The attitude loops were estimated to have similar dynamics, as the drone is symmetrical in design. The slight difference in process gain can be attributed to a slight weight distribution imbalance. For the altitude loop, the estimated process had considerably higher delay compared to the attitude, this is expected as a MoCap system have larger delays compared to the on-board inertial measurements. Finally, the lateral loops test was carried out with a small MRFT relay height; as to reduce the attitude angles during the test. The maximum angular amplitudes recorded during the MRFT phase for the lateral loops b_x and b_y were 0.1042 rad, and 0.0972 rad, respectively. Thus, it was assumed that the damping constants T_{λ_x} and T_{λ_y} were due to lateral motion only. The identified model parameters can be found in Table III.

We use two metrics for the quantitative evaluation of both estimators and controllers performance. The first quantitative metric used is the Root Mean Square Error (RMSE) in which we compute the ℓ_2 norm of the difference between the filter estimates and the ground truth. The RMSE is calculated using the following equation:

$$RMSE = \frac{\sum_{i=1}^{N} ||\mathbf{x_i^f} - \mathbf{x_i^g}||_2}{N} \tag{47}$$

where N is the number of the measurement samples in the trajectory. When assessing controller performance, the vector $\mathbf{x_i^f}$ is replaced with the vector of reference states $\mathbf{x_i^r}$. In the second quantitative metric, we evaluate the Mean Contouring Errors (MCE) as a measure of control performance. The following equation defines the MCE:

$$MCE = \frac{\sum_{i=1}^{N} \min\left(||\mathbf{x_i^r} - \mathbf{x_i^g}||_2\right)}{N} \tag{48}$$

For all results, the ground truth is obtained from post-processing the MoCap measurements. Post-processing improves the accuracy of the ground truth due to non-causality.

Before examining the closed-loop performance of the filters, it would be beneficial to see how well the identified model can estimate the states when running in open-loop. The open-loop performance can be meaningfully investigated for translational dynamics, as in some practical scenarios position measurements might be absent. Specifically, we test the TDEKF estimator performance by neither providing position nor acceleration measurements. Fig 7 shows the estimated position, and acceleration, using the TDEKF model shown in Fig. 4 when performing a figure-eight trajectory. The estimated accelerations follow the ground truth trend, which indicates that the predicted model is close to the actual system. However, due to sudden drop in battery levels and nonlinear asymmetries, we can see that during large accelerations along $\boldsymbol{w_x}$ and $\boldsymbol{w_y}$ the estimate along $\boldsymbol{w_z}$ diverges significantly from the ground truth. These differences in accelerations would accumulate and cause a drift in velocity and position.

B. Effect of Position Measurement Update Rate on Filter Performance

To assess the performance of the proposed filtering scheme, along with the identified model, we perform two figure-eight maneuvers with two different update rates of the position

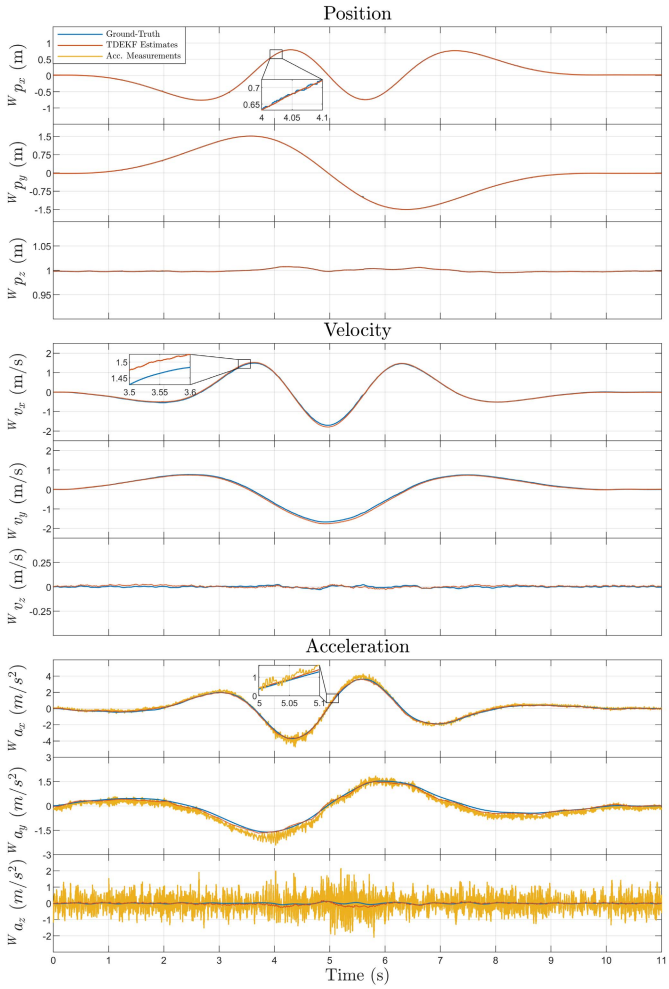


Fig. 8. TDEKF estimates during a figure-eight trajectory with 300 Hz position measurements update.

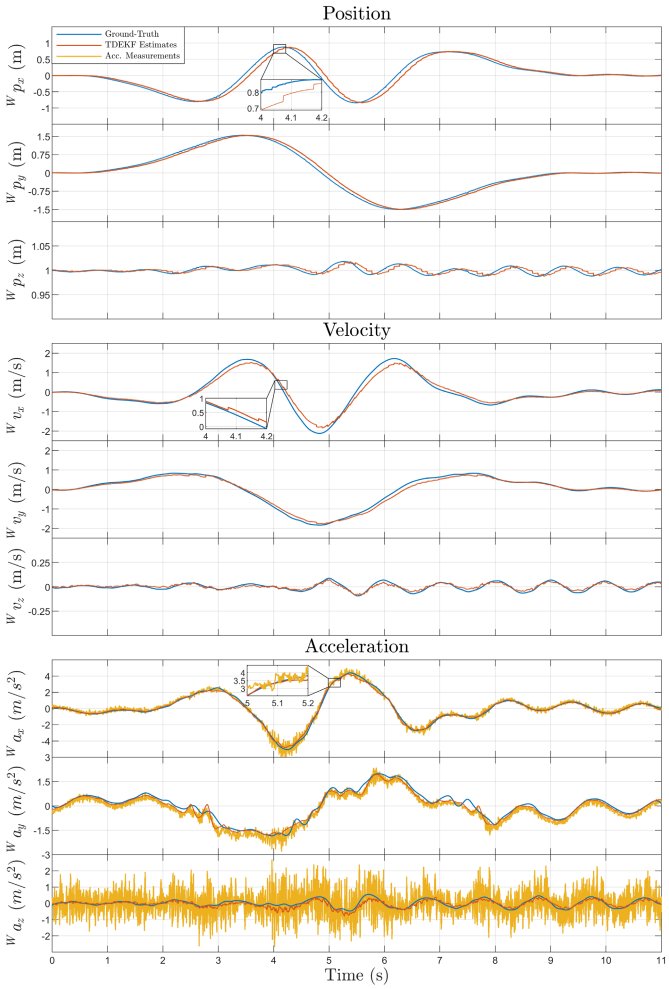


Fig. 9. TDEKF estimates during a figure-eight trajectory with 10 Hz position measurements update.

measurements. The figure-eight measures 3 by 2 meters and takes 10 seconds to execute. In the first figure-eight maneuver, the position measurement rate is 300Hz, we then repeat the previous test with a down sampled position measurement of 10Hz. We chose the 10Hz rate as most commercially available RTK or UWB positioning systems provide position measurements at least at 10Hz. RTK and UWB are the popular choices for UAs operated outdoors, where a MoCap system is not always feasible. For both cases we run TDEKF and RDEKF at 1kHz since our control loop, and hence prediction, runs at 1kHz. Note that as the IMU measurements and the prediction algorithms are running at 1kHz, down sampling the position measurements will only affect the TDEKF performance, hence such comparative assessment does not hold for the RDEKF.

When providing position measurement updates at 10Hz, we observed that the velocity and position estimates lag behind the ground truth, possibly due to the added delay introduced by down sampling. However, we can see that it still provides smooth and stable estimates of position and velocity that are suitable for feedback control (refer to Fig. 9). Due to this lag, the estimates RMSE figures presented in Table IV show an increase in the error figures for the 10Hz case, but these errors

TABLE IV
TDEKF AND RDEKF RMSE FIGURES MEASURING DEVIATIONS BETWEEN THE FILTERS' ESTIMATES AND THE GROUND TRUTH FOR DIFFERENT POSITION MEASUREMENT UPDATE RATES

State		RMSE	
		300 Hz	10 Hz
Position	${}^{\mathcal{W}}p_x$	0.0027 m	0.0903 m
	${}^{\mathcal{W}}p_y$	0.0029 m	0.0764 m
	${}^{\mathcal{W}}p_z$	0.0001 m	0.0044 m
Velocity	${}^{\mathcal{W}}v_x$	0.0297 m/s	0.1573 m/s
	${}^{\mathcal{W}}v_y$	0.0475 m/s	0.0919 m/s
	${}^{\mathcal{W}}v_z$	0.0114 m/s	0.0147 m/s
Acceleration	${}^{\mathcal{W}}a_x$	0.0809 m/s^2	0.1542 m/s^2
	${}^{\mathcal{W}}a_y$	0.0812 m/s^2	0.1769 m/s^2
	${}^{\mathcal{W}}a_z$	0.0638 m/s^2	0.1286 m/s^2
Angle	θ	0.0088 rad	
	ϕ	0.0060 rad	
Rotational Speed	${}^{\mathcal{B}}\omega_x$	0.0296 rad/s	
	${}^{\mathcal{B}}\omega_y$	0.0621 rad/s	

remain small for the practical interest. The increase in the error figures was smallest for the acceleration, then the velocity, and was largest for the position. Such distribution of the increase of the error can be changed by the tuning of process noise Q matrix. We found that with the proposed filter structure,

TABLE V
FIGURE-EIGHT TRACKING RMSE & MCE MEASURING THE DEVIATION BETWEEN THE GROUND TRUTH AND THE REFERENCE TRAJECTORY WHEN DIFFERENT POSITION MEASUREMENT UPDATE RATES ARE USED

State		RMSE	
		300 Hz	10 Hz
Position	${}^{\mathcal{W}}p_x$	0.0204 m	0.1197 m
	${}^{\mathcal{W}}p_y$	0.0127 m	0.0852 m
	${}^{\mathcal{W}}p_z$	0.0030 m	0.0069 m
State		MCE	
		300 Hz	10 Hz
Position ${}^{\mathcal{W}}\boldsymbol{p}$		0.0097 m	0.0273 m

the RMSE results slightly change for a wide range of Q matrix tuning (i.e. filters performance is almost insensitive to changes in Q tuning), which makes the implementation of the TDEKF and RDEKF straightforward and easily applicable to a wide range of UA designs. In tracking, MCE figures observed less increase compared to RMSE when moving from 300 Hz to 10 Hz, as seen in Table V. This implies that the 10Hz position estimates remain close to the ground truth, but are lagged. These results are promising for trajectory tracking applications where only low update rate position sensors are available, and lag in trajectory execution is tolerated.

Table VI shows a comparison between our achieved RMSE and those from [16]. The authors in [16] carried out an experiment using a Global Navigation Satellite RTK System (RTK-GNSS) following a rectangular path to assess their dynamics based filter performance. The position RMSE's reported were 0.3109 m, 0.5895 m, and 1.3527 m, for ${}^{\mathcal{W}}p_x$, ${}^{\mathcal{W}}p_y$, and ${}^{\mathcal{W}}p_z$ respectively. Velocity estimates were also evaluated, and the filter achieved RMSE's of 0.0803 m/s, 0.0757 m/s, and 0.0838 m/s for ${}^{\mathcal{W}}v_x$, ${}^{\mathcal{W}}v_y$, and ${}^{\mathcal{W}}v_z$ respectively. From Table VI we can see that our proposed approach achieved significantly better position estimates, albeit using a MoCap. MoCap has higher position accuracy than RTK-GNSS, however both report high-precision measurements, and both were running at the same sampling frequency of 10 Hz. The velocity estimates reported by [16] slightly out-preformed ours due to the fact that the trajectory flown in [16] is simpler than ours. It is also worth noting that work in [16] performed extensive offline identification, and had full knowledge of the UA physical parameters beforehand.

The RDEKF angular velocity estimates are provided for the 300Hz case in Fig. 10, with their respective RMSE shown in Table IV. The oscillations with a frequency close to 7Hz are due to amplified low frequency body vibrations. The filter successfully rejects all higher frequencies compared to the raw gyroscope measurements, without lagging estimates corresponding to physical movements.

C. Assessment of Acceleration Estimates

Due to the novelty of including accelerations as a state, we can not compare with other dynamics based filters from literature. Authors in [20] did however propose a regression based notch filter for removing the noise from accelerometer measurements in UAs. The performance of the filter

TABLE VI
COMPARISON BETWEEN OUR PROPOSED APPROACH AND 3D-DMAN-UKF WITH GNSS FROM [16]

State		RMSE	
		TDEKF with 10 Hz MoCap	3D-DMAN-UKF with GNSS [16]
Position	${}^{\mathcal{W}}p_x$	0.0903 m	0.3109 m
	${}^{\mathcal{W}}p_y$	0.0764 m	0.5895 m
	${}^{\mathcal{W}}p_z$	0.0044 m	1.3527 m
Velocity	${}^{\mathcal{W}}v_x$	0.1573 m/s	0.0803 m/s
	${}^{\mathcal{W}}v_y$	0.0919 m/s	0.0757 m/s
	${}^{\mathcal{W}}v_z$	0.0147 m/s	0.0838 m/s

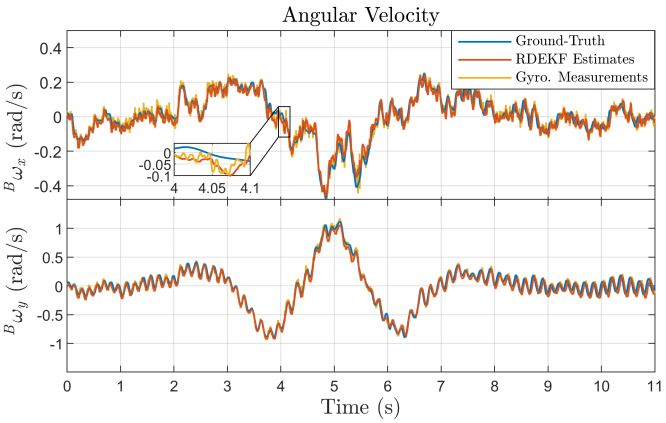


Fig. 10. Proposed RDEKF angular velocity estimates. The high-frequency noises in the sensor measurements have been suppressed.

was not analyzed quantitatively, but the authors claimed that the filter removed most of the noises, without introducing a lag or attenuation in the signal. Similarly, our proposed filter did not introduce lag in estimating the acceleration, even during the 10 Hz position measurements. The advantage of our approach is that the filters are synthesized with complete autonomy in-flight, whereas the filter presented in [20] requires tuning the filter parameters for every UA used.

Although providing a control strategy that incorporates acceleration feedback is not in the scope of this article, a test that shows the advantages of using filtered acceleration in control is necessary to demonstrate the benefits of the proposed filtering scheme. In this test, we provide a ramp reference on ${}^{\mathcal{W}}p_z$ and compare the performance of a Proportional Derivative (PD) controller, to that of a Proportional Double Derivative (PDD) controller. The optimal PD and PDD values were found using the DNN-MRFT model identification and controller tuning framework [28], [30]. We used the filter estimates to track a ramp position reference with a slope of 0.5 m/s for 2 seconds, thus moving the UA a total of 1 m. As seen in Fig. 11 using a PDD controller provided faster response than a PD controller, as the RMSE dropped from 0.0791 m to 0.0449 m, which constitutes 43% increase in control performance.

D. Reduction of Controller Action

An additional advantage of the proposed estimator is the reduction in controller action (i.e. changes in the control signal). Fig. 12 shows the thrust commands produced by the controller when operated with and without the estimators, for

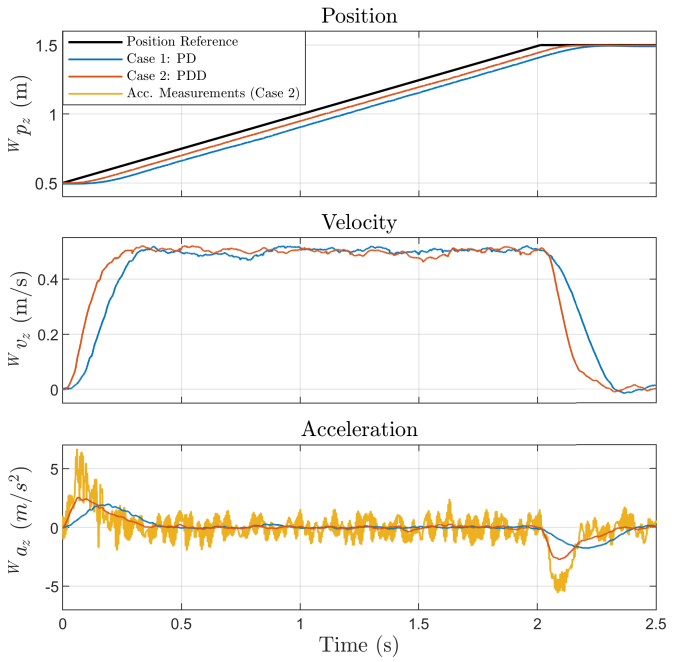


Fig. 11. Ramp tracking performance of PD and PDD controllers using the proposed TDEKF estimates.

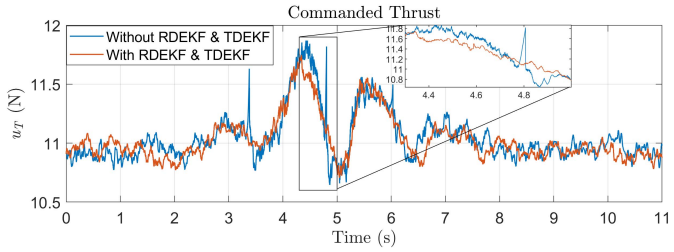


Fig. 12. Controller command reduction when using the proposed estimation scheme.

the figure-eight trajectory described earlier. The estimators clearly smoothed the high rate MoCap position measurements, which resulted in a notable reduction of the controller energy, leading to a smoother flight. It can be noticed that the high-frequency variations in u_T reduced when using the filter. The reduction in controller action was around 6.6%, increasing signal-to-noise ratio (SNR) from 15.73 dB to 20.96 dB, assuming all high frequencies were noises.

VI. Conclusion

In this work, we presented a decoupled estimation scheme that utilized dynamics-based prediction models, and multi-sensor data fusion. The two decoupled estimators increased the states' estimation accuracy, up-sampled states measured at low rate, and to provided estimates for higher order states. Our approach for estimator design consisted of two phases, the identification phase and the operation phase. Our approach has the advantages of being implemented in real-time on-board the UA without any prior knowledge of UA parameters, and not requiring additional sensors such as motor speed encoders. The estimation scheme was validated experimentally, providing smooth and accurate estimates with low-rate position sensors, thus reducing the controller energy. Finally, our approach also provided smooth acceleration estimates, a feature that was not found in previous dynamics based filters for UAs. Having knowledge about the inertial acceleration can increase control performance, as was demonstrated in the ramp tracking experiment in this paper.

The current form of DNN-MRFT performs identification using input-output measurements, which does not guarantee the accuracy of the input to state models. Despite this fact, the estimation scheme performed well, which motivates further theoretical research on the relationship between input-output models and input to state models in the future. Also, the insensitivity of the proposed estimation scheme to the Q matrix tuning offered practical benefits, but requires further in-depth theoretical analysis, which we plan to investigate in our future work.

Appendix
Observability Analysis of the Proposed Filter

The analysis of the observability of the proposed estimator's states is essential to understand if, given the measurements and inputs, the estimator can observe the underlying states.

To study the observability of the nonlinear system equations (18) and (19), we will study the rank of the observability matrix $\mathcal{O}(t)$ [37] of the corresponding LTV system presented in equations (20) and (21). The observability matrix of a generic LTV system can be computed as follows [38]:

$$\mathcal{O}(t) = \begin{bmatrix} \boldsymbol{C}(t) \\ \boldsymbol{C}(t)\boldsymbol{A}(t) \\ \boldsymbol{C}(t)\boldsymbol{A}(t)^2 \\ \vdots \\ \boldsymbol{C}(t)\boldsymbol{A}(t)^{k-1} \end{bmatrix} \tag{49}$$

where k is the dimension of the corresponding state vector. Following [37], a system is said to be locally observable, if the $\text{rank}(\mathcal{O}(t)) = k$, for all t.

In the case where a system is not fully observable, it is important to understand if the system is detectable[1] (i.e. unobserved states are stable) [38]. To check the detectability of the system, it is required first to isolate unobservable states, by performing a canonical decomposition of the state-space equations. To decompose the system, let us assume the existence of a Lyapunov transformation, characterized by the transformation matrix $P(t)$ as follows [38]:

$$\begin{bmatrix} A_o(t) & 0 \\ A_{21}(t) & A_{no}(t) \end{bmatrix} = [P(t)A(t) + \dot{P}(t)]P^{-1}(t)$$

where the index $[\,]_o$ refers to the observable subsystem, while the index $[\,]_{no}$ refers to the unobservable subsystem. For $P(t)$ to qualify as a Lyapunov transformation, $P(t)$ needs to be nonsingular, $P(t)$ and $\dot{P}(t)$ to be continuous, and $P(t)$ and $P^{-1}(t)$ to be bounded [38]. For simplicity, let us assume that we can find such a transformation with $\dot{P}(t) = 0$, *i.e.* with a constant matrix P.

[1]Detectability is a dual concept of stabilizability in the controllability analysis of linear systems

Following this transformation, the decomposed system can be written as follows [38]:

$$\begin{bmatrix} \dot{\boldsymbol{z}}_{\boldsymbol{o}}(t) \\ \dot{\boldsymbol{z}}_{\boldsymbol{no}}(t) \end{bmatrix} = \begin{bmatrix} A_o(t) & 0 \\ A_{21}(t) & A_{no}(t) \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{\boldsymbol{o}}(t) \\ \boldsymbol{z}_{\boldsymbol{no}}(t) \end{bmatrix} + \begin{bmatrix} B_o(t) \\ B_{no}(t) \end{bmatrix} \boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \begin{bmatrix} C_o(t) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{\boldsymbol{o}}(t) \\ \boldsymbol{z}_{\boldsymbol{no}}(t) \end{bmatrix} \tag{50}$$

The submatrices $A_o(t)$, $B_o(t)$, and $C_o(t)$ are associated with the observable transformed state vector $\boldsymbol{z}_{\boldsymbol{o}}(t)$. Note that the observable subsystem is indeed observable, with $\text{rank}(\mathcal{O}_o(t)) = \text{rank}(\mathcal{O}(t))$, where $\mathcal{O}_o(t)$ is the observability matrix of the system defined by $A_o(t)$ and $C_o(t)$.

a) TDEKF observability: Let us first look at the TDEKF system. To linearize our system, we construct the Jacobian of the system presented in Eq. (29); the resultant LTV system is shown below:

$$A_T(t) = \begin{bmatrix} 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & {}^{\mathcal{W}}_{\mathcal{B}}R(t) & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & \Lambda^{-1} & -T_{prop}^{-1}\boldsymbol{w}_z & -\chi\boldsymbol{w}_z \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & -T_{prop}^{-1} & -\chi \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times1} & 0_{1\times1} \end{bmatrix} \tag{51}$$

$$B_T(t) = \begin{bmatrix} 0_{4\times1} & 0_{4\times3} \\ 0_{2\times1} & 0_{2\times3} \\ \chi\boldsymbol{w}_z & {}^{\mathcal{W}}_{\mathcal{B}}R^T(t) \\ \chi & 0_{1\times3} \\ 0_{1\times1} & 0_{1\times3} \end{bmatrix} \tag{52}$$

where the above $A_T(t)$ and $B_T(t)$ are evaluated while assuming ${}^{\mathcal{W}}_{\mathcal{B}}R(t)$ to be estimated externally.

For the analysis of the TDEKF observability, let us first look at the case where a_T is measured, and construct the corresponding observation matrix $C_{T_{a_T}}$ as follows:

$$C_{T_{a_T}} = \begin{bmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times1} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & I_1 & 0_{1\times1} \end{bmatrix} \tag{53}$$

It is easy to see that the corresponding observability matrix $\mathcal{O}_{T_{a_T}}(t)$ has $\text{rank}(\mathcal{O}_{T_{a_T}}(t)) = 11$ for all t; as such the TDEKF is fully observable in the case where a_T is measured.

On the other hand, let us look at the case where a_T is not measurable, and construct the corresponding observation matrix C_T as follows:

$$C_T = \begin{bmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & I_3 & 0_{3\times1} & 0_{3\times1} \end{bmatrix} \tag{54}$$

which yields an observability matrix $\mathcal{O}_T(t)$ such that $\text{rank}(\mathcal{O}_T(t)) = 10$ for all t. Clearly, the system becomes unobservable if a_T is not measurable; in this case, let us analyze if the system is detectable.

For the LTV system defined by $A_T(t)$, $B_T(t)$ and C_T, it is possible to find a Lyapunov transformation satisfying the previously stated conditions, such as:

$$z_{o,i} = x_{T,i} \ \forall i \in \{1, \ldots, 9\} \tag{55}$$
$$z_{o,10} = ma_T + nu_{T_{bias}} \tag{56}$$
$$z_{no} = ma_T - nu_{T_{bias}} \tag{57}$$

where $m, n \in \mathbb{R}^+$ are free constants that can be chosen arbitrarily. The above transformation shows that, if a_T is not measurable, the summation of a_t and $u_{T_{bias}}$ is observable, while their difference is not. It is easy to see that the eigenvalue corresponding to the new state z_{no} is zero; as such, the corresponding state is not stable and the TDEKF system is not detectable. As z_{no} is not stable, it can jeopardize the performance of the proposed filter.

From an experimental point of view, we observed that z_{no} integrates any bias in the system. As the proposed filter is tuned online, such biases are small, and the corresponding value of z_{no} remains small when the system runs for few minutes. However, and to avoid unexpected behavior, in the case where a_T is not measurable, it is convenient to assume that $u_{T_{bias}} = 0$, and to incorporate this assumption in the measurement model. It is easy to see that, following this assumption, the corresponding TDEKF filter is fully observable for all t.

b) RDEKF observability: Similar to the TDEKF, and to linearize our RDEKF system, we construct the Jacobian of the system presented in Eq. (28); the resultant LTV system is shown below:

$$A_R(t) = \begin{bmatrix} \mathcal{Q}_q(t) & \mathcal{Q}_\omega(t) & 0_{4\times2} & 0_{4\times2} & 0_{4\times2} \\ 0_{4\times2} & 0_{2\times2} & I_2 & 0_{2\times2} & 0_{2\times2} \\ 0_{4\times2} & 0_{2\times2} & -\bar{\Gamma}^{-1} & -\bar{T}_{prop}^{-1} & -\eta \\ 0_{4\times2} & 0_{2\times2} & 0_{2\times2} & -\bar{T}_{prop}^{-1} & -\eta \\ 0_{4\times2} & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \end{bmatrix}. \tag{58}$$

$$B_R = \begin{bmatrix} 0_{4\times2} \\ 0_{2\times2} \\ \eta \\ \eta \\ 0_{2\times2} \end{bmatrix} \tag{59}$$

where $\mathcal{Q}_q(t) = \frac{\partial\dot{\boldsymbol{q}}(t)}{\partial\boldsymbol{q}(t)}\Big|_{t=t_n}$, $\mathcal{Q}_\omega(t) = \frac{\partial\dot{\boldsymbol{q}}(t)}{\partial{}^{\mathcal{B}}\bar{\boldsymbol{\omega}}(t)}\Big|_{t=t_n}$,

$$\eta = (T_\Gamma^{-1}\bar{T}_{prop}^{-1}\bar{\boldsymbol{K}}_{\boldsymbol{eq}})^T,$$

and $A_R(t)$ is evaluated while assuming ${}^{\mathcal{B}}\omega_z$ to be estimated externally.

Similar to the TDEKF, it is easy to show that the RDEKF is observable if $\bar{\boldsymbol{\alpha}}_t$ is measurable, while the absence of such measurements render the system unobservable. In the case where $\bar{\boldsymbol{\alpha}}_t$ is not measurable, we can find a Lyapunov transformation similar to the TDEKF system, such as:

$$z_{o,i} = x_{R,i} \ \forall i \in \{1, \ldots, 8\} \tag{60}$$
$$z_{o,9-10} = m\bar{\boldsymbol{\alpha}}_t + n\bar{\boldsymbol{u}}_{M_{bias}} \tag{61}$$
$$z_{no,1-2} = m\bar{\boldsymbol{\alpha}}_t - n\bar{\boldsymbol{u}}_{M_{bias}} \tag{62}$$

where $m, n \in \mathbb{R}^{+\,2\times2}$ are diagonal matrices that can be chosen arbitrarily. Similar to the TDEKF, the above transformation shows that if $\bar{\boldsymbol{\alpha}}_t$, summation between $\bar{\boldsymbol{\alpha}}_t$ and $\bar{\boldsymbol{u}}_{M_{bias}}$ is observable, while their difference is not. It is easy to see that the eigenvalues corresponding to $z_{no,1-2}$ are zero; as such, the unobservable states of the RDEKF are not stable, and the corresponding system is not detectable. Similar to the TDEKF, and to avoid unexpected behavior in the online computation of

the RDEKF system, in the case where $\bar{\boldsymbol{\alpha}}_t$ is not measurable, it is convenient to assume $\bar{\boldsymbol{u}}_{M_{bias}} = \mathbf{0}$. Similar to the TDEKF, such an assumption can be incorporated in the measurement model, rendering the ensuing system observable.

Acknowledgment

The authors would like to thank Abulla Ayyad and Abdulaziz Alkayas for their technical assistance with the experiments.

References

[1] K. Gamagedara, T. Lee, and M. Snyder, "Quadrotor state estimation with IMU and delayed real-time kinematic GPS," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2661–2673, Oct. 2021.

[2] H. Deng, Q. Fu, Q. Quan, K. Yang, and K.-Y. Cai, "Indoor multi-camera-based testbed for 3-D tracking and control of UAVs," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3139–3156, Jun. 2020.

[3] W. Nie *et al.*, "UAV detection and localization based on multi-dimensional signal features," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5150–5162, Mar. 2021.

[4] C. Xu, Z. Wang, Y. Wang, Z. Wang, and L. Yu, "Three passive TDOA-AOA receivers-based flying-UAV positioning in extreme environments," *IEEE Sensors J.*, vol. 20, no. 16, pp. 9589–9595, Aug. 2020.

[5] Q. Lu, Y. Zhang, J. Lin, and M. Wu, "Dynamic electromagnetic positioning system for accurate close-range navigation of multirotor UAVs," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4459–4468, Apr. 2020.

[6] W. You, F. Li, L. Liao, and M. Huang, "Data fusion of UWB and IMU based on unscented Kalman filter for indoor localization of quadrotor UAV," *IEEE Access*, vol. 8, pp. 64971–64981, 2020.

[7] T. Mouats, N. Aouf, L. Chermak, and M. A. Richardson, "Thermal stereo odometry for UAVs," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6335–6347, Nov. 2015.

[8] M. k. Al-Sharman, B. J. Emran, M. A. Jaradat, H. Najjaran, R. Al-Husari, and Y. Zweiri, "Precision landing using an adaptive fuzzy multi-sensor data fusion architecture," *Appl. Soft Comput.*, vol. 69, pp. 149–164, Aug. 2018.

[9] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, Jun. 2008.

[10] S. Bonnabel, P. Martin, and P. Rouchon, "Non-linear symmetry-preserving observers on Lie groups," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1709–1713, Jul. 2009.

[11] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 24–34, Jan. 2020.

[12] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1623–1629.

[13] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[14] D. Abeywardena, S. Kodagoda, G. Dissanayake, and R. Munasinghe, "Improved state estimation in quadrotor MAVs: A novel drift-free velocity estimator," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 32–39, Dec. 2013.

[15] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain, "Quadrotors and accelerometers: State estimation with an improved dynamic model," *IEEE Control Syst.*, vol. 34, no. 1, pp. 28–41, Feb. 2014.

[16] N. Y. Ko, I. H. Choi, G. Song, and W. Youn, "Three-dimensional dynamic-model-aided navigation of multirotor unmanned aerial vehicles," *IEEE Access*, vol. 7, pp. 170715–170732, 2019.

[17] R. Munguía, S. Urzua, and A. Grau, "EKF-based parameter identification of multi-rotor unmanned aerial vehicles models," *Sensors*, vol. 19, no. 19, p. 4174, Sep. 2019.

[18] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-based inertia estimation for a quadrotor using Newton–Euler dynamics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3861–3867, Jul. 2020.

[19] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 1203–1218, May 2021.

[20] M. Hamandi, M. Tognon, and A. Franchi, "Direct acceleration feedback control of quadrotor aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5335–5341.

[21] A. Ayyad, M. Chehadeh, M. I. Awad, and Y. Zweiri, "Real-time system identification using deep learning for linear processes with application to unmanned aerial vehicles," *IEEE Access*, vol. 8, pp. 122539–122553, 2020.

[22] M. Wahbah, M. Chehadeh, M. Hamandi, L. Seneviratne, and Y. Zweiri. (2021). *Experimental Tests*. [Online]. Available: https://youtu.be/UszC9iediq0. Youtube.

[23] M. Wahbah, M. Chehadeh, and Y. Zweiri, (2021). *Decoupled Dynamic EKF for UAV*. [Online]. Available: https://github.com/MWahbahCC/DDEKF

[24] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Eng. Pract.*, vol. 18, no. 7, pp. 691–699, Jul. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066110000456

[25] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, (2012). *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2007-6461

[26] J. Svacha, G. Loianno, and V. Kumar, "Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1109–1116, Apr. 2019.

[27] M. S. Chehadeh and I. Boiko, "Design of rules for in-flight non-parametric tuning of PID controllers for unmanned aerial vehicles," *J. Franklin Inst.*, vol. 356, no. 1, pp. 474–491, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003218306604

[28] A. Y. Alkayas, M. Chehadeh, A. Ayyad, and Y. Zweiri, "Systematic online tuning of multirotor UAVs for accurate trajectory tracking under wind disturbances and in-flight dynamics changes," *IEEE Access*, vol. 10, pp. 6798–6813, 2022.

[29] G. Torrente, E. Kaufmann, P. Fohn, and D. Scaramuzza, "Data-driven MPC for quadrotors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3769–3776, Apr. 2021.

[30] A. Ayyad *et al.*, "Multirotors from takeoff to real-time full identification using the modified relay feedback test and deep neural networks," *IEEE Trans. Control Syst. Technol.*, early access, Oct. 6, 2021, doi: 10.1109/TCST.2021.3114265.

[31] S. O. H. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Apr. 2010, pp. 1–32. [Online]. Available: http://www.x-io.co.U.K./open-source-imu-and-ahrs-algorithms/

[32] Q. Quan, *Introduction to Multicopter Design Control*. Cham, Switzerland: Springer, 2017.

[33] A. Vitali, "6-point tumble sensor calibration," STMicroelectronics, Geneva, Switzerland, Tech. Rep. DT0053 Rev. 1, Dec. 2015.

[34] B. Southall, B. F. Buxton, and J. A. Marchant, "Controllability and observability: Tools for Kalman filter design," in *Proc. BMVC*, 1998, pp. 1–10.

[35] I. Boiko, "Non-parametric tuning of PID controllers: A modified relay-feedback-test approach," in *Advances in Industrial Control*. London, U.K.: Springer, 2012.

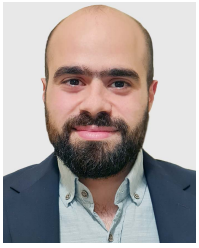
[36] R. Rohrer and M. Sobral, "Sensitivity considerations in optimal system design," *IEEE Trans. Autom. Control*, vol. AC-10, no. 1, pp. 43–48, Jan. 1965.

[37] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York, NY, USA: Academic, 1982.

[38] C.-T. Chen, *State-Space Solutions and Realizations*. New York, NY, USA: Oxford Univ. Press, 2013.

Mohamad Wahbah received the M.Sc. degree in electrical engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2018. He is currently a Researcher with the Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research areas include multisensor fusion, state estimation, and navigation in hazardous environments.

Mohamad Chehadeh (Member, IEEE) received the M.Sc. degree in electrical engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2017. He is currently with the Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interests are mainly focused on identification, perception, and control of complex dynamical systems utilizing the recent advancements in the field of AI.



Mahmoud Hamandi received the M.Eng. degree in mechanical engineering from the American University of Beirut, Beirut, Lebanon, in 2017, and the Ph.D. degree in automation and robotics from the National Institute for Applied Sciences, Toulouse, France, in 2021. He is currently a Postdoctoral Fellow of the Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interests include perception, design, and control of robotic systems, with applications to aerial robots.

Lakmal Seneviratne is a Professor of Mechanical Engineering and the Founding Director of the Centre for Autonomous Robotic Systems (KUCARS), Khalifa University, United Arab Emirates. He has also worked as Associate Provost for Research and Graduate Studies and an Associate VP Research at Khalifa University. Prior to joining Khalifa University, he was a Professor of Mechatronics, the Founding Director of the Centre for Robotics Research, and the Head of the Division of Engineering, King's College London. He is a Professor Emeritus with the King's College London. His main research interests are centred on robotics and automation, with particular emphasis on increasing the autonomy of robotic systems interacting with complex dynamic environments. He has published over 400 peer-reviewed publications on these topics. He is a member of the Mohammed Bin Rashid Academy of Scientists in United Arab Emirates.

Yahya Zweiri (Member, IEEE) received the Ph.D. degree from the King's College London in 2003. He is currently an Associate Professor with the Department of Aerospace Engineering and the Theme Leader of the Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University, United Arab Emirates. He was involved in defense and security research projects in the last 20 years at the Defence Science and Technology Laboratory, King's College London, and the King Abdullah II Design and Development Bureau, Jordan. He has published over 100 refereed journals and conference papers and filed ten patents in USA and U.K., in the unmanned systems field. His central research interests include interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization, and advanced control.