A Machine Learning Approach for Estimating Air Data Parameters of Small Fixed-Wing UAVs Using Distributed Pressure Sensors

Kasper Trolle Borup, Thor Inge Fossen, *Fellow, IEEE*, and Tor Arne Johansen, *Senior Member, IEEE*

***Abstract*—This paper presents a method for estimating the air data parameters for a small fixed-wing, unmanned aerial vehicle (UAV) using an arrangement of low-cost MEMS-based pressure sensors embedded in the surface of the UAV. The pressure measurements are used in a machine learning (ML) model to estimate the angle of attack (AOA), sideslip angle (SSA), and airspeed. Two ML algorithms based on artificial neural networks (NNs) and linear regression (LR) are implemented, tested, and assessed using data collected from wind tunnel experiments and a flight test and the results are compared to a benchmark flight test. Training the ML algorithms using wind tunnel data was found to introduce several potential error sources that need to be addressed in order to provide accurate estimation on the benchmark flight test, whereas training the algorithms using flight data provides lower estimation RMSE values. The performance of the NN structures has been found to slightly outperform the linear regression algorithms in estimation accuracy. Lastly, results from using different sensor configurations and a pseudo Reynolds number are presented in an effort to evaluate the influence of sensor number and placement on the accuracy of the method.**

***Keywords*—Unmanned Aerial Vehicle, UAV, Airspeed, Angle of Attack, Sideslip Angle, Air Data Parameters, Fixed-Wing, Pressure Sensors, Wind Tunnel Experiments, Flight Tests, Machine Learning, Neural Networks, Linear Regression, Pressure Sensors.**

I. Introduction

Knowledge of the wind velocity surrounding a fixed-wing UAV is critical to safe and efficient UAV control and operation. The relative velocity of the UAV with respect to the air contains information from which the angle of attack (AOA), sideslip angle (SSA), and airspeed are directly computable. The AOA, SSA, and airspeed variables are commonly referred to as the air data parameters and the values during flight are directly related to the performance and safety of the unmanned aircraft. For a given UAV wing profile, the AOA will, for example, determine when the wing is under stall conditions where a large increase in separation of flow over the wing leads to a significant drop in lift force. Because of the one-to-one map between air data parameters and relative velocity, these variables will throughout the rest of the paper simply be referred to as air data parameters.

Kasper Trolle Borup, Professor Thor I. Fossen, and Professor Tor A. Johansen are with the Department of Engineering Cybernetics, Centre for Autonomous Marine Operations and Systems, The Norwegian University of Science and Technology, NO-7059 Trondheim, Norway, e-mail:kasper.borup@ntnu.no, thor.fossen@ntnu.no, and tor.arne.johansen@itk.ntnu.no

Larger fixed-wing aircraft are often equipped with air data systems, such as vanes and multi-hole pitot probes, but for smaller UAVs, there are often strict requirements on size, weight, power consumption, and cost. It is therefore highly desirable to have an air data parameter estimation system only utilizing measurements that are obtainable for a UAV through standard sensor suite measurements and/or extra sensors that have low cost and can easily be integrated into the UAV. The solution presented in this document is based on a machine learning (ML) approach using pressure sensors embedded in the surface of the UAV.

Several papers have been published on air data estimation for UAVs using a standard sensor suite consisting of an inertial measurement unit, a GNSS receiver, a heading and attitude reference, and an airspeed sensor. Long and Song [18] used sensor fusion combined with an aerodynamic and kinematic model to estimate airspeed and AOA without the airspeed sensor. Langelaan et al. [14] proposed a direct computation of the wind velocity, rate of change of wind velocity, and wind velocity spatial gradient based on GNSS velocity and acceleration measurements using linearized expressions for the aerodynamic forces and moments. Ramprasadh and Arya [24] obtained AOA and SSA estimates by using a Newton-Rahpson solver on an aerodynamic model with an EKF. Lie and Gebre-Egziabher [17] proposed a cascaded EKF and aircraft model structure for estimating the air data without airspeed measurements. Cho et al. [7] assumed having airspeed measurements scaled by an unknown factor, and used an EKF to estimate AOA, SSA, and the unknown airspeed sensor scaling factor. Johansen et al. [13] proposed a model-free, kinematic approach for estimating wind velocity and airspeed scaling factor and proves global exponential stability of the system under persistence of excitation of the aircraft angular rates. Rhudy et al. [25] presented a nonlinear Kalman filter airspeed estimation method that assumed measurements of the AOA and SSA from wind vanes. Wenz et al. [30] presented an EKF structure approach to estimating air data that exploits a simplified aerodynamic model for lift and drag combined with the Dryden wind model (as described in [9]). By employing a moving horizon estimator, Wenz and Johansen [29] built on the previous result and improved the accuracy.

Using a distributed pressure sensor approach to estimating air data on fixed-wing aircraft has also been investigated. NASA conducted research on flush air data sensing (FADS) systems in response to the problems experienced with protruding Pitot probes. The FADS systems utilize pneumatic pressure

orifices that are flush with the surface in a symmetric circular pattern on the nose of the aircraft that when combined with an aerodynamic sphere pressure model, allows for air data estimation. This approach was demonstrated in wind tunnels for subsonic airspeeds by Larson et al. [15]. Larson et al. extended the FADS system method to transonic airspeeds [16], and Whitmore et al. [31] demonstrated the system in-flight.

Using the pneumatic FADS system in combination with NNs to estimate freestream static and dynamic pressure was proposed and demonstrated by Rohloff et al. [26]. Rohloff et al. [27] proposed an air data sensing system, where NNs were used in combination with an aerodynamic model of the nose of the aircraft to estimate the air data estimates. Quindlen and Langelaan [23] presented a nose FADS system that used NNs to estimate air data for a soaring UAV. The system was trained using wind tunnel data and tested in flight without a ground truth sensor to provide verification. Furthermore, the size of the pneumatic system necessitated removing the electric motor, thereby requiring a less optimal launching procedure. Instead of pneumatic pressure sensors, Callegari et al. [6] presented the idea of combining a maximum likelihood estimator with strips of capacitive pressure sensors applied to the wings of a UAV to estimate the airspeed and AOA. This was demonstrated in simulation, but only with a low level of noise and for small values of AOA. Samy et al. [28] used a matrix of pneumatic pressure orifices placed on the leading edge of the wing in combination with a NN to estimate the air data parameters and tested this in a wind tunnel. The pneumatic system designed by Samy et al. was connected to pressure transducers placed outside of the aircraft and the system in its presented configuration was therefore not usable for flight.

The solution presented in this paper consists of combining non-intrusive low-cost MEMS-based pressure sensors embedded in the surface of the UAV with an LR or NN modeling approach. A strength of the presented solution lies in the flexibility of the sensor placement since there are few geometric constraints. Depending on the UAV, this potentially allows equipping a UAV with an air data parameter estimation system where other solutions are not viable. An example are aircraft with nose propellers, which denies a nose FADS system. Instead, the pressure sensors can be embedded in the wings or fuselage of the aircraft, thereby still allowing air data parameter estimation. The wide range of possible layouts of the solution can be chosen to accommodate the exact needs of the UAV it is designed for. The presented solution is not pneumatic, i.e. does not require tubing to pressure scanners that can be sensitive to mechanical stress. Furthermore, the approach removes the need for mounting a protruding probe that is exposed and susceptible to damage during landing. Another contribution consists of the validation obtained from extensive testing and comparing the results obtained from different ML models and sensor configurations in both wind tunnel and flight experiments.

Section II states the problem and the assumptions behind the method and Section III presents the two ML approaches to estimating the air data parameters from pressure measurements. Section IV contains a description of the experimental setup and Section V presents the obtained results. The presented results are based on both wind tunnel tests and a flight test with a UAV.

II. Main Principles

For a UAV the relative velocity can be expressed in the body coordinate frame as the difference of the ground velocity and the wind velocity:

$$\mathbf{v}_r^b = \mathbf{v}^b - \mathbf{v}_w^b \tag{1}$$

where $\mathbf{v}^b = [u, v, w]^\top$ is the velocity over ground vector of the UAV, $\mathbf{v}_r^b = [u_r, v_r, w_r]^\top$ is the relative velocity vector and $\mathbf{v}_w^b = [u_w, v_w, w_w]^\top$ is the wind velocity vector. One can use pressure sensor measurements to estimate the airspeed V_a, AOA α, and SSA β defined as:

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{2}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{3}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{4}$$

Both the airspeed V_a and AOA α are directly related to the lift and drag forces, and knowing these is valuable in controlling the aircraft. The relationship between the air data parameters and the pressure distribution across a cambered airfoil and a fuselage is not trivial to model. For an airfoil of a given shape at a given AOA, the resultant aerodynamic force, R, is dependent on five different parameters: The airspeed V_a[1], the freestream density ρ_∞, the viscosity of the fluid μ, the size of the body by a reference length c, and the speed of sound ${a_\infty}$[2]. By application of the Buckingham pi Theorem, dimensional analysis allows expressing the aerodynamic force in terms of a dimensionless force coefficient $C_R = R/\frac{1}{2}\rho_\infty V_a^2 c^2$, as a general function of only two variables, the freestream Reynolds number Re$= \rho_\infty V_a c/\mu$ and the Mach number $M = V_a/a_\infty$. For a treatment of dimensional analysis for an airfoil, the reader is directed to Anderson [2]. Furthermore, for Mach numbers below 0.3, the effects of compressibility are negligible and the flow can be considered incompressible. Since most small UAVs operate in Mach numbers well below 0.3, it is assumed that the use of Mach numbers for the dimensional analysis can be neglected. Extending dimensional analysis from two dimensional airfoil to three dimensional aircraft requires the SSA. Hence, dimensional analysis allows for stating the force coefficient C_R for an aircraft of a given shape as a general function of only the Reynolds number, the AOA, and the SSA, where the expression for the Reynolds number also contains the airspeed.

The theoretical basis for the method presented in this paper revolves around the relation from air data to force coefficient and pressure distribution. In general, the pressure distribution and flow over a given 3-dimensional object moving through

[1] Often referred to as the freestream velocity in aerodynamic literature.

[2] Actually the compressibility of the fluid, but that value is representable by the speed of sound.

a fluid is highly nonlinear and very complicated to model. Unless the object has certain favorable geometric properties, finding the pressure and flow around the object requires solving the Navier-Stokes equations numerically. In this paper it is assumed that given a sufficient set of pressure measurements from a set of measurement points distributed on the surface of a UAV, it is possible to inversely map this spatially sampled pressure distribution back to the air data.

The pressure measurements obtained by the sensors have been used differentially with respect to a designated reference sensor, i.e. a selected sensor measurement has been subtracted from all the other sensor measurements. Biases and sensor noise will be addressed later, but for this discussion we assume perfect sensor measurements. Instead of using the pressure measured directly, which is a function of altitude, temperature, and humidity, the differential pressure instead directly relates the measured pressure to the pressure distribution on the aircraft. The price is the reduction of the sensor measurements by one dimension, but should enable - given a proper training data set and a suitable ML structure - to make the ML method robust with respect to changes in ambient conditions. With the differential sensor approach, there is for a n-sensor setup in effect only $n-1$ differential pressure measurements available and throughout the rest of this paper, they will be referred to using their sensor numbers as p_{s_1} to $p_{s_{n-1}}$.

Finally, it is assumed that the pressure differences stemming from hydrostatic differences between sensors are negligible. For example, if the UAV is engaged in a banked turn, one wing will be higher than the other. For small UAVs the pressure difference between the sensors on that wing relative to the sensors on the other wing will be relatively small and therefore not affect the results, i.e. the hydrostatic term in Bernoulli's equation can be neglected due to the low density of air and the small differences in height.

III. Machine Learning Modeling Approach

The two ML approaches used in this paper are LR and NNs. The results obtained using both approaches are presented, compared, and discussed in the following sections. However, the concept used in this paper is not restricted to these ML algorithms and an algorithm such as support vector machines should for example be able to replace the mapping from pressure measurements to air data parameter estimates, although the quality of the results will depend on algorithm, training method, quality of training data, etc.

A. Linear Regression (LR)

LR has been called the "work horse" of ML [21]. Contrary to what the name implies, LR is not limited to modeling linear functions. The algorithm minimizes a least-squares error cost function to map a linear combination of input variables to an output, but these input variables can be chosen to be nonlinear functions of the basis input. Augmenting the basis input with nonlinear functions of the input is known as basis function expansion. The high complexity of the physics relating the air data to the pressure distribution over the aircraft does not allow for a first principles approach to choosing the basis function expansions. In this paper, three generic basis function expansions have been tested, compared, and verified by experiments. Basis function expansions can potentially provide a higher accuracy to the linear regression method, but comes with the cost of a higher computational demand, both when computing the weights, but also during run-time. Another common consideration for model choice is the principle of parsimony, where the model ideally use the fewest possible parameters to adequately represent the input-output relations. The model choice is based on the fundamental trade-off between underfitting and overfitting the data and thereby balancing model bias with model variance [5]. For the LR approach presented, the basis input consists of the differential pressure measurements. Three basis function expansions will be considered, including two polynomial basis functions. Using low-level polynomial basis function expansions is a common modeling approach in LR [4, 21]. The basis input is the vector of differential pressure measurements:

$$[p_{s_1},\ p_{s_2}, \ldots,\ p_{s_{n-1}}]$$

The first basis function expansion uses the first order cross terms between the differential pressure measurements. Using polynomial cross terms is a widely used in LR modeling (see [12], where the cross terms are referred to as interactions):

$$[p_{s_1}p_{s_2},\ p_{s_1}p_{s_3}, \ldots,\ p_{s_2}p_{s_3},\ p_{s_2}p_{s_4}, \ldots,\ p_{s_{n-2}}p_{s_{n-1}}]$$

where the p_{s_i} represents the i'th sensor differential pressure measurement. For a differential n sensor setup, this corresponds to an augmentation of k vector entries, where k is given by the binomial coefficient as:

$$k = \binom{n-1}{2} = \frac{(n-1)!}{2!(n-3)!} \tag{5}$$

The second expansion is a square of the input measurements:

$$[p_{s_1}^2,\ p_{s_2}^2, \ldots,\ p_{s_{n-1}}^2]$$

The third and last expansion is the input measurements cubed:

$$[p_{s_1}^3,\ p_{s_2}^3, \ldots,\ p_{s_{n-1}}^3]$$

There are obviously an infinite amount of possible basis function expansions and there are probably ones that could provide better results than the expansions chosen. However, the chosen basis function expansions are generic choices and should serve as a first approximation for testing the presented concept and could instead be augmented with more tailored functions, if available.

When presenting the LR results, the following naming abbreviations have been used to keep the results concise:

- **B**: Basis input vector.
- **X**: First order cross term expansion.
- **Q**: Quadratic expansion.

- **C**: Cubic expansion.

Following this convention, **BXC** will refer to a the basis input vector expanded by first order cross terms and the cubic terms, where **BQC** will refer to the basis input vector expanded by the quadratic and cubic terms, etc.

For the wind tunnel experiments, the data used for LR has been partitioned into a 85% training set and a 15% test set, and the data has been randomly divided into these two sets. The LR weights are found using the training data and the test set is used in providing a measure of the performance of the method. The influence of the training and test partition sizes were found to be negligible when kept in the ranges of 60/40% to 85/15% and the partition sizes of 85/15% was chosen to allow a large set of training data. The stochastic element from randomly dividing the data into training and test sets will result in different outcomes for every set of linear regression weights calculated. Therefore, whenever numerical results are presented, a mean of ten different training set results have been used to reduce the impact of randomness.

B. Artificial Neural Networks (NNs)

The structure of interconnected neurons in artificial NNs provide a generic method for approximating continuous functions from observational data. The NNs in this paper have been designed using the MATLAB nftool toolbox. The NNs are trained using the Levenberg-Marquardt (LM) backpropagation algorithm (Marquardt [19]), since it is very efficient for networks consisting of a few hundred weights or less [10]. The LM algorithm minimizes a least-squares cost function by blending gradient descent with Gauss-Newton's algorithm, retaining the favorable stability properties of gradient descent and the speed of the Gauss-Newton algorithm. The LM algorithm uses the Gauss-Newton Jakobian-based method of estimating the Hessian, but with an added positively scaled identity matrix

$$\mathbf{H} \approx \mathbf{J}^\top \mathbf{J} + \mu \mathbf{I} \tag{6}$$

where $\mathbf{H}$ is the Hessian, $\mathbf{J}$ is the Jakobian, and μ is a positive scalar called the combination coefficient. The weights are updated using the update rule

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1} \mathbf{J}_k \mathbf{e}_k \tag{7}$$

where k is the training iteration index, $\mathbf{w}$ are the weights, and e is the model output error vector. The combination coefficient can be viewed as a blending factor between gradient descend, if μ is large, and the Gauss-Newton algorithm, if μ is close to zero. The blending factor μ is reduced for each step that reduces the cost function output, and increased for each step that increases the cost function output, thereby allowing the LM algorithm the stability of gradient descent, but with faster convergence.

The NNs have been chosen with the hidden layer using the tan-sigmoid function and the output layer using a linear function [10]. The NNs have been designed with one or two hidden layers and a varying number of hidden neurons. It is known from NN approximation theory that with a sufficient number of hidden neurons, continuous functions can be approximated to arbitrary accuracy. The optimal structure depend on the training data and the chosen structures will be validated and compared through results. The three different air data parameters (airspeed, AOA, and SSA) will be estimated each by a separate NN and the NNs presented will therefore all have a single output neuron. Figure 1 shows the structure of a NN for estimating the airspeed from the measurements of $n-1$ pressure sensors using 10 neurons in a single hidden layer.

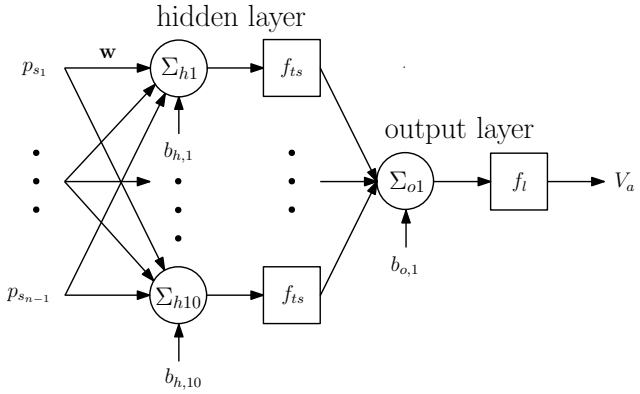


Fig. 1: An illustration of a NN that provides estimates of the airspeed, V_a. The NN is here illustrated with $n-1$ differential pressure measurements as input, a single hidden layer with 10 tan-sigmoid function neurons f_{ts}, and a linear function output layer that outputs the airspeed estimate.

Since all the presented NNs are trained using the same algorithm and all have a single output neuron, the only variables varied will be the number of hidden layers and the number of neurons in each hidden layer. As with the linear regression results, a naming abbreviation has been employed to keep the results concise. $\mathbf{F}_k$ is used to denote the first hidden layer containing k neurons and $\mathbf{S}_l$ denotes the second hidden layer containing l neurons. Hence, $\mathbf{F}_8\mathbf{S}_0$ denotes a NN with a single hidden layer with 8 neurons and $\mathbf{F}_{12}\mathbf{S}_8$ denotes a NN with two hidden layers with respectively 12 and 8 neurons in first and second hidden layer.

For the wind tunnel experiments using NNs, the data has been partitioned into 70% training data, a 15% validation data, and a 15% test data. The data has randomly been divided into these three sets. Other possible data partition sizes, including 60/20/20% and 50/25/25%, were also tested, and the partition sizes were found to have a very small influence when kept in this range. Just as for the linear regression approach, dividing the data randomly results in varying results for each new NN trained and a mean of ten NN results are therefore provided whenever numerical results are provided and both training and test set results are provided.

IV. Experimental Setup

The results in this paper are obtained using a Skywalker X8 Flying Wing UAV. The Skywalker X8 is a consumer grade UAV with a wingspan of 2.12 meters that is usually flown in airspeeds ranging from 15 to 25 m/s. A picture of a Skywalker X8 in flight is shown in Figure 2.

The method presented in this paper utilizes an array of low-cost pressure sensors strategically embedded in the surface of the UAV. The method is not dependent on a specific type of pressure sensor, but the accuracy of the chosen sensor will affect the results and so the sensor choice will be a trade-off between cost, accuracy, weight, size, power consumption, robustness, and ease-of-implementation. For the Skywalker X8 prototype implementation, our objective is a proof of concept, and the BMP280 sensor by Bosch was chosen with the breakout board designed by Adafruit. The Bosch BMP280 is a MEMS-based[3] digital pressure and temperature sensor that offers a decent accuracy and a small footprint for a low cost, which makes it ideal for this application. The BMP280 has an absolute accuracy of ± 100 Pa and a relative accuracy of ± 12 Pa, a temperature range of -40 to 80 °C, and is in the highest resolution capable of logging at 26.3 Hz. The difference in absolute accuracy and relative accuracy stems from a slowly varying bias. A 24 hour test was conducted, where the bias for 16 sensors only varied minimally and it was therefore concluded that the BMP280 biases can be assumed as constant throughout a UAV flight and as a counter measure, these biases have been removed from the tests. This was done in a pre-flight calibration by finding the biases from data where there was no wind velocity on the Skywalker X8 as the difference between the mean measurement of a specific sensor and the mean measurement of all the sensors. This calibration method has been employed for all the data used in this paper. Hence, the BMP280, when used in a differential setup, is assumed to have an accuracy of ± 12 Pa. Throughout the wind tunnel test the logging frequency was 7 Hz and for the flights it was increased to 20 Hz. The BMP280 comes with built-in low pass filters, but these filters have not been employed, since an objective of the presented method is to be able to estimate air data parameters in high dynamics.

A total of 16 BMP280 sensors have been embedded in the surface of the Skywalker X8 used in all the tests. The sensors are connected to an Arduino Mega 2560 through Serial Peripheral Interface (SPI). The wire-based communication usable for a MEMS-based sensor relieves the conventional setup of rubber tubes connected to a pressure scanner, which will simplify the implementation. The Arduino Mega 2560 collects the pressure and temperature readings and outputs the readings through serial to USB to either a laptop (the setup used the wind tunnel tests) or an ODROID-XU4 that logs the data (the setup employed in the flight tests). The 16 sensors have been distributed in a symmetric pattern with 5 sensors on a cross section of each wing that is parallel to the (longitudinal) xy-plane. The remaining 6 sensors have been embedded in the nose of the Skywalker X8 in a pattern roughly resembling a circle. Figure 3 shows the implementation of the pressure sensors on the wings, where the sensor installment has been attempted to result in as small an intrusion as possible to the aircraft surface. The wing is shown before a layer of film is applied to reduce the roughness of the surface. After applying the layer of film, small rectangles were cut in the film to leave the BMP280 sensors open to the environment. Figure 4, shows the placement of all BMP280 sensors on the portside of a 3D model of the Skywalker X8.

[3]Microelectromechanical systems.

A. Wind tunnel tests setup

A part of the development and verification of the results in this paper is based on wind tunnel testing. The tests were conducted in the BLTW Slovak University of Technology (STU) wind tunnel in Bratislava [11]. The BLTW STU wind tunnel allows testing in both laminar and turbulent wind flow depending on the section of the tunnel used. For the purpose of the method presented in this paper, laminar wind flow was chosen and all results presented are from tests conducted in the designated laminar flow area of the tunnel. The BLTW STU wind tunnel is 14.6 m long and has a cross section of 2.6 x 1.6 m.

For the wind tunnel tests a PTU-D48 pan-and-tilt unit by FLIR Systems was employed as a part of the mount of the Skywalker X8 in order to control AOA and SSA. The PTU-D48 offers a precision up to 0.006° and is controlled through a computer, which enabled altering the AOA and SSA during the tests without having to stop/start the wind tunnel for every single AOA and SSA. Since the wind tunnel has a few minutes transition time before reaching a steady state airspeed, this allowed for a much faster data collection rate. A picture of the wind tunnel setup is shown in Figure 5. The data from PTU-D48 and the BMP280 sensors were collected and synced using DUNE: Unified Navigation Environment, developed by the Underwater Systems and Technology Laboratory [22].

The BLTW STU is equipped with three Scanivalve DSA3217 pneumatic pressure scanners. The DSA3217 pressure scanners measures with a frequency of 10 Hz and offer a ± 1.25 Pa full scale long term accuracy, corresponding to an accuracy improvement compared to the BMP280 of almost a factor 10. The DSA3217 pressure scanners have been connected through rubber tubes to thin cobber pipes embedded in the Skywalker X8, flush with the surface. The pressure scanners have been used to measure the pressure at the points at the tips of the cobber pipes at the surface of the Skywalker X8. The high accuracy pressure scanners allows insight into the impact of sensor accuracy on the estimation method by comparing results obtained using the high accuracy DSA3217 pressure scanners to the results obtained using the lower accuracy BMP280s. Since it is not possible to place both types of pressure sensors directly on top

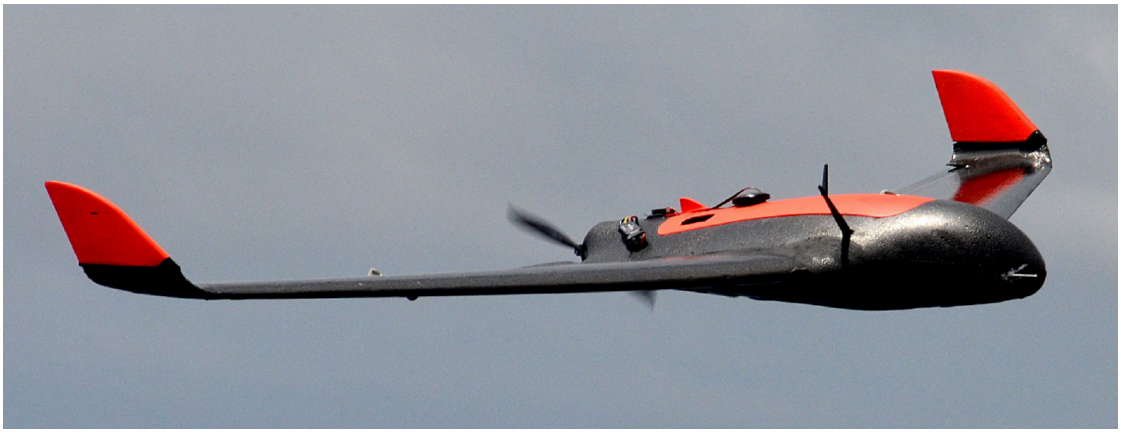


Fig. 2: A Skywalker X8 in flight.

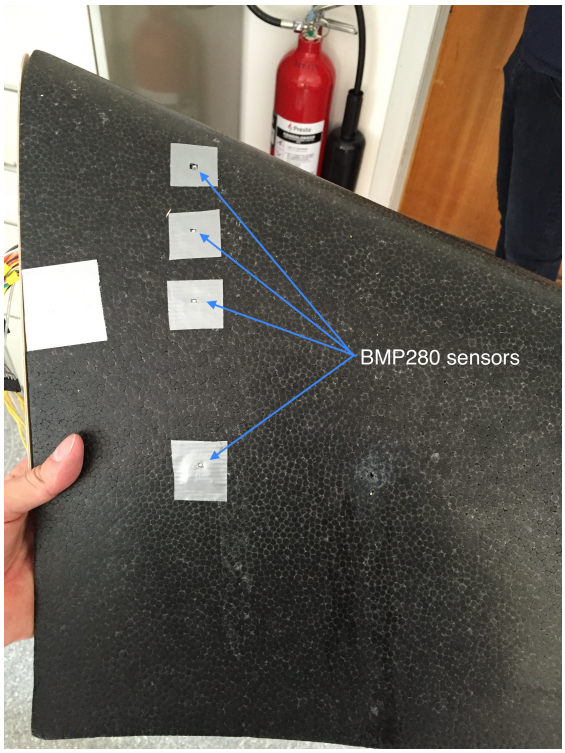


Fig. 3: A top view of the Skywalker X8 starboard wing showing four BMP280 pressure sensors.

of each other and to avoid cross influence between sensors, the DSA3217 pneumatic cobber tubes have been placed in different positions from the BMP280s and furthermore, a total of 22 pneumatic sensors has been placed on the Skywalker X8 and the placement is shown in Figure 4. It should be noted that since the placement and numbers of sensors will influence the results, the comparison between DSA3217 pneumatic and MEMS-based pressure sensors is not ideal, but should still serve to give insight into the advantage of a higher number of higher accuracy sensors on the presented method.

The wind tunnel tests were conducted the 7th and 8th of September, 2016. The tests encompassed six different airspeeds: 11.01 m/s (T = 27.4 °C, ρ = 1.1573 m/kg^3), 12.09 m/s (T = 27.4 °C, ρ = 1.1573 m/kg^3), 13.12 m/s (T = 27.5 °C, ρ = 1.1568 m/kg^3), 14.14 m/s (T = 27.6 °C, ρ = 1.1564 m/kg^3), 15.21 m/s (T = 27.6 °C, ρ = 1.1559 m/kg^3), 16.23 m/s (T = 27.7 °C, ρ = 1.1555 m/kg^3), and 17.32 m/s (T = 27.8 °C, ρ = 1.1555 m/kg^3). At high airspeeds, the Skywalker X8 looked to be under some flutter and a high static wing load. For that reason, no tests with airspeed higher than 17.32 m/s were attempted, although the Skywalker X8 is usually flown in airspeeds up to 25 m/s. The inability to cope with high airspeeds in the wind tunnel are assumed to primarily be attributed to two different factors. Firstly, the Skywalker X8 blocks out a sizeable amount of the tunnel cross section, this will result in the airspeed being higher around the aircraft because of the blockage effect. Secondly, the free stream turbulence is approximately 5% and since the shear stress in turbulent flow is higher than in laminar [3], this results in higher drag on the X8.

The AOA and SSA values were chosen to span a large range of different values, since ML algorithms are better suited for interpolation than extrapolation. The Skywalker X8 was therefore also tested in stall conditions. The nonlinear relation between lift coefficient and AOA for the stall regions, is assumed to also influence the pressure distribution in a nonlinear fashion, making the ML modeling more difficult. The AOA values tested for extend up to 35 °, which is beyond the AOA values normally identified with regular flight. AOA values in this range are usually only associated with special maneuvers such as deep stall landings, see Mathisen et al. [20], or other agile maneuvers or high turbulence. The intended purpose of the air data parameter measurement system should be considered when selecting the training data range for AOA, SSA and airspeed. Given the discussion in

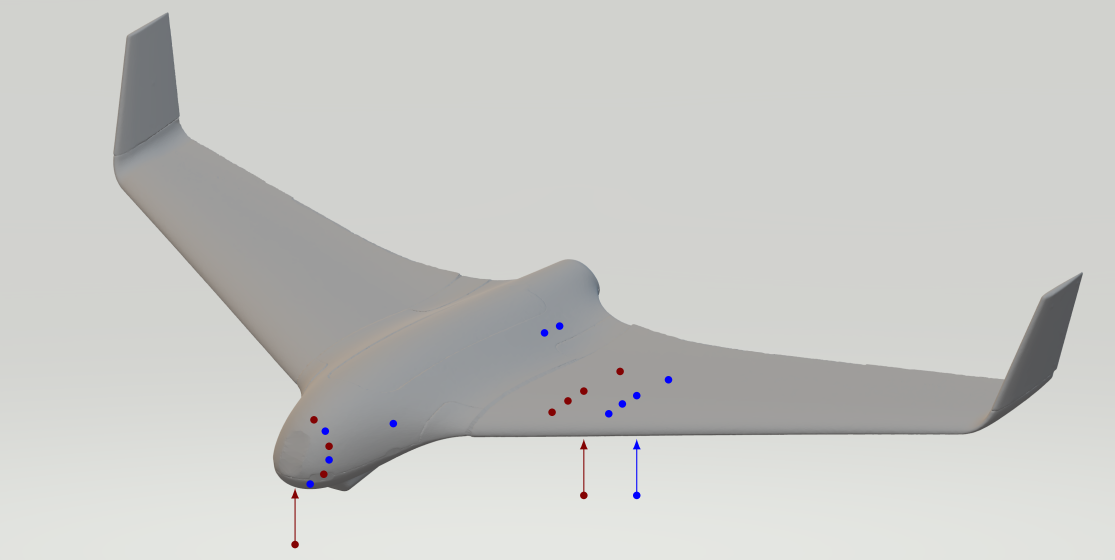


Fig. 4: Sensor placement illustrated on the Skywalker X8 UAV. The red dots correspond to the BMP280 sensors and the blue dots correspond to the DSA3217 pressure scanner measurement points described further down. The dots with connected line and arrow denotes a sensor or measurement point, placed on the underside. Note that besides the two BMP280 middle nose sensors, the remaining sensors and measurement points are only shown on the port side, however, the placement is symmetric on the starboard side.

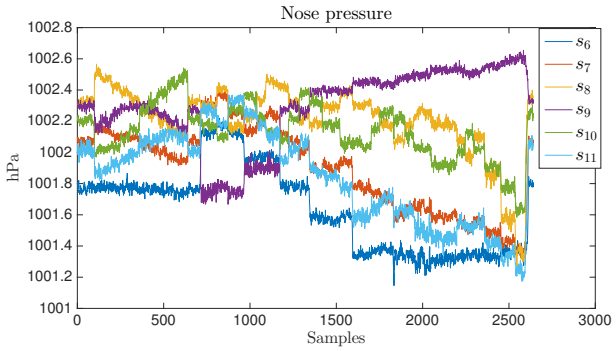
Fig. 5: A sideways view of the wind tunnel setup with the Skywalker X8 mounted on the PTU-D48.

Section II, it is also important to have data spanning a large range of Re numbers, and since the Re number is directly proportional to the airspeed, this was attempted fulfilled by varying the airspeed.

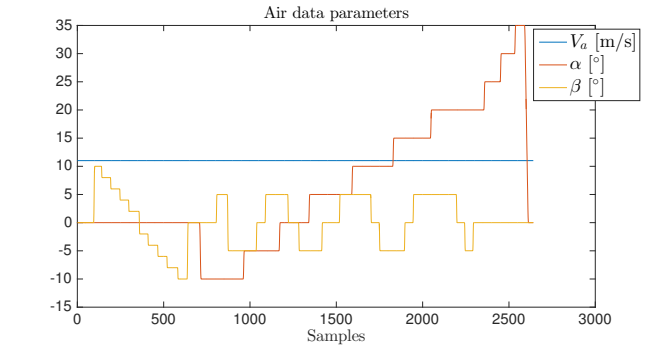
The raw pressure data for the Skywalker X8 nose BMP280 sensors and the corresponding air data parameters for the test with airspeed 11.01 m/s are shown in Figure 6. It is worth noticing that some of the sensors appear to have a very small response to changes in SSA while other sensors likewise appear to largely not be affected by the changes in AOA. This is related to the geometry of the Skywalker X8 relative to the wind and indicates that these sensors would be less valuable in estimating the corresponding parameter compared to the sensors with a larger change in response.

B. Flight test setup

For the flight test, the same Skywalker X8 as employed in the wind tunnel tests was used. However, since flight does not allow for connection by cable, the test setup was different from the tunnel test setup in a few ways. The Scanivalve DS3217 pressure scanners have not been possible to use, since they are too heavy and cumbersome for the Skywalker X8. The 16 embedded BMP280 pressure sensors are therefore the only embedded pressure sensors used during flight. The BMP280 sensors are read using an Arduino Mega 2650 and are sent through a serial connection to an ODROID-XU4 where it is logged along with GNSS and IMU data with precise timestamps using a synchronization board developed by Albrektsen and Johansen [1]. As a source of ground-truth air data parameters for ML and testing, the Micro Air Data System by the Aeroprobe Corporation has been used. The Micro Air Data System consists of a 5-port air data probe connected through rubber tubes to a small pressure scanner. The Micro Air Data System has an accuracy of $\pm 1^\circ$ on flow angles and has a total flow velocity accuracy <1 % or 1 m/s (whichever is larger). The Micro Air Data System air data probe is mounted on the nose section of



(a) The Skywalker X8 nose pressure measured by the six BMP280s embedded in surface.



(b) The air data parameters, where α and β are the angles of the PTU-D48 pan-and-tilt unit.

Fig. 6: The raw data from the wind tunnel test with an airspeed of 11.01 m/s.

the Skywalker X8. The mounting mechanism can result in a slight misalignment between the probe and the aircraft axes from which the AOA and SSA are defined with respect to, and this should be compensated for. Another error source is that the Aeroprobe saturates beyond $\pm 21°$ for AOA and SSA and any measurements outside this range will be truncated to $\pm 21°$. This will have a negative influence when using ML to estimate AOA and SSA values above 21°, and will also inhibit result assessments for angles outside of this range.

The flight test was conducted on the 20th of September 2016 on a field in Udduvoll, Norway. For the period of the flight, there was a temperature of approximately 16°C. The data set begins at the catapult launch of the Skywalker X8 and ends directly after landing. During the flight that lasts just below 37 minutes, the pilot operated the UAV in a large square with some changes in pitch and altitude. For the last part of the flight, the pilot engaged in some high dynamic maneuvers. The flight has been divided into two separate segments. The first five minutes of the flight has been separated from the remaining 31 minutes and will throughout the test be used as a flight benchmark test in order to assess the ML algorithms capabilities in estimating air data parameters for a flight scenario.

V. Results

This section contains the results obtained using the ML approaches on both wind tunnel test data and flight data. First, the BMP280 wind tunnel data is used to train and test the NN and LR approaches to see if the ML methods are capable of modeling the wind tunnel data input and output relations. Afterwards, a comparison with the more accurate DS3217 follows. The main benchmark test of the ML method capabilities will be a five minute test segment of flight with the Skywalker X8. The wind tunnel trained ML algorithms will be tested on this data to assess whether it is feasible to use the obtained wind tunnel data in order to train the ML algorithms for estimating air data parameters during flight. Afterwards, ML algorithms trained using the other segment of flight will be tested on the five minute benchmark flight and the results will be compared and discussed. Then, the impact of choice of sensor configuration on the accuracy of the results will be evaluated, as well as using a pseudo Reynolds number as an extra input.

A. Wind tunnel test results

The focus of this section will be on the results obtained using the BMP280 sensors on the wind tunnel data and whether the ML algorithms are capable of modeling the air data parameters from the measured pressure. Results obtained with the BMP280 data using different LR and NN configurations are displayed in Table I and II, respectively. The tables contain the RMSE for the air data parameter estimates from both the training data and the test data, however, the validation data results from the NNs have been left out.

Looking into the numerical LR results, it appears that the basis function expansion of adding first order cross terms is the single function augmentation that yields the best numerical results. However, it does also expand the input vector with 105 extra entries, since

$$\text{number of extra entries} = \binom{n}{k} \tag{8}$$

where n=15 and $k = 2$, whereas the other feature expansions only increase the vector size with an additional 15 entries each. The remaining basis function expansions yield similar results. The best numerical results are obtained for the basis function expansion **BXQC**, but the RMSE values are relatively close to the results obtained expanding only with the first order cross terms. The choice of LR basis function expansion will therefore be a trade-off between accuracy and computational cost, but for this data set it appears that the chosen function expansions leave a limit on the level of obtainable accuracy when compared to the NNs.

The results obtained using NNs are from an accuracy perspective superior to the LR results. The estimate RMSE

	Linear Regression				
	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.4519	0.1541	0.3005	0.3280	0.1465
$\bar{V}_a$ [m/s]	0.4541	0.1558	0.2987	0.3260	0.1505
$\check{\alpha}$ [°]	1.2490	0.4833	0.8252	0.8967	0.4371
$\bar{\alpha}$ [°]	1.2650	0.4889	0.8333	0.9044	0.4704
$\check{\beta}$ [°]	1.0462	0.6079	0.8033	0.7972	0.5950
$\bar{\beta}$ [°]	1.0647	0.6384	0.8073	0.8088	0.6094

TABLE I: The results obtained using LR with different basis function expansions on the wind tunnel data. Vee denotes the training set RMSE and bar denotes the test set RMSE.

	Neural Networks					
	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.1369	0.1171	0.0991	0.0877	0.0829	0.0498
$\bar{V}_a$ [m/s]	0.1391	0.1239	0.1060	0.0986	0.0938	0.0689
$\check{\alpha}$ [°]	0.3950	0.3522	0.3491	0.3275	0.3303	0.3286
$\bar{\alpha}$ [°]	0.4086	0.3860	0.3799	0.3675	0.3716	0.3642
$\check{\beta}$ [°]	0.5605	0.4830	0.4456	0.4138	0.3942	0.3690
$\bar{\beta}$ [°]	0.5738	0.5200	0.5018	0.4650	0.4608	0.4305

TABLE II: The results obtained using NNs with different structures on the wind tunnel data. Vee denotes the training set RMSE and bar denotes the test set RMSE.

decreases with the increase in the NN structure complexity which might hint that more neurons and more layers provide more accurate estimates. It appears that the difference between training and test set RMSE grows with NN complexity, which could indicate a small degree of overfitting for the more complex NN structures, as expected.

Figure 7 consists of two plots of the results obtained using LR and Figure 8 consists of the two corresponding plots using NNs. These plots contain the true values along with the estimates using the complete data set, i.e. both training data and test data (and validation data for the NN), and the error between true values and estimates. For the LR results, it appears that the accuracy of the airspeed is relative constant, but for NN results, the accuracy is slightly higher for lower airspeeds. For higher AOA and SSA, both methods exhibit less accurate estimates compared to lower AOA and SSA values.

Figure 9 shows the difference between the NN and LR estimates from Figure 7 and Figure 8. It appears that for the wind tunnel tests, the biggest difference between the two ML algorithms occurs when the AOA assumes high values and this is assumed to be attributed to nonlinearities and turbulence introduced when the wing is under stall conditions.

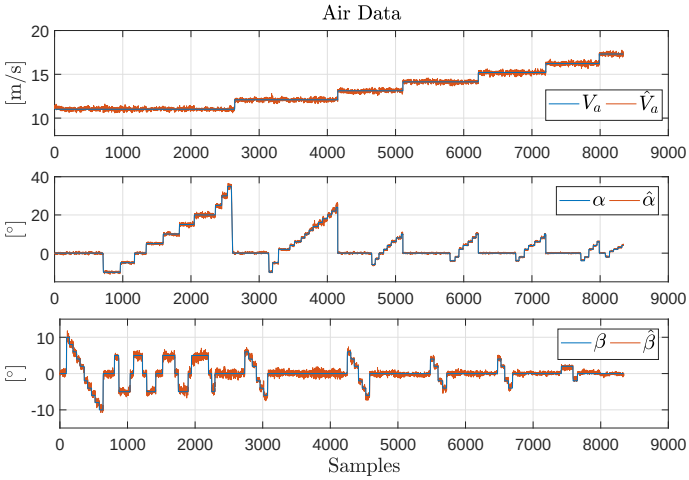
B. Comparison with higher accuracy pressure sensor

To obtain insight into the advantage of using more sensors of higher accuracy on the method, a comparison with the Scanivalve DSA3217 pneumatic pressure scanner setup is presented. As mentioned previously, the DSA3217 is too heavy and large to install on the Skywalker X8 and was therefore not possible to use during flight. Furthermore, the setup utilizes more sensors than the BMP280 setup, and the measurement points are not identical. The comparison between the two sets of sensors can therefore as such not provide a complete picture of the influence of sensor amount and sensor accuracy on the method, but it can function towards an impression of the highest attainable estimation accuracy given the used sensor accuracy, as well as an understanding of the influence of ML algorithm choice on the results. For the comparison between BMP280 sensors and the DSA3217, only a single LR and NN structure has been chosen, **BXQC** and $\mathbf{F}_{10}\mathbf{S}_0$, respectively. A plot of the $\mathbf{F}_{10}\mathbf{S}_0$ NN estimation results are shown in Figure 10 and the numerical results are listed in in Table III.

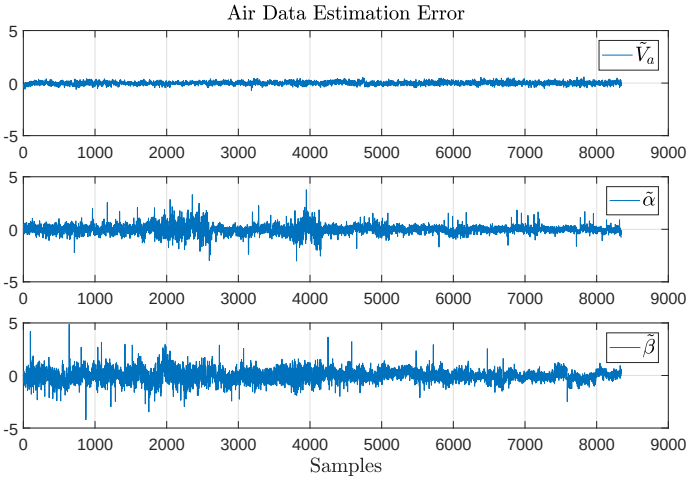
The spikes in estimation error seen in Figure 10, is assumed primarily to be a result of the built-in low pass filter of the DSA3217 pressure scanner, resulting in a transient phase between reaching the PTU-D48 pan-and-tilt unit set points and the pressure measurements. There are therefore two sets of results in Table III. In the first set of results, the data has been used without any processing (labeled as *raw data*). In the second set of results, the transition phase and has been removed from the data before training and testing (labeled as *stationary data*). It appears that the higher number of sensors, and higher accuracy of the DSA3217 pressure scanners improves the numerical results for the ML approaches when compared to the less accurate BMP280 measurements. Based on these results, it is expected that an increase in number of sensors and sensor accuracy will improve the output of the ML algorithms, and that the choice of sensor and quantity of

	BXQC (raw data)	$\mathbf{F}_{10}\mathbf{S}_0$ (raw data)	**BXQC** (stationary data)	$\mathbf{F}_{10}\mathbf{S}_0$ (stationary data)
$\check{V}_a$ [m/s]	0.0205	0.0172	0.0169	0.0128
$\bar{V}_a$ [m/s]	0.0213	0.0179	0.0182	0.0140
$\check{\alpha}$ [°]	0.2213	0.1796	0.0285	0.0235
$\bar{\alpha}$ [°]	0.2347	0.2062	0.0328	0.0247
$\check{\beta}$ [°]	0.3497	0.3311	0.0447	0.0370
$\bar{\beta}$ [°]	0.3678	0.3610	0.0452	0.0373

TABLE III: The results obtained using LR and NN on the DSA3217 pressure scanner measurements. Vee denotes the training set RMSE and bar denotes the test set RMSE.

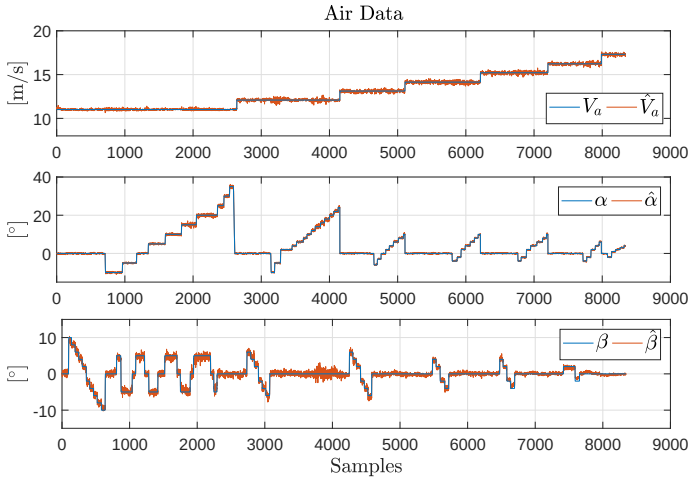


(a) True air data parameter values and the corresponding LR estimates, denoted by the hat accent.

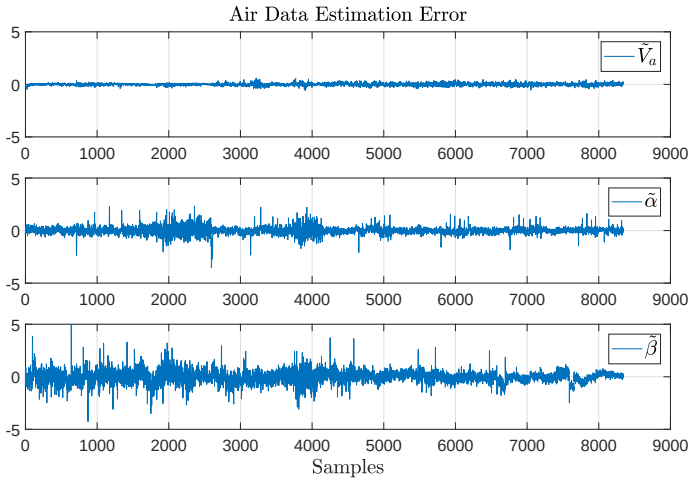


(b) The LR air data parameter estimation error, denoted by the tilde accent.

Fig. 7: Results obtained using LR with the basis input vector augmented with the first order cross terms, the quadratic term, and the cubed term (**BXQC**). The results displayed consists of the complete wind tunnel data set.



(a) True air data parameter values and the corresponding NN estimates, denoted by the hat accent.



(b) The NN air data parameter estimation error, denoted by the tilde accent.

Fig. 8: Results obtained using NNs with 10 neurons in a single hidden layer ($\mathbf{F}_{10}\mathbf{S}_0$). The results displayed consists of the complete wind tunnel data set.

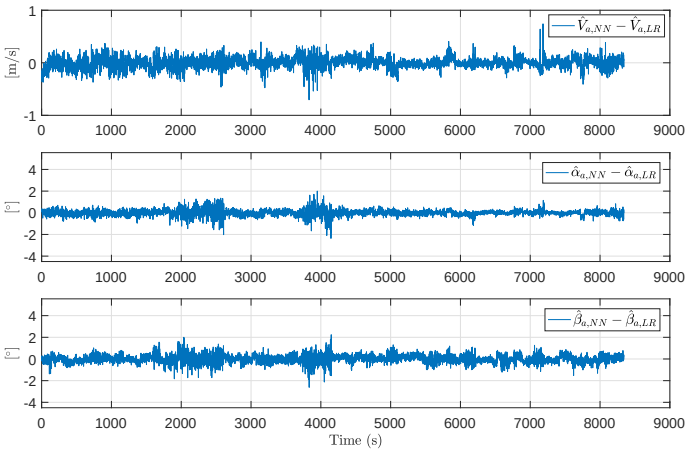
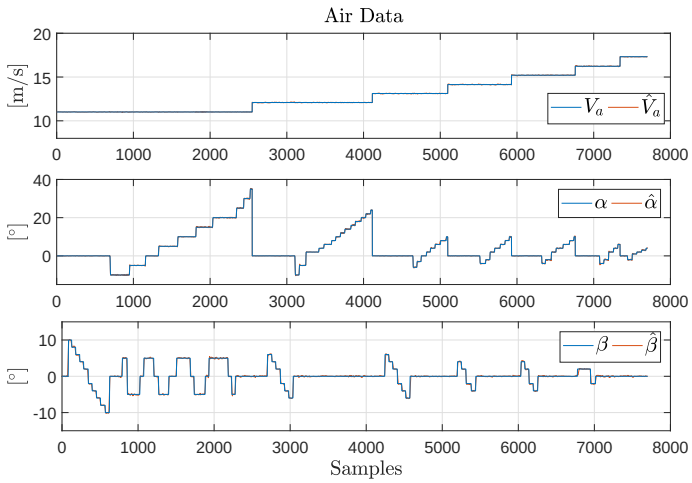
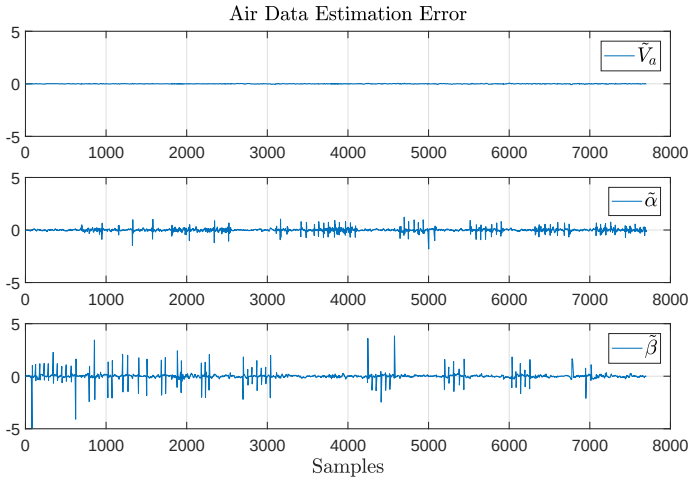


Fig. 9: The difference between the NN and LR estimates, respectively denoted by $\hat{V}_{a,NN}$ and $\hat{V}_{a,LR}$, from Figure 7 and Figure 8.



(a) True air data parameter values and the corresponding NN estimates, denoted by the hat accent.



(b) The NN air data parameter estimation error, denoted by the tilde accent.

Fig. 10: Results for the NN ($\mathbf{F}_{10}\mathbf{S}_0$) estimation method using the DSA3217 pressure scanner measurements and displaying the complete wind tunnel data set.

sensors will be a trade-off between price, weight, size, and power consumption of the sensors and the accuracy of the air data parameter estimates.

The results presented so far indicates that a more complex structure of the ML approach provides better results and that the NNs have slightly lower RMSE values than the LR. Also, there is reason to believe that using higher accuracy sensors results in higher accuracy estimates. However, a weakness with the approach so far is that the estimates are trained and tested with a set of data with identical conditions and only a rather limited set of different air data parameter points. Should the trained LR and NN structures be tested with a different set of data with air data parameters differing from the points trained for, there is no guarantee on the quality of the results. The test and training of ML methods on flight benchmark test data is therefore the primary focus of the remaining part of the results section.

C. Wind tunnel trained ML algorithms on benchmark flight test

This section contains the results obtained using the ML methods trained using the wind tunnel data on the benchmark flight test. For this comparison, 100% of the wind tunnel data has been used in training the LR method and for the NN method, the wind tunnel data has been divided into 85% training data set and a 15% validation data set. The initial numerical results using the wind tunnel data for training (not included here) showed low estimation accuracy on the benchmark flight test. Given a significant difference between the RMSE of the training set results and the test set results, it is assumed that the poor results are not primarily attributable to poor sensor accuracy, but instead stems from one or several of the following factors.

- There might have been a misalignment between the AOA and SSA in of the Skywalker X8 in the wind tunnel and the mounting of the Aeroprobe sensor. This would result in biases on the AOA and SSA estimates corresponding to the misalignment.
- As mentioned previously, the Skywalker X8 blocked out a sizeable amount of the wind tunnel. This could therefore result in the airspeed around the Skywalker X8 being higher than the wind tunnel airspeed used in training the ML methods. This could in turn result in the estimates provided by the ML methods being too low.
- ML methods are not suited for extrapolation and a considerable fraction of the airspeeds experienced by the Skywalker X8 in the flight test, was higher than the highest value tested for in the wind tunnel tests.

Courrieu [8] suggested a geometric approach of using a convex hull polytope to find the domain of validity of a feedforward NN and this limit is obviously violated by the airspeed, without even having to consider the remaining values of the estimated parameters.

- Too few training data points to properly model the (biased) input-to-output relationship. Rohloff et al. [26] discusses a binning technique to choose a proper training set from a larger data set. The training data is divided into bins in the different dimensions and choosing data following this technique then ensures a decent density distribution of the training data. However, for the wind tunnel data this technique could be misleading since the data could appear to be evenly divided into different bins, while in reality, many of the points in the bins would be from the same static air data parameter wind tunnel set points. Instead, data collected from flight tests will add a stochastic element that ensures that no air data parameter points are exactly the same, and that should result in a finer resolution and a better distribution compared to the wind tunnel.

Furthermore, training the ML methods using data containing stall conditions introduce strong nonlinearities into the system, and modeling will require even more data and represents an unnecessary difficulty if the intention is to not fly under stall conditions[4].

In order to remove the effect of blockage and misalignment on the results, a new set of results have been generated where the training data have had bias correction terms added to it. The bias have been estimated as the mean of the error for 10 instances of training $\mathbf{F}_{10}\mathbf{S}_0$ NNs with the wind tunnel data and testing on the benchmark flight. The biases were found to be 3.24 m/s for the airspeed, 10.59° for the AOA, and -1.63° for the SSA. The results are listed in Table IV and V.

Modifying the training data in an attempt to account for the effects of blockage and misaligned sensors improved the numerical results. However, the high AOA and SSA RMSE for the LR method, may indicate that there are significant nonlinearities in the wind tunnel data. The NN results indicate that the more advanced structures are prone to overfitting. As mentioned previously, the numerical results are found as the mean of the results obtained from 10 different instances of NNs with identical structure. The results from each of these 10 NNs show a large variation in the RMSE of the different estimates. The best NNs were capable of obtaining a RMSE on the flight benchmark test of 0.54 m/s for the airspeed, 0.77° for the AOA, and 1.37° for the SSA, which is significantly lower than the mean of the results from the 10 NNs. This indicates that the results could be improved by a training data choosing algorithm, perhaps using a binning technique as described previously. The high variance on the accuracy of the results is assumed to be primarily due to a lack of a high resolution wind tunnel training data set that covers all possible air data parameter values that the Skywalker X8 can be expected to experience in flight.

[4]It should be noted that the comparison sensor, the Micro Air Data System by the Aeroprobe Corporation, is not capable of measuring in the stall region either.

D. Flight trained ML algorithms on benchmark flight

This section contains the results of training the ML methods with a 31 minute flight training set and evaluating the methods on the flight benchmark test. The last part of the training segment involved the Skywalker X8 flying in high dynamic maneuvers and the training data set contains data that eclipses the test set and the algorithms will therefore not have to rely on extrapolation. The results are shown in Table VI and VII. The results obtained by training the ML algorithms using flight data gives much better results than the ones obtained using the wind tunnel data for training. It appears that the test LR results obtained are not that dependent on the structure of the LR, although it seems beneficial to add a single basis function expansion to increase the the accuracy of the estimates. For the NN, the results appear to be relatively independent on the chosen structure, which indicates that for the conditions trained and tested in, the relation between pressure and air data parameters must be possible to model by three five-neuron NNs. Furthermore, the numerical results are relatively close in accuracy and both methods seems viable to use for similar flight conditions if properly trained.

Figure 11 and 12 contains plot of the results obtained using the flight data trained **BXQC** LR and $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithms. The plots show relatively low estimation errors from both algorithms. The improvements of the results from using flight data compared to wind tunnel data is expected to be caused by the removal of the error points discussed in the previous section - there is no misalignment of angles, the airspeed used in the training data is not biased due to blockage effect, and there is no extrapolation. However, the biggest improvement is assumed to be attributed to the data being distributed in relevant flight conditions and with a much finer parameter resolution which allows the methods a better chance at correctly modeling the input/output relations.

Figure 13 shows the difference between the NN and LR algorithm estimates Figure 11 and Figure 12. The plot indicates that the two algorithms provide the largest differences in estimates when the air data ground truth values deviate the furthest from steady flight conditions. This is assumed to be due to less training data covering these air data values.

E. Sensor configuration influence

So far, all the algorithms have been trained using the data from all the BMP280 sensors. However, a part of the novelty of the presented method lies in the option of adjusting the sensor configuration to whichever aircraft is the desired platform of implementation. This section investigates the relation between different sensor configurations and their corresponding results in estimating the air data parameters. The results will obviously only be valid for the Skywalker X8 UAV, but can perhaps serve as an indicator of how the

	Linear Regression				
	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.4521	0.1541	0.3001	0.3275	0.1468
$\bar{V}_a$ [m/s]	1.2075	1.4815	1.7577	3.5591	1.0415
$\check{\alpha}$ [°]	1.2512	0.4828	0.8260	0.8974	0.4406
$\bar{\alpha}$ [°]	8.6618	2.4905	3.4954	17.0525	7.7749
$\check{\beta}$ [°]	1.0488	0.6115	0.8036	0.7986	0.5959
$\bar{\beta}$ [°]	4.1666	5.1462	2.8018	5.3993	4.9833

TABLE IV: The results obtained using LR trained with wind tunnel data with added bias corrections on the flight benchmark test. Vee denotes the training set RMSE and bar denotes the test set RMSE.

	Neural Networks					
	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.1341	0.1154	0.0961	0.0818	0.0773	0.0567
$\bar{V}_a$ [m/s]	0.7337	0.9642	1.0603	1.2318	1.0871	1.3066
$\check{\alpha}$ [°]	0.4004	0.3566	0.3411	0.3190	0.3334	0.3148
$\bar{\alpha}$ [°]	1.0877	1.5216	1.7246	3.7936	2.2111	3.8493
$\check{\beta}$ [°]	0.5665	0.5003	0.4471	0.4220	0.4122	0.3784
$\bar{\beta}$ [°]	2.2016	2.4157	2.5866	3.1320	2.6473	2.0639

TABLE V: The results obtained using NNs trained with wind tunnel data with added bias corrections on the flight benchmark test. Vee denotes the training set RMSE and bar denotes the test set RMSE.

	Linear Regression				
	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.3815	0.2865	0.3213	0.3227	0.2760
$\bar{V}_a$ [m/s]	0.4439	0.3587	0.3655	0.3696	0.4088
$\check{\alpha}$ [°]	0.3112	0.2010	0.2325	0.2465	0.1896
$\bar{\alpha}$ [°]	0.2736	0.2379	0.2609	0.2591	0.2328
$\check{\beta}$ [°]	1.3654	0.9162	1.0928	1.1371	0.8652
$\bar{\beta}$ [°]	1.3358	1.0058	0.9997	1.0572	1.0161

TABLE VI: The numerical results obtained using LR trained with flight data on the flight benchmark test set. Vee denotes the training set RMSE and bar denotes the test set RMSE.

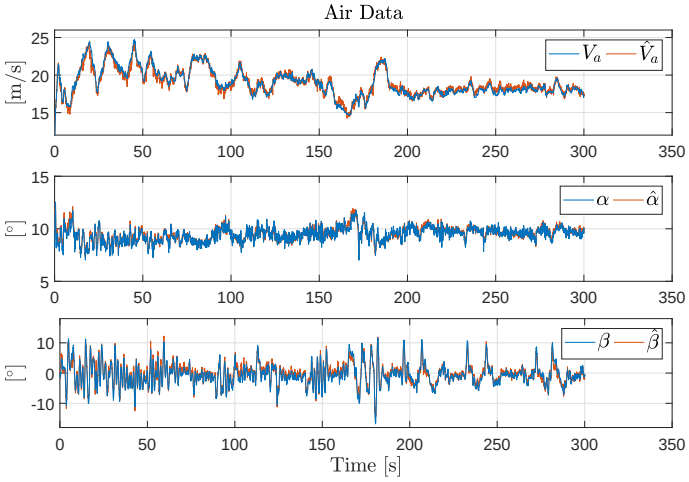
results would be on similar platforms (which would have to be trained using data acquired using that specific platform). The flight data trained **BXQC** LR and $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithms have been chosen as the basis ML algorithms for this sensor configuration assessment. Furthermore, a single test is included where the the full sensor configuration is augmented with a pseudo Reynolds number. The pseudo Reynolds number uses ambient pressure and temperature from the autopilot, but with the relative velocity in the forward direction, u_r, obtained from the Pitot-static tube as an approximation for the airspeed.

Reverting back to Figure 3 and 4, the naming convention of the sensors is that the first five sensors, p_{s_1} - p_{s_5} are located in the starboard wing. The topside sensor at the leading edge is p_{s_1} and the sensor numbers then increment towards and around the trailing edge with p_{s_5} as the sensor on the bottom of the wing. For the nose, the sensor located at twelve o'clock in Figure 4 is the reference sensor and the sensor on the right is p_{s_6}. Incrementing in a clock-wise pattern, the sensor to the left of the reference sensor is $p_{s_{10}}$. The port wing follows the same system as the starboard wing starting with sensor $p_{s_{11}}$ and ending with $p_{s_{15}}$ on the bottom side of the wing. For the tests, the LR and NN structures **BXQC** and $\mathbf{F}_{10}\mathbf{S}_0$ have been used once again. Regarding the choice of sensor configurations, it appears intuitive to evaluate the configurations that involves using only nose or wing sensors, since there might be limitations on a UAV platform related to these, e.g. propellers that hinders the use of pressure sensors in either nose or wings. It would also be advantageous to get an impression of the influence of using fewer sensors on the attainable accuracy. Fewer sensors would reduce the cost of the setup, reduce complexity of the implementation, and lower the risk of a single sensor malfunctioning.

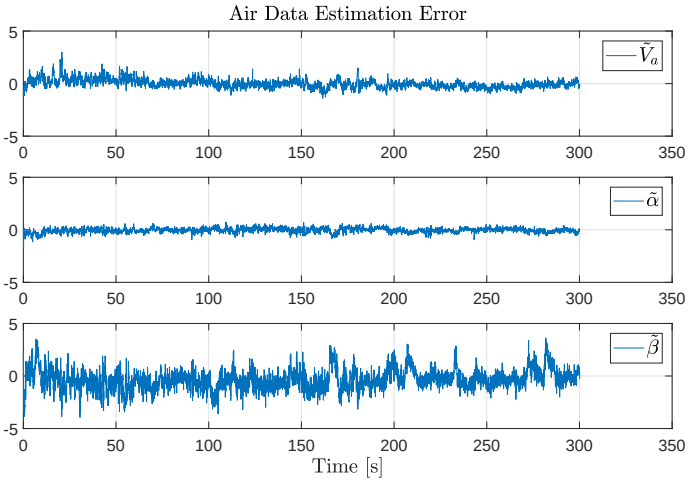
The numerical results from different sensor configurations and augmenting with the pseudo Reynolds number are shown

	Neural Networks					
	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_{1}0\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.2745	0.2516	0.2388	0.2305	0.2437	0.2326
$\bar{V}_a$ [m/s]	0.3503	0.3497	0.3460	0.3478	0.3372	0.3492
$\check{\alpha}$ [°]	0.1905	0.1735	0.1654	0.1613	0.1677	0.1622
$\bar{\alpha}$ [°]	0.2341	0.2298	0.2415	0.2445	0.2298	0.2313
$\check{\beta}$ [°]	0.8504	0.7584	0.7055	0.6592	0.6922	0.6676
$\bar{\beta}$ [°]	0.9165	0.9329	0.9585	0.9674	0.9322	0.9180

TABLE VII: The numerical results obtained using NNs trained with flight data on the flight benchmark test set. Vee denotes the training set RMSE and bar denotes the test set RMSE.

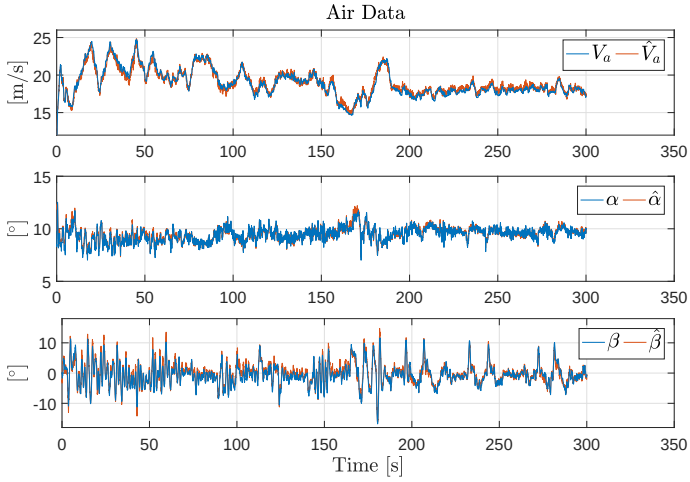


(a) True air data parameter values and the corresponding LR estimates, denoted by the hat accent.

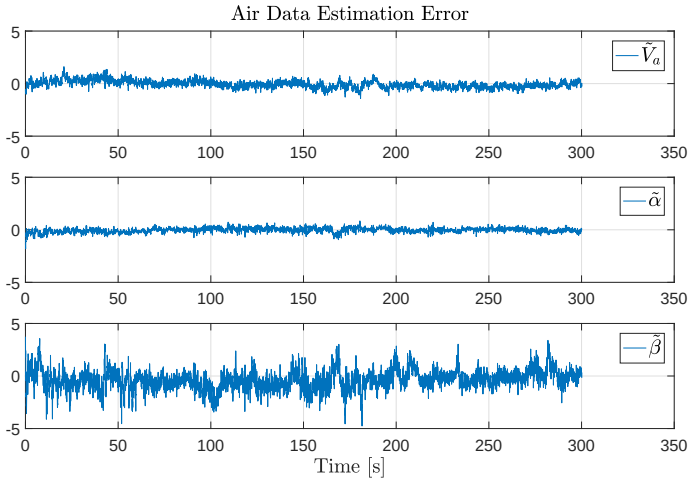


(b) The LR air data parameter estimation error, denoted by the tilde accent.

Fig. 11: Results obtained using a **BXQC** LR trained with flight data on the flight benchmark test set..



(a) True air data parameter values and the corresponding NN estimates, denoted by the hat accent.



(b) The NN air data parameter estimation error, denoted by the tilde accent.

Fig. 12: Results obtained using a $\mathbf{F}_{10}\mathbf{S}_0$ NN trained with flight data on the flight benchmark test set.

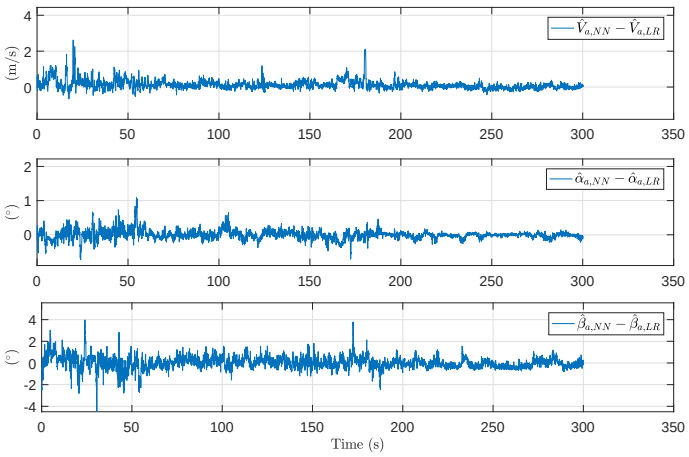


Fig. 13: The difference between the NN and LR estimates, respectively denoted by $\hat{V}_{a,NN}$ and $\hat{V}_{a,LR}$, from Figure 11 and Figure 12.

in Table VIII and IX, where the numbers are used for denoting the sensors used in the configuration, i.e. 5 corresponds to using sensor p_{s_5} in the configuration. Both tables show the same trends in accuracy with respect to the different sensor configurations. The nose sensors alone are relatively poor at estimating the airspeed, whereas adding a sensor from each wing greatly increase the accuracy. The wing sensors alone are on the other hand decent at estimating the airspeed, but suffer from poor SSA estimates, since the sensors only measure in two parallel planes, both perpendicular to the SSA. The choice of sensor configuration should be dependent on the UAV platform of implementation and will be a trade-off between accuracy, cost, and ease of implementation. Adding the pseudo Reynolds number only appears to provide a small benefit to the results. However, the flight results contain relatively constant ambient conditions, and the pseudo Reynolds number can potentially provide a higher level of robustness with respect to these conditions or be used in non-dimensionalizing the pressure measurements into pressure coefficients.

VI. Conclusion

A method for estimating the air data parameters for small fixed-wing unmanned aerial vehicles have been presented. The method comprises a set of low-cost MEMS-based pressure sensors embedded in the surface of the unmanned aerial vehicle combined with machine learning algorithms. A strength of the presented method is the flexibility of the pressure sensor placement. Two different machine learning algorithms, neural networks and linear regression, have been implemented and tested. Both algorithms have been evaluated on data obtained through wind tunnel experiments and real flight data. The neural network algorithm was found to generally provide a slightly lower estimation error than the linear regression approach presented. However, linear regression allows for basis function expansions that could potentially improve the results further. By comparing the results obtained from using the low-cost sensors on the aircraft against the results from the expensive pressure scanner from the wind tunnel, the influence of the number of sensors and sensor accuracy on the results was assessed. Training the machine learning algorithms using only wind tunnel data was found to have several error sources, namely the wind tunnel blockage effect, misalignment in the aircraft mount, as well as a too sparse (not very rich) training data set. However, trying to account for these error sources showed potential in decreasing the estimation error. Using flight data to train the machine learning algorithms was found to be a feasible approach that allowed estimating the air data parameters for both neural networks and linear regression. Finally, a study of the sensor number and placement influence on the result was conducted along with an assessment of augmenting the system input by using a pseudo Reynolds number.

Acknowledgment

This work was supported by the Norwegian Research Council (grant no. 221666 and 223254) through the Center of Autonomous Marine Operations and Systems (NTNU AMOS) and FORNY (project no. 90189003) at the Norwegian University of Science and Technology. The authors are very thankful for the syncboard provided by Sigurd Albrektsen, the theoretical expertise provided by Richard, and the collaboration in data collecting together with Andreas Wenz, all from NTNU. The authors would also like to thank for the help, hospitality, and wind tunnel time provided by Ol'ga Hubová and Boris Rohal'-Ilkiv at the Slovak University of Technology in Bratislava (STU). Lastly, the authors would like to thank the skilled pilots here at NTNU, Lars Semb and Pål Kvaløy, who flew the Skywalker X8 for the flight data set.

References

[1] Albrektsen, S. M. and Johansen, T. A. [2017], SyncBoard - A high accuracy sensor timing board for UAV payloads, *in* '2017 International Conference on Unmanned Aircraft Systems (ICUAS)'.

[2] Anderson Jr., J. D. [2011], *Fundamentals of Aerodynamics: Fifth Edition*, McGraw-Hill.

[3] Beard, R. W. and McLain, T. W. [2012], *Small Unmanned Aircraft - Theory and Practice*, Princeton University Press.

[4] Bishop, C. M. [2006], *Pattern Reckognition and Machine Learning*, Springer.

[5] Burnham, K. P. and Anderson, D. R. [2011], *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd Edition*, Springer.

[6] Callegari, S., Talamelli, A., Zagnoni, M., Golfarelli, A., Rossi, V., Tartagni, M. and Sangiorgi, E. [2004], 'Aircraft Angle of Attack and Air Speed Detection by Redundant Strip Pressure Sensors', *SENSORS, 2004 IEEE* .

[7] Cho, A., Kang, Y., Park, B. and Yoo, C. [2013], Airflow angle and wind estimation using GPS/INS navigation data and airspeed, *in* '2013 13th International Conference on Control, Automation and Systems (ICCAS)'.

BXQC LR

	All sensors + Re	**All sensors**	**Nose sensors**	**Wing sensors**	**5,6,7,8,9,10,15**	**5,7,9,15**
$\check{V}_a$ [m/s]	0.2593	0.2760	0.9329	0.6121	0.5120	0.7303
$\bar{V}_a$ [m/s]	0.4043	0.4088	0.8858	0.7293	0.5765	0.7371
$\check{\alpha}$ [°]	0.1809	0.1896	0.4213	0.2890	0.2955	0.3228
$\bar{\alpha}$ [°]	0.2271	0.2328	0.3837	0.4086	0.2737	0.3185
$\check{\beta}$ [°]	0.8546	0.8652	1.1743	2.2266	1.0059	1.1251
$\bar{\beta}$ [°]	1.0104	1.0161	1.0874	3.7345	0.9274	0.9826

TABLE VIII: The numerical results from using different sensor configurations and adding a pseudo Reynolds number when training and testing the **BXQC** LR algorithm. Vee denotes the training set RMSE and bar denotes the test set RMSE.

$\mathbf{F}_{10}\mathbf{S}_0$ NN

	All sensors + Re	**All sensors**	**Nose sensors**	**Wing sensors**	**5,6,7,8,9,10,15**	**5,7,9,15**
$\check{V}_a$ [m/s]	0.2375	0.2534	0.8512	0.5680	0.4534	0.6684
$\bar{V}_a$ [m/s]	0.3402	0.3535	0.8363	0.7891	0.4557	0.6713
$\check{\alpha}$ [°]	0.1655	0.1741	0.4030	0.2688	0.2815	0.3225
$\bar{\alpha}$ [°]	0.2167	0.2318	0.3873	0.4167	0.2857	0.3277
$\check{\beta}$ [°]	0.7439	0.7575	1.0746	2.0189	0.9051	1.0188
$\bar{\beta}$ [°]	0.9326	0.9507	1.0568	3.7503	0.9168	1.0232

TABLE IX: The numerical results from using different sensor configurations and augmenting with a pseudo Reynolds number when training and testing the $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithm. Vee denotes the training set RMSE and bar denotes the test set RMSE.

[8] Courrieu, P. [vol 7, pp. 169-174, 1994], 'Three algorithms for estimating the domain of validity of feedforward neural networks', *Neural Networks* .

[9] Flying [1995], Qualities of Piloted Aircraft, Technical report, United States of America Department of Defense.

[10] Hagan, M. T. and Menhaj, M. B. [vol. 5 pp. 989 - 993, 1994], 'Traiining Feedforward Networks With the Marquardt Algorithm', *IEEE Transactions on Neural Networks* .

[11] Hubová, O. and Lobotka, P. [vol. 10, pp. 1-9, 2014], 'The Multipurpose New Wind Tunnel STU', *Civil and Environmental Engineering* .

[12] James, G., Witten, D., Hastie, T. and Tibshirani, R. [2013], *An Introduction to Statistical Learning*, Springer.

[13] Johansen, T. A., Cristofaro, A., Sørensen, K. L., Hansen, J. M. and Fossen, T. I. [2015], On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors, *in* '2015 International Conference on Unmanned Aircraft Systems (ICUAS)'.

[14] Langelaan, J. W., Alley, N. and Neidhoefer, J. [vol. 34, pp. 1016-1030, 2011], 'Wind Field Estimation for Small Unmanned Aerial Vehicles', *Journal of Guidance, Control, and Dynamics* .

[15] Larson, T. J. and Siemers III, P. M. [1981], 'Subsonic Tests of an All-Flush-Pressure-Orifice Air Data System', *NASA TP-1871* .

[16] Larson, T. J., Whitmore, S. A., Ehernberger, L. J., Johnson, J. B. and Siemers III, P. M. [1987], 'Qualitative Evaluation of a Flush Air Data System at Transonic Speeds and High Angles of Attack', *NASA TP-2716* .

[17] Lie, F. A. P. and Gebre-Egziabher, D. [vol. 50, pp. 1234-1249, 2013], 'Synthetic Air Data System', *Journal of Aircraft* .

[18] Long, H. and Song, S. [2009], Method of estimating angle-of-attack and sideslip angle based on data fusion, *in* '2009 2nd International Conference on Intelligent Computing Technology and Automation'.

[19] Marquardt, D. W. [vol. 11, pp. 431-441, 1963], 'An Algorithm for Least-Squares Estimation of Nonlinear Parameters', *Journal of the Society for Industrial and Applied Mathematics* .

[20] Mathisen, S. H., Gryte, K., Fossen, T. I. and Johansen, T. A. [2016], Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing, *in* 'AIAA Infotech @ Aerospace'.

[21] P. Murphy, K. [2012], *Machine Learning: A Probabilistic Perspective*, The MIT Press.

[22] Pinto, J., Dias, P. S., Martins, R., Fortuna, J., Marques, E. and Sousa, J. [2013], The LSTS toolchain for networked vehicle systems, *in* 'OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension'.

[23] Quindlen, J. and Langelaan, J. [2013], Flush Air Data Sensing for Soaring-Capable UAVs, *in* '51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition'.

[24] Ramprasadh, C. and Arya, H. [vol. 49, pp. 93-100, 2012], 'Multistage-Fusion Algorithm for Estimation of Aerodynamic Angles in Mini Aerial Vehicle', *Journal of Aircraft* .

[25] Rhudy, M. B., Fravolini, M. L., Gu, Y., Napolitano, M. R., Gururanjan, S. and Chao, H. [vol. 51, pp. 1980-

1995, 2015], 'Aircraft model-independent airspeed estimation without pitot tube measurements', *IEEE Transactions on Aerospace and Electronic Systems* .

[26] Rohloff, T. J., Whitmore, S. A. and Catton, I. [vol. 36, pp. 2094-2101, 1998], 'Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method', *AIAA Journal* .

[27] Rohloff, T. J., Whitmore, S. A. and Catton, I. [vol. 36, pp. 541–549, 1999], 'Fault-Tolerant Neural Network Algorithm for Flush Air Data Sensing', *Journal of Aircraft* .

[28] Samy, I., Postlethwaite, I., Gu, D. and Green, J. [vol. 47, pp. 18-31, 2010], 'Neural-Network-Based Flush Air Data Sensing System Demonstrated on a Mini Air Vehicle', *Journal of Aircraft* .

[29] Wenz, A. and Johansen, T. A. [2017], Estimation of Wind Velocities and Aerodynamic Coefficients for UAVs using standard Autopilot Sensors and a Moving Horizon Estimator, *in* '2017 International Conference on Unmanned Aircraft Systems (ICUAS)'.

[30] Wenz, A., Johansen, T. A. and Cristofaro, A. [2016], Combining model-free and model-based Angle of Attack estimation for small fixed-wing UAVs using a standard sensor suite, *in* '2016 International Conference on Unmanned Aircraft Systems (ICUAS)'.

[31] Whitmore, S. A., Davis, R. J. and Fife, J. [vol. 33, pp. 970–977, 1996], 'In-flight demonstration of a real-time flush airdata sensing system', *Journal of Aircraft* .

Kasper Trolle Borup received the B.Sc. and M.Sc. degrees in Electrical Engineering from the Technical University of Denmark in 2010 and 2013, respectively. For the M.Sc. degree he attended the Honors Programme with a specialization in Automation and Robot Technology. He received the Ph.D. degree in Engineering Cybernetics from NTNU, Trondheim, Norway, in 2018. He is currently working at UBIQ Aerospace, Trondheim, Norway, where he is involved in developing an icing protection system for unmanned aircraft. His research interests include nonlinear observers and nonlinear control and their applications to unmanned aerial vehicles.

Thor Inge Fossen is a naval architect and a cyberneticist. He received an MTech degree in Marine Technology in 1987 and a PhD degree in Engineering Cybernetics in 1991 both from the Norwegian University of Science and Technology (NTNU), Trondheim. He is currently professor of guidance, navigation and control and co-director of the NTNU Centre for Autonomous Marine Operations and Systems. Fossen's expertise covers guidance systems, inertial navigation systems, autonomous systems, nonlinear control and observer theory, vehicle dynamics, hydrodynamics, autopilots and unmanned vehicles. He has authored six textbooks. Fossen is one of the co-founders and former Vice President R&D of the company Marine Cybernetics AS, which was acquired by DNV GL in 2012. He is also co-founder of SCOUT Drone Inspection AS (2017). He received the Automatica Prize Paper Award in 2002 and the Arch T. Colwell Merit Award in 2008 at the SAE World Congress. He has been elected to the Norwegian Academy of Technological Sciences (1998) and elevated to IEEE Fellow (2016).

Tor Arne Johansen received the MSc degree in 1989 and the PhD degree in 1994, both in electrical and computer engineering, from the Norwegian University of Science and Technology, Trondheim, Norway. From 1995 to 1997, he worked at SINTEF as a researcher before he was appointed Associated Professor at the Norwegian University of Science and Technology in Trondheim in 1997 and Professor in 2001. He has published several hundred articles in the areas of control, estimation and optimization with applications in the marine, aerospace, automotive, biomedical and process industries. In 2002 Johansen co-founded the company Marine Cybernetics AS where he was Vice President until 2008. Prof. Johansen received the 2006 Arch T. Colwell Merit Award of the SAE, and is currently a principal researcher within the Center of Excellence on Autonomous Marine Operations and Systems (NTNU-AMOS) and director of the Unmanned Aerial Vehicle Laboratory at NTNU.