International Forum on Aeroelasticity and Structural Dynamics
IFASD 2022
13-17 June 2022, Madrid, Spain

Comparison of EKF and Neural Network based wing shape estimation of a flexible wing demonstrator

Bence Zs. Hadlaczky[1], Noémi Friedman[1], Béla Takarics[1] and Bálint Vanek[1]

[1] Institute for Computer Science and Control
Eötvös Lóránd Research Network
Kende u. 13-17. 1111, Budapest, Hungary
hadlaczky.bence.zsombor@sztaki.hu
friedman.noemi@sztaki.hu
takarics.bela@sztaki.hu
vanek@sztaki.hu

Keywords: aeroelasticity, structural dynamics, LPV, Kalman filtering, neural network, KalmanNet.

Abstract: Structural flexibility of advanced, large-wingspan aircrafts is a crucial factor which has huge influence on the dynamics and stability of these vehicles. In case of a highly flexible wing structure, there is a need for an efficient observer to measure and predict the structural changes and dynamics of the wing. However, the modal coordinates of the wing cannot be measured directly so designing a state observer is necessary. Since the flexible aircraft model is nonlinear, the classical Kalman filter approach can have limited performance. Instead, two state observer approaches are investigated in the paper. First, we present a model-based method for designing an extended Kalman filter (EKF) when only a linear parameter-varying model (LPV) is available to describe the behaviour of the real aircraft. Second, we present a data-driven approach for this problem which is based on the new KalmanNet architecture. Finally, the results of the two methods are evaluated on the T-Flex model of the FLiPASED H2020 project.

1 INTRODUCTION

In the recent years, research and development trends in the aerospace industry placed more emphasis on increasing fuel efficiency [1]. The greatest portion of the operating costs of an aircraft today comes from fuel consumption, so achieving better fuel economy is a key aspect for cost reduction. To achieve these goals the decrease of structural mass and use of more flexible components are the most widely investigated solutions. The use of flexible components can mean the utilization of highly flexible wings [2]. However, this approach is not without its difficulties. First, the fuel consumption of the aircraft is largely dependent on the aerodynamic drag acting on the wings, so reducing drag with a suitable wing shape controller results better fuel economy. The other difficulty is - for one - the aeroelastic flutter which can cause serious structural damage to the wings, and the occurring instability and handling quality issues of the flight controller caused by the flexible structure dynamics of the vehicle [3]. So, supressing these effects with a flutter controller is a must to ensure safe flights. Control based active flutter suppression is investigated in several recent research projects. These are the Performance Adaptive Aeroelastic Wing (PAAW) project in the USA [4]. However, both the controller for drag reduction and the flutter controller [5] requires some information about the flexible

dynamics of the wings, which is not possible with direct measurements. As a result, a state observer/estimator is required to acquire information about the states which determine the flexible behaviour of the wings. This paper presents results from the ongoing Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FLiPASED) [6] project which is the continuation of the Flutter Free Flight Envelope Expansion for Economical Performance Improvement (FLEXOP) project. The FLEXOP [1] project's main goal was to develop an effective flutter suppression system. Therefore, one of the main goals of the FLiPASED project is to develop drag reducing control for aircrafts with highly flexible wings. The main motivation of this paper is to present 2 different approaches to estimate the flexible dynamics of the T-Flex demonstrator which can be then used for the design of a wing shape controller for drag reduction purposes.

The most straightforward solution for designing a state predictor is the Kalman filter [7] for linear systems and the extended Kalman filter (EKF) in the case of nonlinear systems. The EKF is proved to be useful in inertial estimation of wing shape [8]. However, it has two main drawbacks. First, the EKF requires the exact mathematical state-space description of the nonlinear system, which might not be available or simply it is too complex to be efficiently used in calculations. The second drawback is that the knowledge of the noises and disturbances related to observations and states is necessary. Defining observation noise is the less challenging task but giving an accurate estimation about the uncertainties and disturbances related to the model states might not be possible.

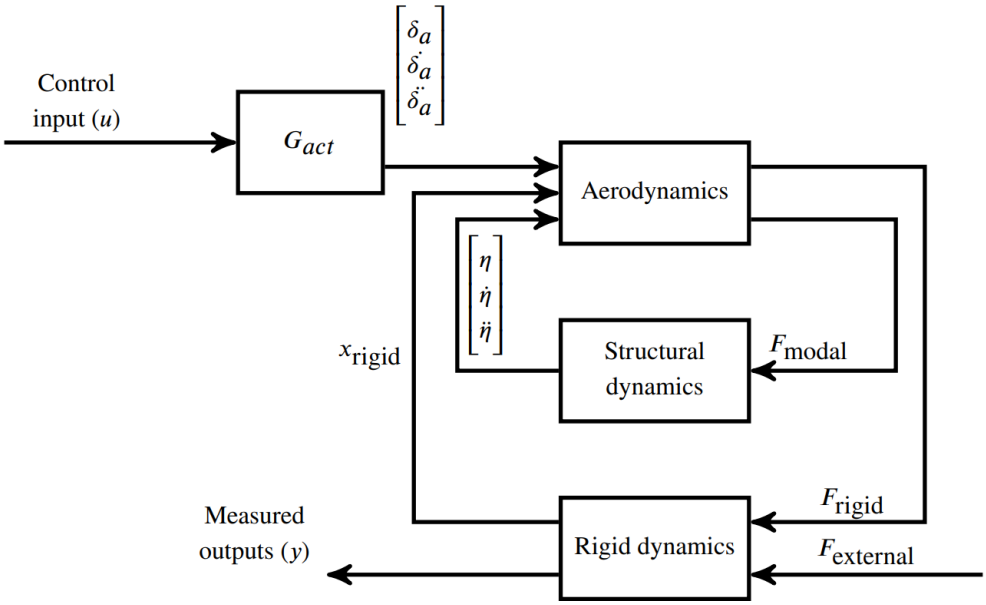
To solve the first issue, the approximation of the full, nonlinear system with a Linear Parameter Varying (LPV) model [9] can be considered. Using an LPV model can be less computation heavy and implementing an LPV-based EKF for predicting states is a feasible solution [10]. However, this solution still requires noise information. Data-driven approaches have the great advantage that they can be used for inertial odometry [11] and inertial aided navigation [12] problems without needing any specific information about model or observation uncertainties. However, estimating flexible states is a more complex problem. This is where the new KalmanNet architecture can be useful [13]. The KalmanNet is based on Kalman filtering however it uses a Recurrent Neural Network (RNN) to estimate a Kalman gain. So as a result, it does not need any information about the noises and model uncertainties present.

This paper presents the working of the LPV-based EKF and the KalmanNet for predicting the modal coordinates and aerodynamic lag states of the nonlinear model of an Unmanned Aerial Vehicle (UAV) T-Flex, which was created during the FLEXOP project for demonstrator purposes. The testing was carried out with MATLAB, Simulink simulations where the model received square wave, doublet control surface inputs to excite the flexible dynamics of the aircraft. The LPV-based EKF was also tested with the baseline controller of the aircraft which keeps the vehicle on an oval shaped track. The paper is organized as follows. In Section 2., the dynamic model of the FLEXOP demonstrator is presented. Section 3. introduces the reduced, LPV model of the original nonlinear system, while also discussing the working of the LPV-based EKF. In Section 4. the KalmanNet's basic structure is summarized with the chosen training hyperparameters. Section 5. contains the presentation of the results of the modal coordinate and lag state estimations. The accuracy of the LPV-based EKF and the KalmanNet is compared, while for the filter a real-life application is also presented.

2 FLEXOP DEMONSTRATOR DYNAMIC MODEL

The chosen system for our research is the nonlinear, state space representation of the FLEXOP demonstrator aircraft. The model consists of 3 main parts: states that are responsible for the description of the rigid body dynamics; states related to flexible dynamics and aerodynamics, and finally, states that represent the control surface inputs and their first derivatives. The

interconnection of these subsystems forms the nonlinear aeroservoelastic model of the aircraft as seen in 1. Figure.



1. Figure – Aeroservoelastic subsystem interconnection

The structural dynamics of the demonstrator was represented with the use of a finite element (FE) complemented by Guyan reduction to reduce the number of degrees of freedom (DOF). Based on this model the equations of motion were formulated both for the rigid and the flexible aircraft motion. The aerodynamic model is based on a set of trapezoidal aerodynamic boxes which represent the lifting surfaces. The flow field of the demonstrator is approximated either with vortices or doublets applied to the aerodynamic boxes. The first method is called vortex lattice method (VLM) while the second is the doublet lattice method (DLM). VLM represents the steady aerodynamics, DLM incorporates the unsteady aerodynamics as well which results in extra – so called – aerodynamic lag states. The complete aeroelastic model was created with connecting the structural dynamics and the aerodynamics using splining. The details of the modelling can be found in [2]. Since the aeroelastic model is of very high order, first model order reduction needs to be carried out. The model order reduction is based on the bottom-up modelling approach and the details are given in [14]-[15].

The rigid body motion is represented with a 6-DOF model with 12 states: states of translational and angular velocities, position, and orientation. All of these are defined in the body coordinate system of the aircraft.

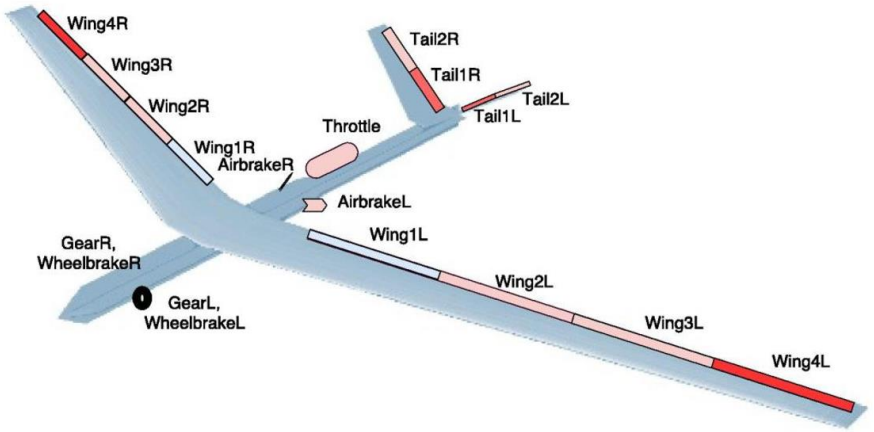
$$x_{rigid} = [u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi\ x\ y\ z]^T \tag{1}$$

The states which describe the flexible dynamics are the modal coordinates and their first derivatives. Due to the reduced order modelling only the 6 most significant modal coordinates and 2 aerodynamic lag states were considered.

$$x_{flex} = \left[U_{f1}\ U_{f2}\ U_{f3}\ U_{f4}\ U_{f5}\ U_{f6}\ \dot{U}_{f1}\ \dot{U}_{f2}\ \dot{U}_{f3}\ \dot{U}_{f4}\ \dot{U}_{f5}\ \dot{U}_{f6}\ lag_1\ lag_2\right] \tag{2}$$

The T-Flex aircraft model has 19 inputs (2. Figure): 2 landing gears (*GearR/L*), 2 landing gear wheel brakes (*WheelbrakeR/L*), 2 airbrakes located on the aircraft's fuselage (*AirbrakeR/L*) and 1 turbofan engine (*Throttle*). The demonstrator has 12 control surfaces: 4-4 ailerons

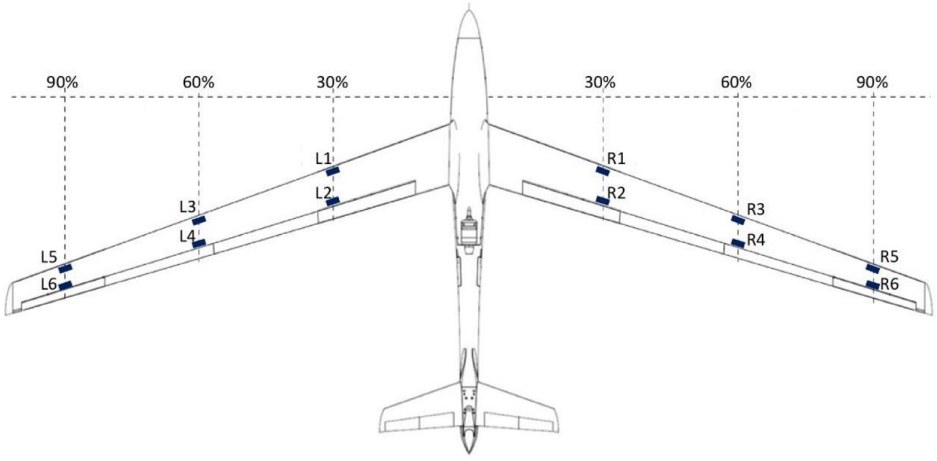
(*AileronR/L*) on each wing and on the V-tail 2-2 '*ruddervators*'. From the inputs, the landing gear related ones are insignificant in our research since the estimation of the structural dynamics only conducted during airborne operations.



2. Figure – Demonstrator input and control surfaces

The model has 23 rigid body related outputs, which provide information about the aircraft's position (x_E, y_E, z_E), orientation (Φ, Θ, ψ), translational (v_N, v_E, v_D) and angular velocity (p, q, r), and acceleration (a_{xB}, a_{yB}, a_{zB}). Furthermore, the course angle (χ), angle of attack (α), sideslip angle (β), air (p_a) and total pressure (p_T), barometric altitude (h_{baro}), indicated (v_{IAS}) and the true airspeeds (v_{TAS}) are measured as well.

Each wing of the demonstrator has 6-6 inertial measurement units (IMUs). An IMU provides acceleration and angular velocity data around the x-, y- and z-axis of its coordinate system. The IMUs communicate with the flight control computer (FCC) via CAN. However, as the bandwidth of the CAN-bus is limited and other sensors use the CAN as well, one IMU can only provide 3 measurement data. We opted for such an IMU configuration, where the IMUs on the leading-edge measure accelerations in the *x*, *y*, and *z* directions. The IMUs on the trailing-edge provide angular velocity data around the *x*- and *y*-axis, and acceleration data in the *z* direction. The exact location of the IMUs can be seen on 3. Figure. So, in total, the 12 IMU measurements add up as 36 additional outputs.



3. Figure - IMU configuration

In addition – in the light of the recent research conducted by the **Technical University of Munich (TUM)** – we assumed, that the wingtip coordinates can be measured with a mono

camera. On each wing, the coordinates of 4 wingtip points are measured in each direction which results 12 additional outputs.

3 MODEL BASED ESTIMATION OF FLEXIBLE DYNAMICS

3.1 LPV model

The linear parameter varying (LPV) model is an approximation to describe the behaviour of a nonlinear system [9]. It is essentially a pointwise linearization of a state-space system: the nonlinear system is linearized at different trim points. These trim points are defined by – so called – scheduling parameters. The scheduling parameters create a multidimensional grid, and a linear, state-space model is assigned to every grid point. The state-space description of a discrete time LPV system can be written as:

$$\begin{aligned} \underline{x}[k] &= \boldsymbol{A}(\rho[k])\underline{x}[k-1] + \boldsymbol{B}(\rho[k])\underline{u}[k] \\ \underline{y}[k] &= \boldsymbol{C}(\rho[k])\underline{x}[k] + \boldsymbol{D}(\rho[k])\underline{u}[k] \end{aligned} \tag{3}$$

where $\rho[k]$ is the time varying vector of the scheduling parameters.

In our work, we created an LPV approximation of the nonlinear bottom-up model of the T-Flex demonstrator aircraft with 2 scheduling parameters: the true airspeed (v_{TAS}) and the roll angle (Φ) sensor outputs. The grid for the LPV model consisted of airspeed values form *30 m/s* to *50 m/s* with a *1 m/s* resolution while the roll angles from *0°* to *40°* with *10°* resolution. Then the nonlinear model was trimmed at each grid-point. The resulting LPV model structure was then further refined to *0.1 m/s* and *1°* resolution with the spline interpolation method of the LPVTools MATLAB toolbox [16].

3.2 LPV-based Kalman filtering

For the model-based wing-shape estimation, an extended Kalman filter (EKF) was used. The EKF pipeline requires the full, nonlinear state-space description of the system as well as information about the model noise and observation noise in the form of noise covariance matrices. The nonlinear system's state-space representation in discrete time is written as:

$$\begin{aligned} \underline{x}[k] &= \underline{\boldsymbol{f}}\big(\underline{x}[k-1], \underline{u}[k]\big) + \underline{w}[k] \\ \underline{y}[k] &= \underline{\boldsymbol{h}}\big(\underline{x}[k], \underline{u}[k]\big) + \underline{v}[k] \end{aligned} \tag{4}$$

where $\underline{x}[k]$ is the state vector, $\underline{u}[k]$ is the input vector, $\underline{y}[k]$ is the output vector at timestep k. Nonlinear function $\underline{\boldsymbol{f}}(.)$ is called state-transition function, while $\underline{\boldsymbol{h}}(.)$ is called state-observation function. The $\underline{w}[k]$ and $\underline{v}[k]$ vectors are the model noise and observation noise vectors respectively. However, the explicit mathematical description – the nonlinear state-transition and state-observation functions – of the T-Flex demonstrator was not available for us, therefore a unique approach was necessary for designing the EKF.

The general workings of the EKF consists of 2 main steps: prediction and update. In these steps, pointwise linearization is used to approximate the behaviour of the nonlinear system. More precisely the Jacobians of the nonlinear state-transition and state-observation functions are calculated to get the linear, state-space matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ at each timestep.

$$\boldsymbol{A}[k] = \frac{\partial \underline{\boldsymbol{f}}}{\partial \underline{x}}\left(\underline{x}[k-1], \underline{u}[k]\right) \quad \boldsymbol{B}[k] = \frac{\partial \underline{\boldsymbol{f}}}{\partial \underline{u}}\left(\underline{x}[k-1], \underline{u}[k]\right)$$
$$\boldsymbol{C}[k] = \frac{\partial \underline{\boldsymbol{h}}}{\partial \underline{x}}\left(\underline{x}[k-1], \underline{u}[k]\right) \quad \boldsymbol{D}[k] = \frac{\partial \underline{\boldsymbol{h}}}{\partial \underline{u}}\left(\underline{x}[k-1], \underline{u}[k]\right) \tag{5}$$

In the prediction step the prior state estimation and the prior state estimation covariance is calculated using the inputs of the current timestep and the estimations from the previous timestep.

$$\underline{\hat{x}}[k|k-1] = \underline{\boldsymbol{f}}(\underline{\hat{x}}[k-1|k-1], \underline{u}[k]) \tag{6}$$

$$\boldsymbol{P}[k|k-1] = \boldsymbol{A}[k]\boldsymbol{P}[k-1|k-1]\boldsymbol{A}[k]^{\boldsymbol{T}} + \boldsymbol{Q} \tag{7}$$

In the update step, first the so-called innovation and innovation covariance is calculated which is used directly to get the near-optimal Kalman gain. With the help of the Kalman gain, the posterior state vector and state prediction covariance is computed.

$$\underline{\tilde{y}}[k] = \underline{y}[k] - \underline{\boldsymbol{h}}\left(\underline{\hat{x}}[k|k-1], \underline{u}[k]\right) \tag{8}$$

$$\boldsymbol{S}[k] = \boldsymbol{C}[k]\boldsymbol{P}[k|k-1]\boldsymbol{C}[k]^T + \boldsymbol{R} \tag{9}$$

$$\boldsymbol{K}[k] = \boldsymbol{P}[k|k-1]\boldsymbol{C}[k]^T\boldsymbol{S}[k]^{-1} \tag{10}$$

$$\underline{\hat{x}}[k|k] = \underline{\hat{x}}[k|k-1] + \boldsymbol{K}[k]\underline{\tilde{y}}[k] \tag{11}$$

$$\boldsymbol{P}[k|k] = (\boldsymbol{I} - \boldsymbol{K}[k]\boldsymbol{C}[k])\boldsymbol{P}[k|k-1] \tag{12}$$

In the equations, the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are the model and the observation noise covariance matrices respectively.

To obtain an appropriate pointwise linearization we used our LPV model. During simulation the true airspeed and roll angle is measured at each timestep which then can be used to select an approximating linear system from the LPV model. The selected model's state-space matrices are fed to the EKF as the current $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ matrices. Then the EKF conducts the prediction and update steps. Acquiring model and observation noise covariance matrices the following was done. Both the nonlinear and the LPV model was simulated with doublet inputs on the control surfaces and then the measured outputs and states where compared. The variance of the output and state differences were taken between the models and ordered into observation and model noise (co)variance matrices respectively. In the observation covariance matrix, the T-Flex's onboard sensors' noise variances were incorporated as well. These were specified based on the sensors' datasheets. Note that we used the assumption that both noises have 0 mean, normal distributions, and the noise vectors at each timestep are mutually independent.

4 DATA-DRIVEN ESTIMATION OF FLEXIBLE DYNAMICS

4.1 KalmanNet architecture

The other approach for estimating the flexible dynamics of the demonstrator is to use artificial intelligence, more precisely a neural network. Our choice was to use the relatively new KalmanNet architecture. [13]

KalmanNet – as its name suggests – combines Kalman filtering with a neural network. It still uses the current inputs and observations for giving state estimations, however the near optimal Kalman gain is provided by a trained recurrent neural network. The main advantage of this is that KalmanNet does not require neither the model (***Q***) nor the observation noise covariance matrices (***R***) and it can effectively overcome any uncertainties or errors in the model of the dynamic system.

The KalmanNet pipeline is the following. It still consists of a prediction and an update step just like a Kalman filter. In the prediction step however only the prior state prediction is calculated, the state prediction covariance (***P***) is not.

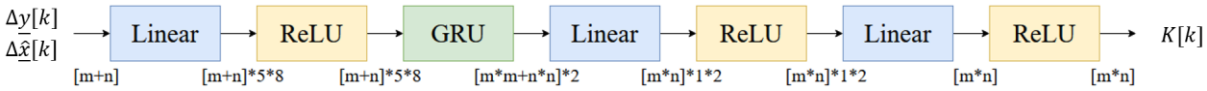
$$\underline{\hat{x}}[k|k-1] = \boldsymbol{A}[k]\underline{\hat{x}}[k-1|k-1] + \boldsymbol{B}[k]\underline{u}[k] \tag{13}$$

In the update step, first the innovation difference (*Δy[k]*) and the forward update difference (*Δx[k]*) are computed:

$$\Delta\underline{y}[k] = \underline{y}[k] - \underline{\hat{y}}[k|k-1] \tag{14}$$

$$\Delta\underline{\hat{x}}[k] = \underline{\hat{x}}[k-1|k-1] - \underline{\hat{x}}[k-1|k-2] \tag{15}$$

These act as the input features for the recurrent neural network. The RNN uses Fully Connected Layers with Rectified Linear Units (ReLU) and a Gated Recurrent Unit (GRU) to provide the actual Kalman gain. The network architecture can be seen on 4. Figure with each layer's input and output dimensions where *m* denotes the number of states (in our case *50*) and *n* denotes the number of outputs/observations (in our case *64*).



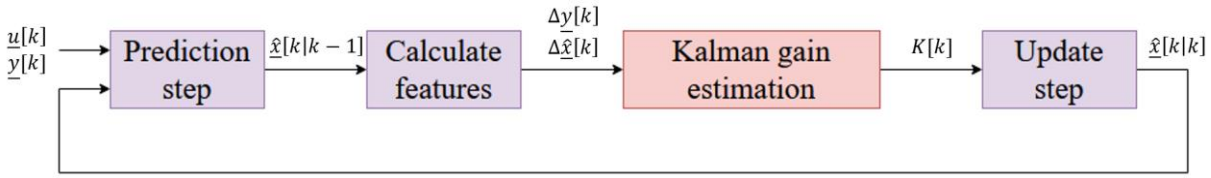
4. Figure - KalmanNet architecture

With the Kalman gain and using the innovation, the *a posteriori* state prediction vector is calculated.

$$\underline{\tilde{y}}[k] = \underline{y}[k] - \boldsymbol{C}[k]\underline{\hat{x}}[k|k-1] + \boldsymbol{D}[k]\underline{u}[k] \tag{16}$$

$$\underline{\hat{x}}[k|k] = \underline{\hat{x}}[k|k-1] + \boldsymbol{K}[k]\underline{\tilde{y}}[k] \tag{17}$$

The whole pipeline for the KalmanNet is presented at 5. Figure.



5. Figure - KalmanNet pipeline

As it can be seen, since neither the state estimation covariance (***P***) nor the innovation covariance (***S***) is used, the noise covariance matrices are not required. The whole pipeline works without any information about the model or observation noises.

4.2 Training parameters

The hyperparameters for training were set with the following values. The learning rate value was $5*10^{-7}$ with 'on plateau' learning rate scheduler. This method reduces the learning rate with a predefined factor if the validation accuracy hasn't changed for a given number of 'patience' epochs. The reduction factor was set to *0.5* and the 'patience' epoch number was *3*. The chosen optimizer for training was the ADAM algorithm with a weight decay value of 10^{-5}. The prediction accuracy was calculated with mean squared error (MSE) function. However – although the linearized aircraft model is a stable system, the system's poles are relatively close to the unstable region. So, a stability criterion was added to the MSE loss function.
It is possible to describe the complex system of the aircraft model joined with the Kalman filter with an error system:

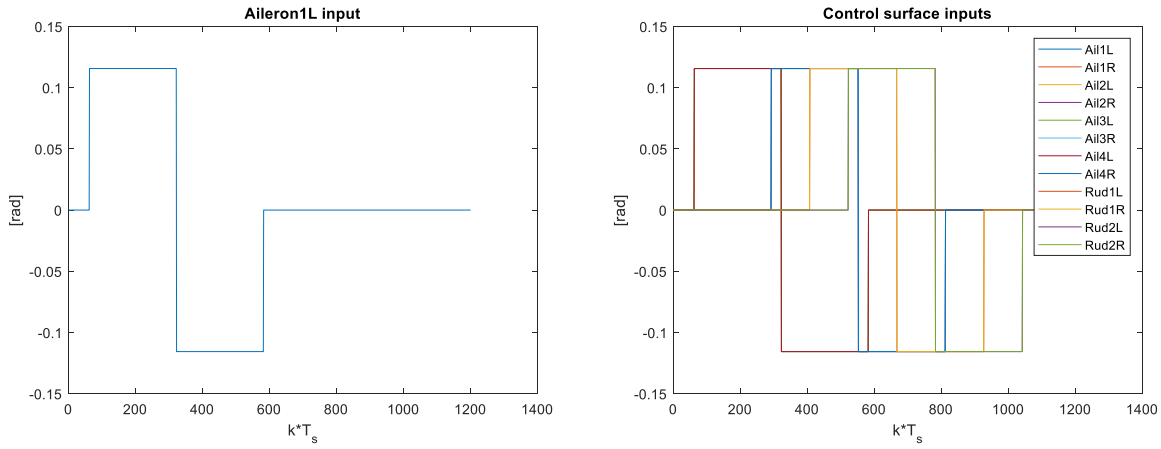
$$\underline{e}[k+1] = (\boldsymbol{A} - \boldsymbol{KC})\underline{e}[k], \tag{18}$$

where $\boldsymbol{K}$ is the Kalman gain, *e[k]* is the state prediction difference at timestep *k*. If the error system's state transition matrix ($\boldsymbol{A}$-$\boldsymbol{KC}$) has any unstable poles, then the whole system is unstable. So, the MSE loss was extended with the distance of the error system poles from the boundary of stability if it is larger than 0, thus making the loss value larger if the computed Kalman gain results an unstable error system.
As for weights initialization, both the linear layers' and the GRU cell's weights were initialized with *0* mean, 10^{-5} standard deviation normal distribution. The reason for having such a small standard deviation is the fact that if the initialized weights of the layers are too large, the newly initialized network produces such Kalman gains that makes the whole system so unstable, that the gradients explode because of the huge prediction error values.

5 RESULTS

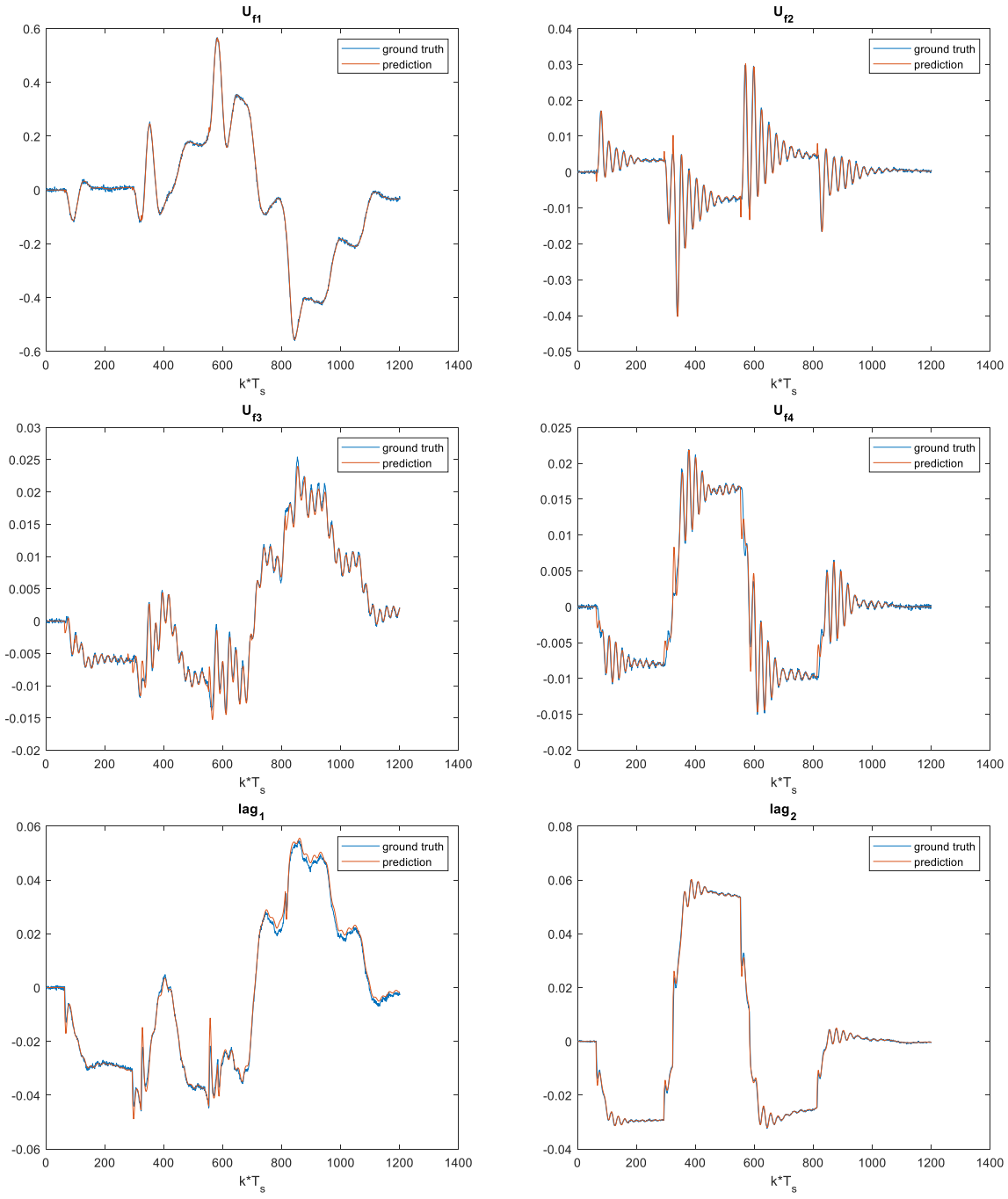
5.1 LPV-based EKF

The behaviour of the LPV-based EKF was tested with 2 different control signal configurations. The first configuration was 2.6 seconds long square-wave doublet inputs on each control surface with an amplitude of *6.6°* (*0.12 rad*) while there was a 3-second-long throttle input with an amplitude of *10%*, which accelerated the aircraft. For constructing the LPV model, the linear models were discretized with $T_s = 5\ ms$ sampling time. The used control surface inputs and a single control surface input is shown at 6. Figure.



6. Figure – Single aileron input (left) and all control surface inputs (right)

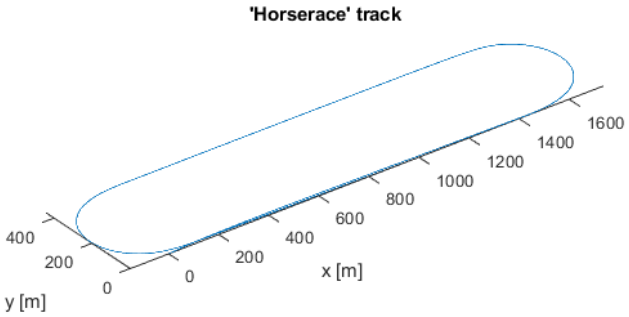
The simulation lasted for 6 seconds which adds up to 1200 samples in total. The initial flight conditions of the aircraft model were *42 m/s* true airspeed at *800 m* altitude with an initial course angle of *2°*. The initial vertical speed and the roll angle were set to 0.
The results for the doublet inputs are shown in 7. Figure, where the data with the ground truth label show the states of the nonlinear model, while the predictions show the states estimated by the filter. Since the main purpose of the observer design is to observe the flexible dynamics of the states, only the results for these states are presented. The first 4 modal coordinates are plotted where U_{f1} is the 1st symmetric bending and U_{f2} the 1st asymmetric bending mode. U_{f3} denotes the 1st symmetric torsion mode and U_{f4} is the 1st asymmetric torsion mode. The 2 aerodynamic lag states are plotted as well.



7. Figure - LPV-based EKF with doublet inputs

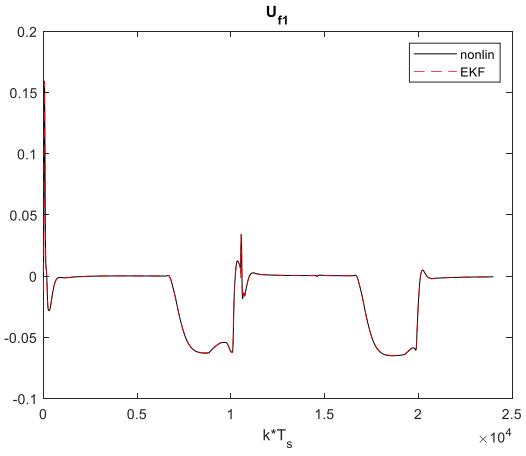
It can be seen, that the LPV-based EKF provides good estimations on the considered state dynamics. However, it is important to highlight that the predictions of U_{f4} and lag_1 have inaccuracies and small spikes can be seen around those timesteps where the doublet, square-wave signal was given on a control surface.

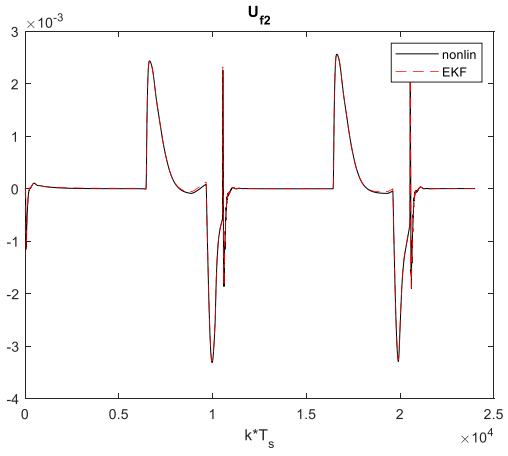
The second input configuration was provided by a baseline controller which keeps the aircraft at an oval – the so-called 'horserace' – track (8. Figure).

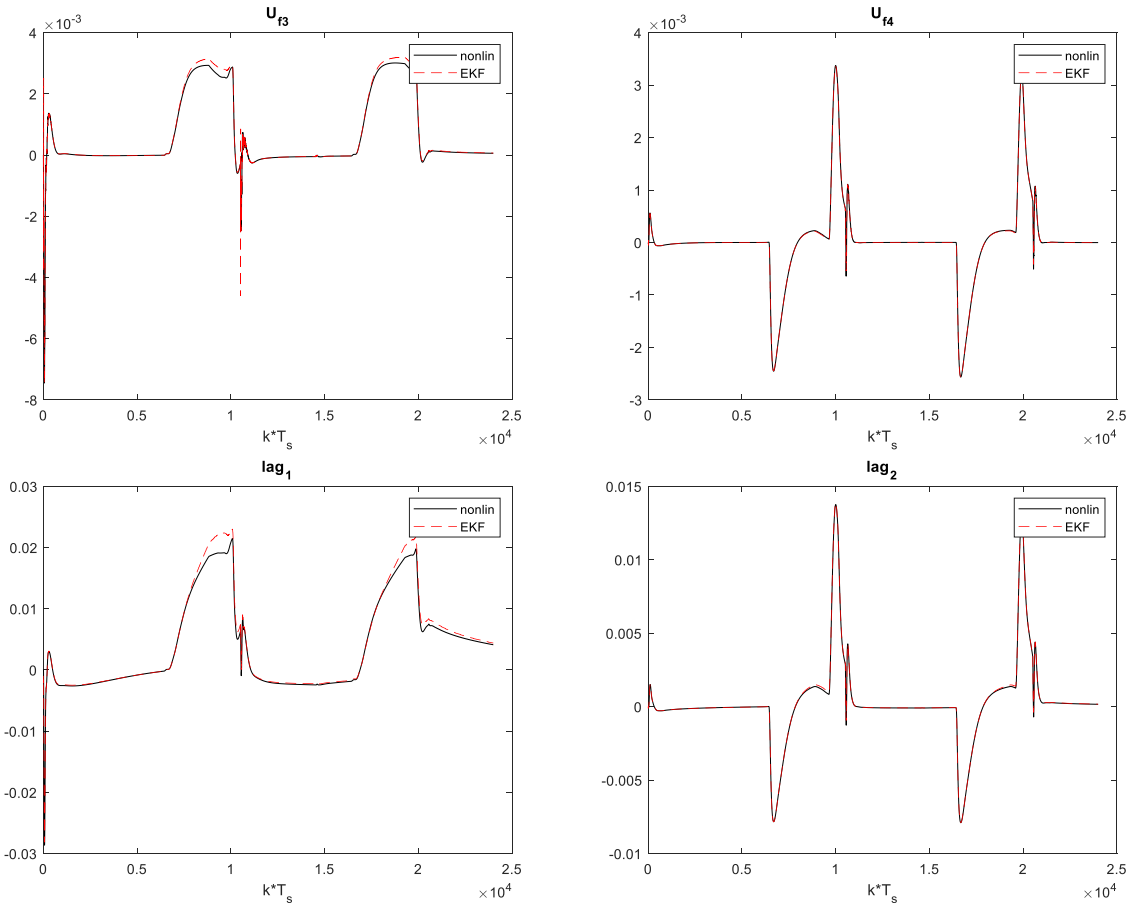


8. Figure - Horserace track

This controller is used during real-life flight tests on the T-Flex demonstrator. The LPV model was specifically designed for this application that is why the roll angle was selected as a scheduling parameter in addition to the true airspeed since during a turning manoeuvre the aircraft rotates around its longitudinal axis.
The initial conditions were the same as in the previous case: *42 m/s* flight speed at *800 m* altitude, with *2°* course angle. The whole simulation lasted for 120 seconds which corresponds to 1 full lap around the track.







9. Figure - LPV-based EKF with baseline controller

From the results (7. Figure) it can be concluded that the designed filter accurately predicts the modal coordinates and the aerodynamic lag states. Minor errors occur only during turning manoeuvres in U_{f3} and lag_1 states. The reason behind these is that the LPV model is still just an approximation of the real, nonlinear system. However, these inaccuracies are inside the error tolerance for this problem.

5.2 KalmanNet

For training, 20 batches of 1200 sample long trajectories were created. For generating the training, validation and test data, the reduced, nonlinear model of the T-Flex was used in Simulink. The inputs for the neural network were the observations and control surface and throttle inputs of the nonlinear model. The target for the network were the nonlinear model's states.

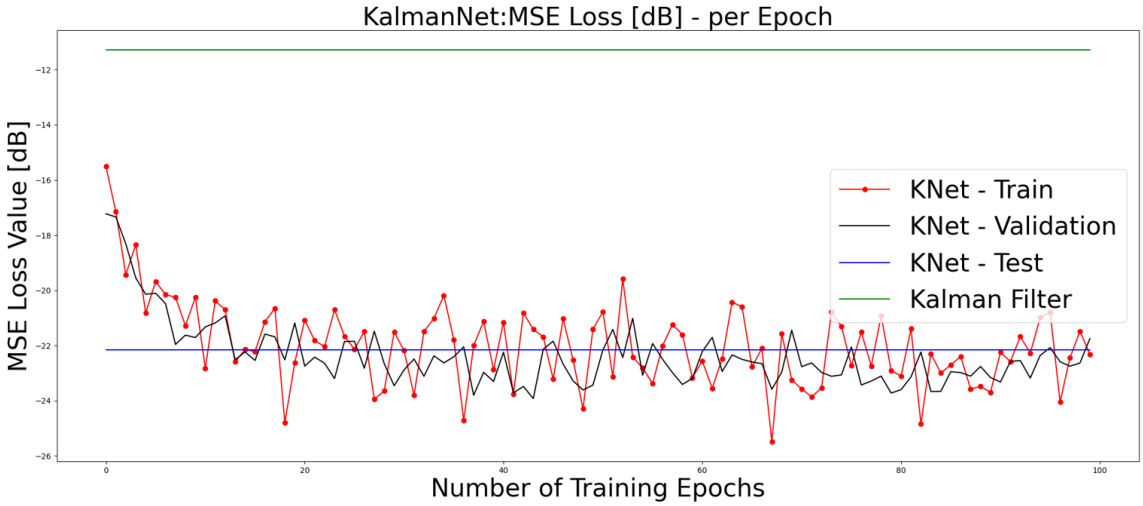
During training, validation and testing the KalmanNet used the linear, state-space matrices of the nonlinear model trimmed at *42 m/s* true airspeed at *800 m* altitude with initial course angle of *2°*.

In each trajectory, similar doublet and throttle inputs were given to the model as in the case of the LPV model. However, the doublet amplitudes were randomly generated between [*6°*; *10°*], the length of a doublet from [*1.5 s*; *2 s*]. The exact start time of a doublet was also randomly picked to make the training, validation, and testing data more diverse. Also, observation and model noise were also incorporated into the data. The noise samples were taken from random normal distributions with 0 mean and $\boldsymbol{R}$ and $\boldsymbol{Q}$ covariance matrices respectively.

The whole training procedure lasted for 100 epochs. In each epoch 5-5 batches were randomly selected from the total 20 for training and validation. The error metrics were defined in decibels for the sake of convenience during plotting, because the freshly initialized network tends to produce large errors. It is simply calculated with the following formula:

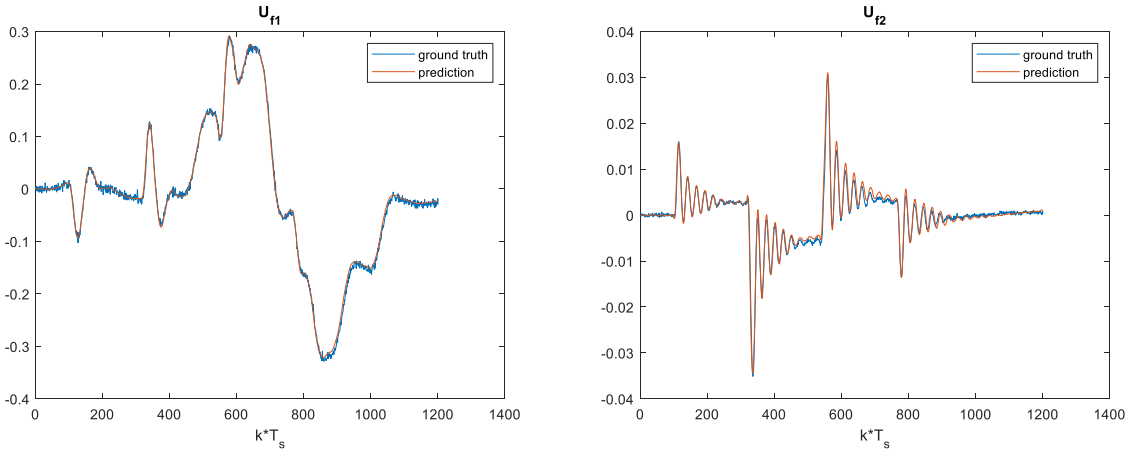
$$loss_{MSE}^{dB} = 10\log_{10}(loss_{MSE}) \tag{19}$$

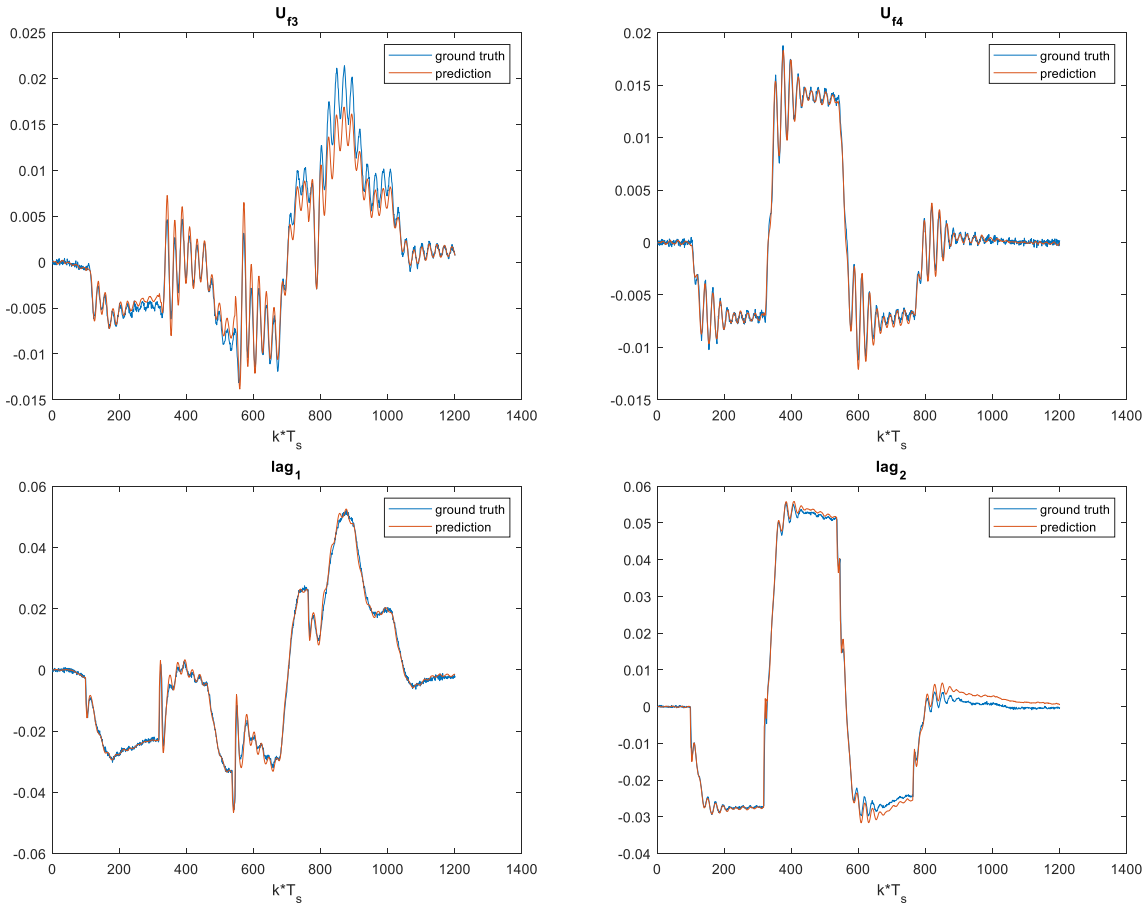
Before the training procedure, a simple Kalman filter was designed to the linear state-space model used by the KalmanNet as a reference. The summary of the training is presented at 10. Figure.



10. Figure - Training graph (Kalman filter included for reference)

The initial flight conditions for testing were set to *42 m/s* true airspeed at *800 m* altitude with initial course angle of *2°* in the T-Flex nonlinear model. The results are shown in 11. Figure





11. Figure - KalmanNet predictions with doublet inputs

The results indicate the following: the network manages to give better predictions in the case of U_{f1}, U_{f4} and lag_1 than the LPV-based filter. However, in the case of the other states, the accuracy of the predictions is worse than the EKF. The reason for this is that the KalmanNet only uses the linear state space model, corresponding to the initial flight conditions, but as the inputs are given on the control surfaces, the behaviour of the nonlinear model starts to differ more and more form the initial linear model. However, even with this handicap, it still manages to perform better than a standard Kalman filter designed for the initial flight condition's linear model *-23 dB* test error compared to *-11.3 dB* for the Kalman filter. This suggests that it can alleviate some of the inaccuracies caused by the discrepancies between the linearized and the nonlinear model.

The training of the neural network was carried out with a *Nvidia Tesla V100* GPU with *32 GB* RAM. Using this setup, the training lasted for approximately 4 - 4.5 hours.
Considering the results, we can conclude that only using a simple Kalman filter fitted to a linearized/trimmed model is a suboptimal solution to accurately predict the flexible dynamics of the nonlinear system. However, using an LPV-based EKF can provide good estimations. The use of the new KalmanNet architecture shows that even when it only has access to the linearized model while seeing data from the nonlinear model, it can overcome some of the inaccuracies coming from the differences between the models while not using any information about these model uncertainties. In order to give quantitative comparison of the performance of the 2 proposed architecture estimating the modal coordinates and lag states the previously mentioned logarithmic error metric was used. This is presented in 1. Table.

1. Table: Prediction errors

Description	$loss^{dB}_{MSE}$, dB
Kalman filter (*as a reference*)	-11.3
LPV-based EKF	-29.5
KalmanNet	-23

In case of the KalmanNet-based flexible dynamics estimation further research is needed in which the network will use the LPV model as the representation of the dynamic system.

6 CONCLUSION

To summarize, in this work we propose a model-based and a data-driven approach to estimate the flexible dynamics of a UAV with large wingspan and highly flexible wings. The model-based approach uses an LPV-based EKF while the data-driven solution utilizes the KalmanNet architecture. We show that the EKF-based estimator is able to predict the flexible and aerodynamic lag states both for doublet 'test' control surface inputs and for baseline controller inputs. The neural network-based approach is also capable of estimating the above-mentioned states, however, to obtain better, more accurate results, further research is needed. Our long-term goals include extending the KalmanNet architecture with the LPV system model to have a better approximation of the original, nonlinear system and conduct training with this modification. It is also a prospective goal to test both architectures in real life flight data and then incorporate them in the T-Flex's FCC for real-time, airborne operations. With this, it will be possible to design a wing shape controller to minimize aerodynamic drag during flights.

7 ACKNOWLEDGEMENT

The research leading to these results is part of the FLiPASED project. This project has received funding from the Horizon 2020 research and innovation programme of the European Union under grant agreement No 815058.

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

Supported by the ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

8 REFERENCES

[1] FLEXOP, "*Flutter Free FLight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP)*," Project of the European Union, Project ID: 636307, 2015-2018.

[2] Wüstenhagen, M., Kier, T., Pusch, M., Ossmann, D., Meddaikar M. Y., Hermanutz, A. (2018). *Aeroservoelastic Modeling and Analysis of a Highly Flexible Flutter Demonstrator,* AIAA AVIATION Forum, Atmospheric Flight Mechanics Conference June 25-29, 2018, Atlanta, Georgia

[3] Robles, R. R. (2022). *A Novel Adaptive Aeroservoelastic Coupling Supprerssion Algorithm for the A330 SMART MRTT Flying Boom Control Laws*, Conference on Guidance, Navigation and Control, 2022

[4] PAAW, *Performance Adaptive Aeroelastic Wing Program*, Supported by NASA NRA "Lightweight Adaptive Aeroelastic Wing for Enhanced Performance Across the Flight Envelope", 2014-2019.

[5] Patartics, B., Luspay, T., Péni, T., Takarics, B., Vanek, B., Kier, T. (2017). *Parameter varying flutter suppression control for the BAH jet transport wing*, International Federation of Automatic Control Conference, 2017.

[6] FLiPASED, "*Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FLiPASED)"*, Project of the European Union, Project ID: 815058, 2019-2022.

[7] Kotikalpudi, A., Danowsky, B. P., Schmidt, D. K., Gupta, A., Regan, C. (2019). *Real-Time Shape Estimation for a Small Flexible Flying-Wing Aircraft,* AIAA SciTech Forum, 7-11 January 2019, San Diego, California

[8] Lustosa, L. R., Kolmanovsky, I., Cesnik, C. E. S., Vetrano, F. (2021). *Aided Inertial Estimation of Wing Shape*. Journal of Guidance, Control and Dynamics, American Institute of Aeronautics and Astronautics, 2021, 44 (2), pp.210-219.

[9] Takarics, B., Vanek, B. (2021). *Robust Control Design for the FLEXOP Demonstrator Aircraft via Tensor Product Models, ASIAN JOURNAL OF CONTROL, 23 (3). pp. 1290-1300. ISSN 1561-8625*

[10] Shereen, M. K., Khan, M. I., Khan, N., Ullah, W. (2016). *By the Design and Implementation of Modified Kalman Filter for LPV Systems*, International Journal of Engineering Works, Vol. 3, Issue 4, PP. 26-31

[11] Dugne-Hennequin, Q. A., Hideaki, U., do Monte Lima, J. P. (2021). *Understanding the Behaviour of Data-Driven Inertial Odometry with Kinematics-Mimicking Deep Neural Network*, IEEE Access, vol. 9, pp. 36589-36619, 2021, doi: 10.1109/ACCESS.2021.3062817.

[12] Zhang, M., Zhang, M., Chen, Y., Li, M. (2021). *IMU Data Processing for Inertial Aided Navigation: A Recurrent Neural Network Based Approach,* arXiv: 2103.14286 v1, 26 Mar 2021

[13] Revach, G., Schlezinger, N., Ni, X., Escorzia, A. L., van Sloun, R. J. G., Eldar, Y. C. (2021). KalmanNet: *Neural Network Aided Kalman Filtering for Partially Known Dynamics*. arXiv 2107.10043 v1

[14] Takarics, B., Vanek, B., Kotikalpudi, A., Seiler, P. (2018). *Flight Control Oriented Bottom-up Nonlinear Modeling of Aeroelastic Vehicles,* 2018. IEEE Aerospace Conference

[15] M. Meddaikar et al. (2019). *Aircraft aeroservoelastic modelling of the FLEXOP unmanned flying demonstrator,* AIAA Scitech 2019 Forum, AIAA, 2019–1815.

[16] Balas, G., Packard, A., Seiler, P., Hjartarson, A. (2015) LPVTools - *A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems*, MUSYN Inc. 2015

COPYRIGHT STATEMENT