ADAPTIVE FLIGHT CONTROL FOR UNMANNED AERIAL VEHICLES USING A NEURAL NETWORK PREDICTOR

Thomas Krüger, Arne Krüger, Joachim Axmann, Peter Vörsmann
Institut für Luft- und Raumfahrtsysteme - Technische Universität Braunschweig
Hermann-Blenk-Str. 23, 38108 Braunschweig, Germany

Andreas Kuhn
Andata Development Technologies
Hallburgstr. 5, 5400 Hallein, Austria

Abstract

Adaptive flight control is an important element to improve the degree of automation of small unmanned aerial vehicles (UAV), especially regarding autonomous operation under difficult atmospheric conditions or even system failures. Machine learning techniques enable the UAV to improve control accuracy during operation and to respond to unknown, non-linear flight conditions. Here, artificial neural networks (ANN) are used to implement such an intelligent flight control system with offline-trained basic knowledge which can be further improved during flight. The online learning step is realised with a controller architecture comprising a neural network controller and a neural observer which predicts the system's dynamics and delivers the critics signal for contoller adaption. To do so, groups of neural controllers and predictors are trained with basic knowledge of the required behaviour, derived from measured data. Afterwards, the basic capabilities are expanded by online network optimisation during flight. The training success of all networks regarding generalisation capabilities and robustness of the basic knowledge is evaluated with statistical methods. The offline training phase showed that the necessary learning task can be achieved by multi-layered feedforward-networks, while the simulation of the implemented neuro controllers showed robust behaviour. An important element of the control strategy is to determine a consistant error signal for online learning of the neural controller. This is done by backpropagation of a measured deviation from the target trajectory through the predictor networks. In summary, the statisic analysis of the robustness of the basic knowledge as well as the implementation of a neural predictor in the control process proved to be central aspects of the control strategy.

Nomenclature

a	=	spline coefficent for x-coordinate
b	=	spline coefficent for y-coordinate
b_j	=	bias value of neuron j
d	=	measured deviation from the spline
$\hat{d}$	=	predicted deviation from the spline
E_j	=	squared error of the network output
f	=	transfer function of a neuron
$\mathbf{J}$	=	Jacobian matrix of the network weights
n	=	argument of the transfer function
p	=	measured roll rate
r	=	measured yaw rate
t	=	parameter for the position on the spline
t_j	=	target of neuron j
u_j	=	output signal of neuron j
w_{ij}	=	weight from neuron i to j
x	=	x-coordinate in the geodetic system
y	=	y-coordinate in the geodetic system
δ_j	=	error signal of a neuron
μ	=	learning rate
ξ_c	=	control signal of the ailerons
ϕ_c	=	pre-filtered bank angle from spline-geometry
ϕ	=	measured bank angle
χ_c	=	pre-filtered heading from spline-geometry
χ	=	measured heading

1 Introduction

Since 2001 one focus of research of the Insitute of Aerospace Systems at the Technische Universität Braunschweig is flight control, sensor systems and navigation for small UAVs. The result of the initial project phase was the *CAROLO P50* - an aircraft with 50 cm wing span and a take-off mass of about 500 g. Since the payload capacity of a micro aerial vehicle is limited for scientific applications, other small UAVs have been developed for different payloads and missions. An overview of all developed planes, their special hardware and the on-board navigation system can be found in [1]. Besides the improvement of the aircraft and their subsystems, a focus of work is the scientific application in the fields of meteorological measurements [2], [3], remote sensing for environmentnal monitoring [4] and civil protection. The systems used for these applications are the *CAROLO T200* and *P200* whereas the *P200* is depicted in Fig. 1. The aircraft can be equipped for automatic photography missions or with a payload of miniaturised meterological sensors. The *T200/P200* has a take-off mass of about 5 kg, a payload of 1 kg and a wing span of 200 cm; the electric propulsion system allows a flight duration of up to 60 minutes at a speed of about 20 m/s. The results for the adaptive flight control strategy presented here have been gathered using the *CAROLO P200* and the corresponding simulation environment as testbed. This simulation applies the non-linear equations of motion, measured actuator and sensor models and an atmospheric model with wind and simplified turbulence to enable adequate realistic behaviour.

Figure 1: The mini UAV *CAROLO P200* used for automated aerial photography during landing approach.

During the last years many unmanned aerial systems have achieved a degree of automation adequate for numerous possible applications. But still there are certain aspects which can highly improve the capabilities of these aircraft, especially concerning micro-sensors, GPS/INS (Inertial Navigation System) data fusion and flight control. Besides the enhancement of classical control strategies, machine learning techniques are used more often to implement adaptive flight control systems. In addition, ANN-strategies have been developed to improve onboard GPS/INS data fusion algorithms [5]. There are different methods under research how to construct intelligent control systems like fuzzy logic, neural networks or combinations of both approaches; the learning task, which is an optimisation problem, can be solved with gradient descent methods like the backpropagation algorithm or, for example, with evolutionary algorithms, stochastic methods using elements of biological evolution ([6], [7], [8], [9]). Since the stability of control systems based on artificial intelligence (AI) is not easily proved, a combination of classical and adaptive controllers might be possible as proposed in [10]. There, the neuro-controller is acting as an add-on to the standard control loop, so that a stable basic function is improved by an ANN-controller.

When using artificial intelligence for flight control of fully automatic UAVs, it is feasible to see the aircraft as an intelligent agent, as it is often done in robotics [11]. The idealised agent is capable of perceiving its environment, reacting according to the task (here the flight plan of the mission), learning from experience and operating with sustained stable performance. While precise perception of the surrounding area is a sensory challenge, the agent's remaining characteristics could be achieved using a neural controller which adapts its knowledge according to stable online critics. The *P200's* onboard sensors measure the rates for pitch, roll and yaw, the dynamic and barometric pressure, the accelerations in all spatial directions; a GPS receiver provides geographic longitude and latitude as well as the altitude. The *CAROLO* UAVs utilise a tighly coupled Kalman-filter for precise attitude determination as decribed in [12] and [13]. This aircraft uses only elevator and aileron, no rudder is applied for lateral control.

This paper analyses how far a neural network approach may contribute to such an adaptive flight control system for unmanned drones. Additionally, it will be outlined that neural flight control is generally able to avoid the time-consuming work of controller tuning and optimisation by teaching the necessary basic abilities offline. Because of the *black box* characteristics of ANNs permanent robust behaviour during operation is hard to prove. Therefore, a new approach is made to verify the abilities adopted during the learning phase. The training data is presented to a multitude of feedforward-networks with varying architectures (number of layers and neurons) to ensure that the training success of a single network-type is not random. Subsequently, the training results of all networks are analysed statistically as presented in [14]; the results emphasise that not empiric network-topology optimisation but rather the selection of representative training data is essential for training success. After the substantial analysis of the basic knowledge, the validated controllers and predictors are implemented into the online-learning simulation environment where the improvement of the flight control characteristics is demonstrated. Here, an important element is the calculation of an error signal for the neural controller; this is gained by backpropagating the measured deviation from the target trajectory through the ANN-predictor trained with the inverse dynamics of the aircraft. In this article the results for spline-based lateral movement control are presented.

2 Spline-based Trajectories and Neural Control Approach

The neural control strategy is based on the existent control system for the *CAROLO T200/P200* mini UAV [1]. The lateral and horizontal movement is controlled by two cascaded control loops consisting of a combination of PD and PI elements for control of elevator and aileron. In the innermost loop there is a damper to reduce high-frequency oscillation, followed by a basic controller and the autopilot in the outermost loop to control the flight path accuracy. A notable improvement of the system's performance is realised by implementing the commanded trajectory on cubic B*é*zier-splines which are interlinked with each other to create the flight path of a mission [15]. An example is shown in Fig. 2.

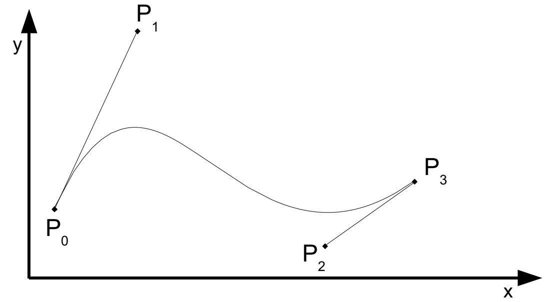

Figure 2: Example of a cubic spline used to define the flight trajectory by interlinking several curves

These splines are defined by four points (P_0 to P_3) given in the geodetic coordinate system and can be determined by:

$$x(t) = a_3 \cdot t^3 + a_2 \cdot t^2 + a_1 \cdot t + x_0 \tag{1}$$

$$y(t) = b_3 \cdot t^3 + b_2 \cdot t^2 + b_1 \cdot t + y_0 \tag{2}$$

where t is the parameter defining the position on the spline ($0 \leq t \leq 1$) while $a_1 = 3(x_1 - x_0)$, $a_2 = 3(x_0 - 2x_1 + x_2)$ and $a_3 = -x_0 + 3x_1 - 3x_2 + x_3$. The coefficents $b_1 - b_3$ from (2) are calculated likewise simply using $y_0 - y_3$. The two main advantages of this method are: Firstly, the desired flight attitude can be determined beforehand for every point on the trajectory, since the desired roll-angle is a function of the velocity and bending of the spline. Secondly, the deviation from the spline, as a direct measure of lateral trajectory tracing, can be measured on every control step. To do so, the actual position derived from the GPS/INS data fusion of the Kalman-filter is compared with the target postion on the spline. This information is crucial for the calculation of the training signal of the neural controller. These spline-based variables are also used for the neural controller and predictor to have more input data regarding the system's dynamics and, hence, more information describing the learning task.

Previous studies made clear that the specialisation of the neural networks on single tasks of the whole control loop is important, meaning that each ANN delivers only one output signal. To teach good basic knowledge to the controllers as well as the predictors, 500 training epochs with five to ten thousand training patterns were used representing the most important manoeuvres and distinctive flight attitudes. The generalisation abilities of these modules are expected to be reliable since ANNs are able to approximate complex functions accurately when supplied with sufficient training data. Besides input signals from sensors and spline geometry, time delays are added so that historical measurment data as well as recent output signals are available as network inputs [16]. This is observable in Fig. 3. This is done because the coherence of the inputs and outputs of a high-dynamic system like an aircraft is time dependent. Just the current inputs of a process are not sufficient to mirror the desired non-static behaviour. Therefore, a wider time frame is used as input to ensure that the time-dependent dynamic behaviour of the system is reproduced by the network. At least one time delay for all signals should be used, so that a kind of short term memory is implemented since the current output is also influenced by the provided historical data.

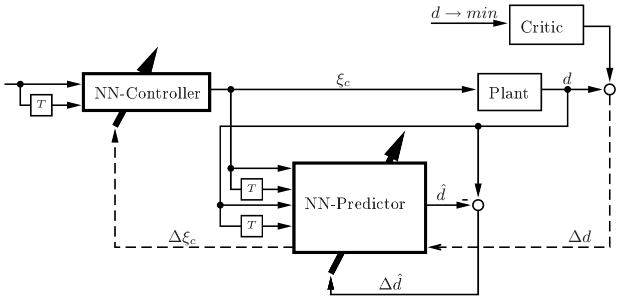

Figure 3: Principal design of the neural control strategy; the delayed inputs ensure the learning of the dynamic behaviour. The amount of delayed time-steps for the input signal is dependent on the complexity of the control problem.

To map the necessary dynamic characteristics in order to calculate the output $o_{NC} = \xi_c$ of the neural control unit, the input vector $\underline{x}$ consists of the following signals:

$$\underline{x}_{NC} = \begin{pmatrix} \xi_{c,T1} \\ \Phi_c \\ \Phi \\ d \\ \chi_c \\ \chi \\ p \\ r \end{pmatrix} \tag{3}$$

where $\xi_{c,T1}$ is the time-delayed controller output of the last control step, Φ_c and χ_c are pre-filtered control signals derived from the spline geometry and Φ, χ, p, r are the measured signals for bank angle, heading, roll rate and yaw rate. In addition, these signals are used with a time-delay as described above, so that 16 inputs are used for the neural controller unit to generate the control singal ξ_c.

Because no direct error signal can be calculated for the controller output ξ_c one main element for the control strategy depicted in Fig. 3 is the neural predictor. It is trained to learn the dynamic behaviour of the system by mapping sensor and control signals to the deviation from the desired spline trajectory. The input vector for the neural predictor is similar to the neural controller, whereas only time-delayed inputs are used to propagate the neural predictor output $o_{NP} = \hat{d}$. This is done to ensure that the measured value d and the predicted value $\hat{d}$ represent the same time step in order to calculate the correct training signal for the predictor network $\Delta\hat{d}$. Again every input has an additional time-delay so that 16 input signals are used for the predictor network.

$$\underline{x}_{NP} = \begin{pmatrix} \xi_{c,T1} \\ \Phi_{c,T1} \\ \Phi_{T1} \\ d_{T1} \\ \chi_{c,T1} \\ \chi_{T1} \\ p_{T1} \\ r_{T1} \end{pmatrix} \tag{4}$$

The permanent measurement of the spline-deviation d compared with $\hat{d}$ provides the direct quality signal $\Delta\hat{d}$ for predictor training. Since there is no direct quality signal for the controller, the inverse dynamic of the predictor is used to backpropagate the spline deviation, so that the error signal $\Delta\xi_c$ for online-training can be generated. Below, a possible procedure to calculate this error signal is presented.

The neural networks applied for the control architecture are standard feedforward-networks using a hyperbolic tangent transfer function and a linear function in the output layer respectively, often used for control purposes [17]. The transfer function is similar to the sigmoid function, but has a codomain ranging from -1 to 1:

$$f = tanh(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \tag{5}$$

where n is the overall argument of the transfer function. Since the knowledge of ANNs is stored in the connection weights they have to be optimised during the training. The training is divided into two phases - offline batch-training and online training during flight. For every training phase a variant of the backpropagation-algorithm, which utilises a gradient descent method, is used: the Levenberg-Marquardt algorithm for offline-training and the much faster standard backpropagation-method without momentum for online application; both algorithms are explicitely described in [18]. In general the forward propagation step of any neuron is calculated by:

$$u_j^L = f\left(\sum_{i=1}^{n} w_{ij}\cdot u_i^{L-1} + b_j\right) \tag{6}$$

where j indicates the neuron of layer L and i the n neurons of layer $L-1$; b is the bias of neuron j and f the fransfer function. If the target value t_j of a network output $o_j = u_j^L$ is known, the resulting squared error of the output signal, which is a measure for the quality of the network's performance can be calculated by:

$$E_j = \frac{1}{2}\left(o_j - t_j\right)^2. \tag{7}$$

Now, the change of the weights, which is the step size on the surface of the error function of one optimisation step, can be determined by the following equation for a gradient descent:

$$\Delta w_{ij} = -\mu\frac{\partial E_j}{\partial w_{ij}} = -\mu\mathbf{J} \tag{8}$$

with the learning rate $\mu > 0$ and the Jacobian matrix $\mathbf{J}$ of the error function E_j. In [18] the complete algorithm is deduced, also for hidden layers. Important is, that after a repeated application of the chain rule the Jacobian matrix can be calculated only with elements from the forward propagation step.

$$\mathbf{J} = \frac{\partial E_j}{\partial w_{ij}} = \delta_j\cdot u_i^{L-1}. \tag{9}$$

In equation (9) the backpropagated error δ_j for a neuron j in the output layer can be calculated by:

$$\delta_j = (o_j - t_j)\cdot f_L^{'}(w_{ij}\cdot u_i^{L-1}) \tag{10}$$

As depicted in Fig. 3 there is no possibility to directly measure the error of the control signal $\Delta\xi_c$. If the aileron signal ξ_c is the input j of the predictor network it is possible to calculate a corresponding backpropagated error for this input neuron δ_j^1 in correspondance to a measured spline deviation Δd. This backpropagated error signal characterises the change of the error function E_j with regard to an altered input signal ξ_c. Thus, if $\hat{d}$ is equal or nearly equal to d, Δd can be used to calculate the training signal $\Delta\xi$ for the controller training.

3 Basic Training Results

During the offline training phase the neural network learns the necessary basic knowledge offline so it can be implemented as a controller into a simulation or onboard. Concerning the network architecture it is not the goal to find one highly specialised ANN-topology, but to take groups of standard networks varying in size and demonstrate the quality of their overall performance. If all networks show good characteristics while learning the necessary basic knowledge, one has first evidence for stable behaviour. At this point, it is clear that the knowledge can only be as profound as the information contained in the training data, so the usage of appropriate training patterns is most important. Firstly, all data sets were obtained through simulation runs. Because of the promising training success, all training and testing was done with complex, asymmetric trajectories which had a flight duration of about five to ten minutes. These training trajectories take two things into account: the amount of training patterns is large enough for a well-grounded training and the variety of manoeuvres is sufficient for good generalisation abilities concerning unknown flight paths. An example for such a trajectory is shown in Fig. 4, the altitude change during flight is not depicted here because of the focus on lateral spline trajectory control.

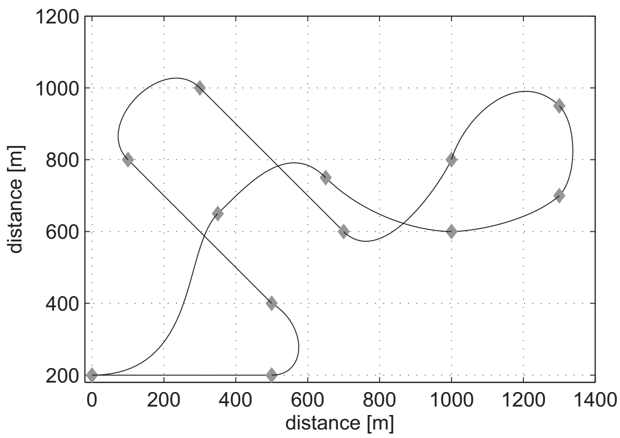

Figure 4: Example of a trajectory used to train the neural networks; start and ending point is at $x = 0$ m and $y = 200$ m. The dots mark the intersections between splines.

To ensure a good network reaction towards unknown situations all training data was divided into training, validation and test data (70%, 15%, 15%), whereas only the training

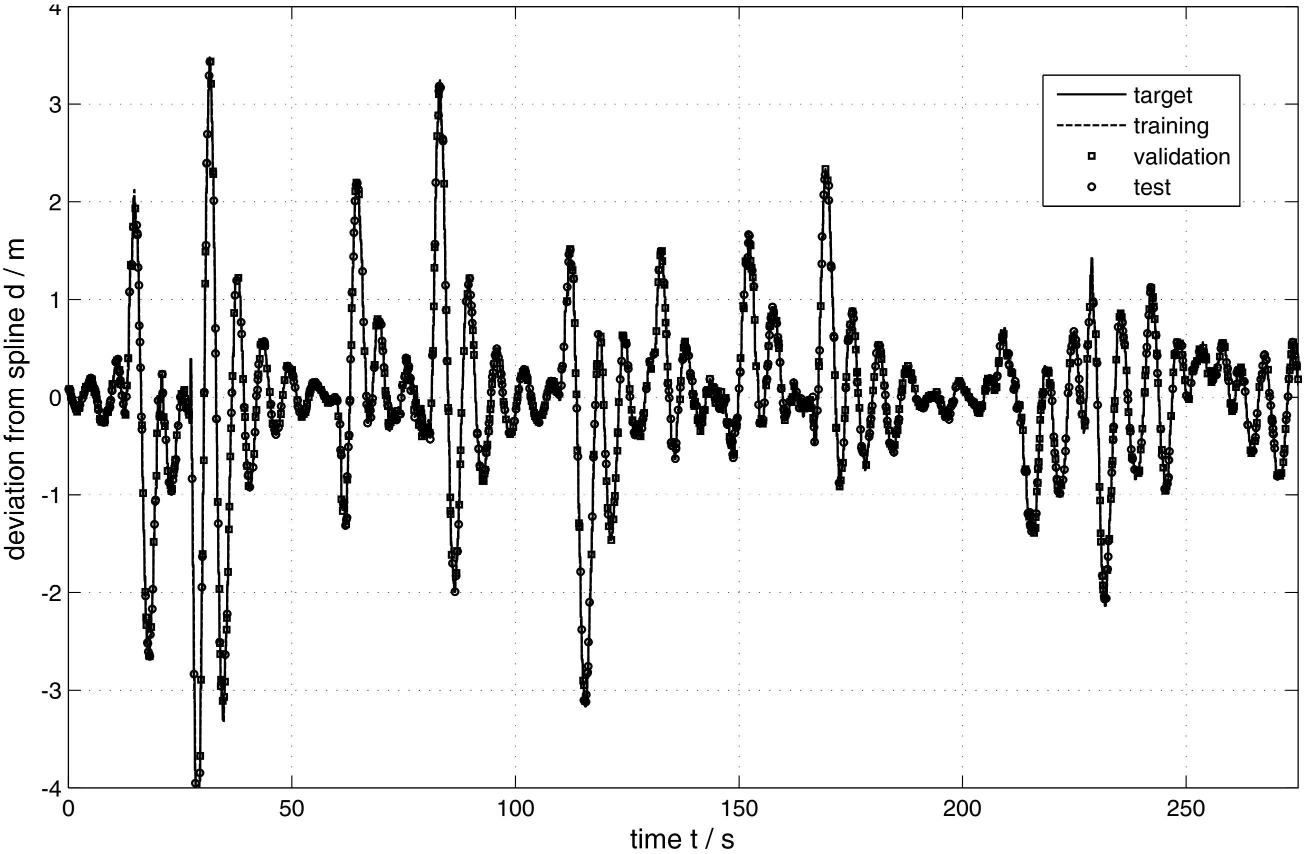

Figure 5: Network response for training, test and validation data. The outputs follow the targets very closely over the whole time frame of the trajectory. The results for validation and test data imply good generalisation capabilities.

error alters the connection weights. The initial connection weights are randomly assigned for every network. The allocation of the patterns to training, validation and test data is a random process for every training epoch and every network of the learning task. This guarantees statistical reprensentative results over all networks of a training cycle. The validation and test data is unknown to the network, whereas the training is stopped when the validation error does not decrease over time. This prevents unnecessary training and also over-fitting, meaning that the training specialises on certain parts of the data without improving the generalisation abilities, which are highly important for stable performance. Hence, the mean square error for validation and test is a first information concerning the generalisation capabilities of the neural controller/predictor. The ability of ANNs to approximate complex functions was observed when propagating the predicted deviation from the spline as depicted in Fig. 5. This is an example for a three-layered network with 30 neurons in the hidden layer which is representative for all training results. The offline-training of the control units showed very similar quality characteristics. Regarding the size of all controller and predictor networks maximal two hidden layers where used while the number of neurons for the whole ANN ranged from 20 to at most 60.

To underline the adequate selection of input parameters, network topology and optimisation methods, Fig. 6 shows the probability of the mean square error of the deviation d for all 60 trained networks. The plot also shows the results for test and validation data to ensure that the generalisation abilities of all networks are adequate. The statistical evaluation of the learning results with respect to all networks shows that the training success is not random for a single network. This gives important information wether the overall learning strategy is chosen appropriately. Apparently, the dynamics of the lateral movement can be learned by all neural networks and not only by the best ones. Since the correct mapping of the deviation from the spline is a precondition for succesful online training robust predictor characteristics are crucial.

Although all connnection weight changes are conducted offline, the trained neural controller and predictor elements are a good basis for test and simulation, since the results of the training phase are promising. Further, the offline-acquired knowledge is necessary when thinking about flight tests with the neuro-controlled UAV because this would be nearly impossible without basic abilities. In the next step the ANNs can be tested as controller/predictor units to verify the robustness of their behaviour. Afterwards, online-learning using the control deviation as quality criterion for the optimisation of the connection weights is tested. If the dynamics are learned appropriately by the predictor, one can backpropagate the resulting control deviation through the predictor to generate the corresponding error of the control signal. This error signal is then used to optimise the knowledge of the controller network which leads to a more accurate trajectory following.

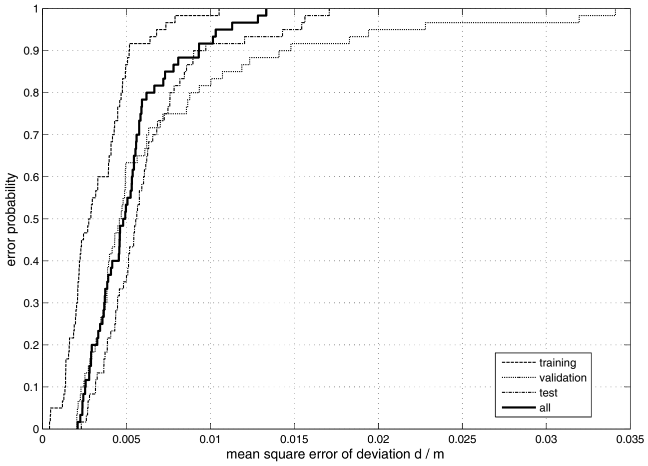

Figure 6: Probability of the mean square error of the spline deviation d for 60 trained predictor networks. All networks show similar stable performance for training, validation and test data.

4 Online Training Results

All tests of the neural controllers were done using the simulation environment of the *CAROLO* drone implemented in MATLAB/Simulink. All simulations were conducted over longer time frames (at least 300 seconds) including the modelling of wind and simplified turbulence, based on the Dryden spectrum. This is done to examine if the offline trained generalisation capabilities lead to robust flight characteristics considering limited atmospheric disturbences. At first, every neural control unit was tested with the trajectory it was trained for to see if the specialised knowledge for one trajectory can be recalled. The results of the ANN-controllers were continuously adequate taking into account that the neuro-controller just works with basic knowledge; in all tests the performance was about the same as the reference controller (they are not depicted here because the tests of unknown trajectories are more significant). Summarizing training and first tests, the choice of the input and output signals, the use of historical measurements for the output generation and the structure of the training patterns regarding the required information for the task seem appropriate. Subsequently, the neuro-controllers were tested with unknown flight paths to see if the desired generalisation capabilities and robust behaviour can be observed. The unknown trajectories have an average flight duration of about 500 seconds. According to the trained trajectory with a flight duration of about 300 seconds the chosen time for the untrained track should be adequate for testing the desired generalisation characteristics. After testing the basic knowledge for trained and unknown trajectories flights with online network adaption (for controller and predictor) were performed.

The correct functioning of the predictor is essential for the backpropagtion of the error signal needed for online controller optimisation. The robust basic knowledge shown in Fig. 5 is stable during operation which is depicted in Fig. 7 for a flight path not comprised in offline training. This trajectory is in principle similar to the one shown in Fig. 4, though it is much longer with a varied spline geometry. Beside a few peaks to an inaccuracy of about 0.8 metres the neural network is able to predict the lateral deviation from the trajectory very precisely leaving an error of 10 to 20 cm. Moreover, the trained knowledge is not lost with online predictor optimisation; this can be supported with a rather small initial learning rate μ to limit the optimisation step size. Taking into account measurement inaccuracies of the onboard sensors these prediction errors are very small and hence adequate for the flight path accuracy an UAV can achieve. The very small prediction errors show that the lateral system dynamics are robustly learned by the ANN and that the knowledge is not lost during longer operation.

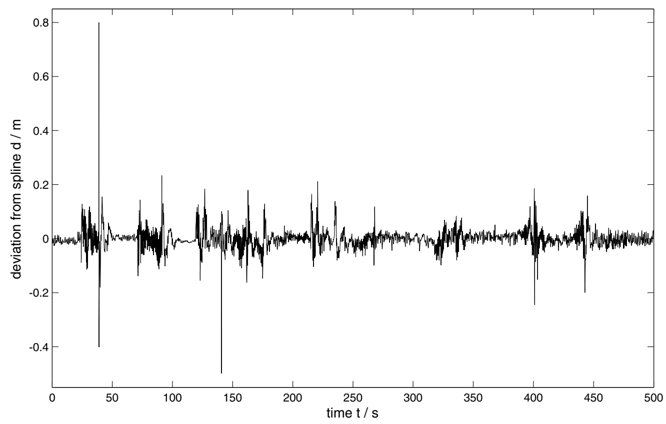

Figure 7: Prediction error for an untrained trajectory using further online optimisation of the predictor.

Fig. 8 shows the deviation from the target spline of the last section of the same, not offline trained, trajectory from Fig. 7. The results for a neural controller with and without online training are plotted to show the improvements due to online learning. The first 250 seconds are not shown because the difference between the basic ANN-controller and online learning is not very significant at first. This is the case because, again, a small learning rate is chosen, so that the optimisation step size is limited. The risk of knowlegde loss is low and the learning effects are stronger after some time of operation. The online learning phase uses the same neural predictor discussed before to backpropagate a quality signal for online learning. For the basic ANN-controller the shown deviation remains at this range for the whole trajectory, similar to the results from offline training depicted in Fig. 5 and underlines the earlier statement: the neural controller is able to recall the acquired knowledge in a generalised way since the overall spline deviation is in a absolute range of about 2,5 to 3 metres. It becomes apparent that the focus on adequate training data and the systematic analysis of the learning phase are very important for the quality of the basic ANN-controllers.

The challenging task during online learning is to preserve the generalisation capabilities of both predictor and controller

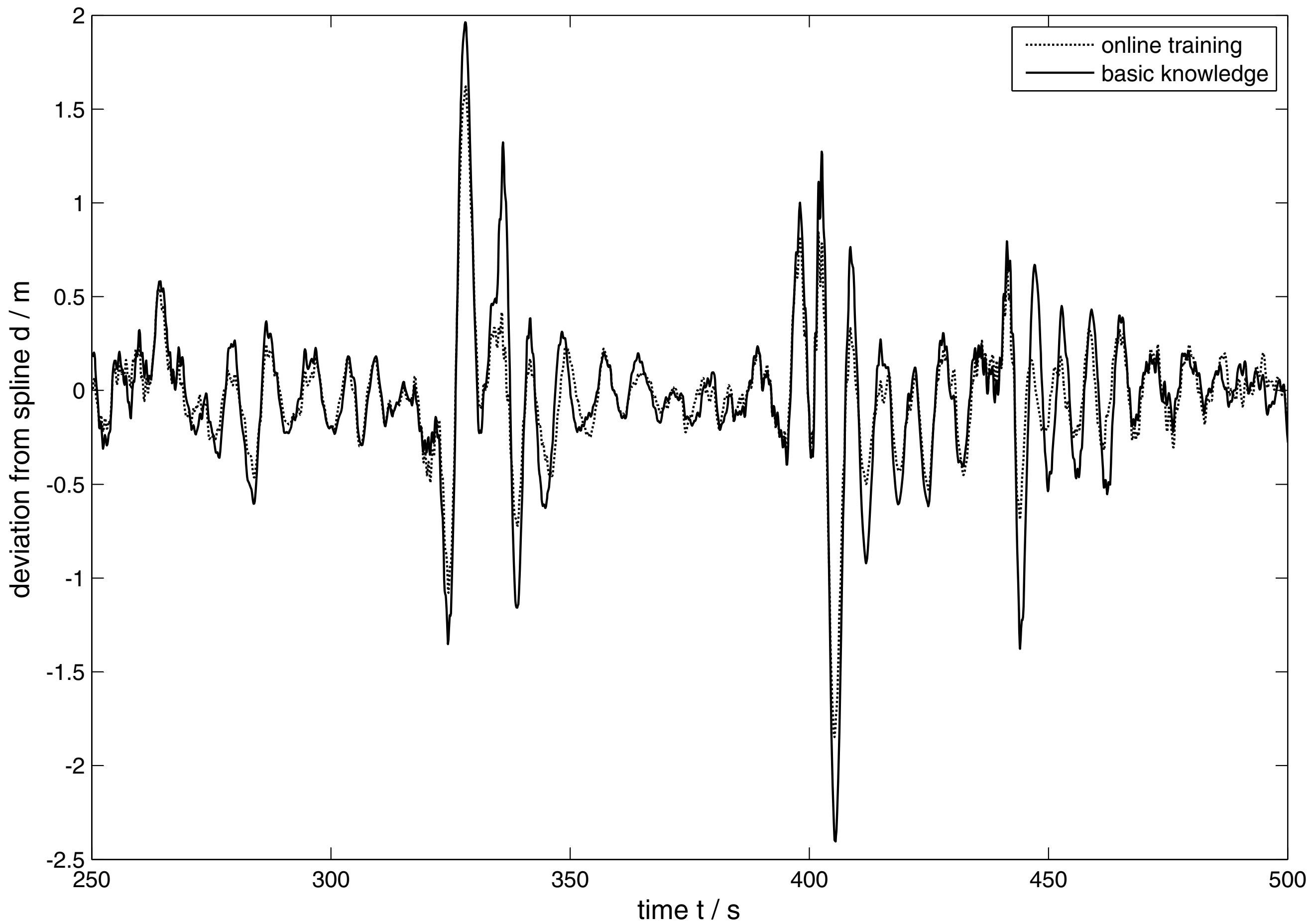

Figure 8: Deviation from the desired spline; this trajectory was not trained, so that the generalisation capabilities and the further online improvements of the ANN-controller become apparent.

and to gradually expand the knowledge. The performance of the neural controller on a completely unknown trajectory as shown in Fig. 8 underlines the improvements during operation. Tests with different initial learning rates showed that knowledge and therefore performance can be lost if the step size on the error surface becomes to large. Here, the known boundary conditions of gradient descend methods take effect. The variation of the learning rate and an additional momentum for weight changes have to be used more carfully as in offline batch-training because only a very limited number of training patterns is used for every online optimisation step. Nonetheless, using a small learning rate, a constant diminishment of the spline-deviation is achieved, especially where the basic ANN-controller showed higher peaks. Obviously the online training is able to redruce the error significantly on unknown trajectories which shows the effectiveness of the control strategy in a more general way. Also, the quality of the predictor network is a central element for online learning since it provides a reliable training signal.

The discussed results are exemplary for the good performance of the different ANN topologies, some perform better and some a bit weaker. Overall, it can be stated that the approach of using modular networks for the control purpose and for analysing the topologies' performance with statistical methods to prove their robust properties is reasonable for the layout of ANN control systems. Additonally, this concept allows an evaluation of the basic knowlegde of the system as well as to implement the abiliy of online-learning, so that the basic konwledge can be adapted during operation. This significantly improves the control process and offers an adaptive control strategy, also taking into account that the optimisation method has to be fast enough for real-time operation. Further research will deal with pure neural control but also with combinations of neural-adaptive and analytic control.

5 Conclusion and Outlook

The application of artificial intelligence in flight control to give the system a certain learning aptitude is an important approach to improve the abilities of UAVs. Modularly implemented neural networks are able to learn the necessary charcteristics to act as flight control elements for an UAV. Not a highly optimised network topology for a specialised application but a stable network generation for the physical boundary conditions of flight control is important. This especially concerns the selection of physically adequate inputs and outputs, which map the non-linear charcteristics the networks shall obtain. The quality of the training data is the essential aspect for offline training and, with it, for the basic knowledge of the control strategy. Not least, suitable optimisation methods have to be chosen for offline and online training, whereas the real-time operation requires a rather fast algorithm. Using groups of networks and the statistical analysis of their training success provides a systematic approach to examine generalisation abilities and robustness. Especially because of the *black box* behaviour of neural networks a methodic training approach is a way to document the ANN-characteristics concernig flight control.

The basic knowledge trained offline provides an adequate foundation for testing the neural control units and allowed the implementaion of a predictor-based online learning architecture. The two main aspects of this analysis are on the one hand the modularity of the ANN-approach granting special knowledge to every neural process unit, combined with a statistical analysis of the performance of many networks supporting the whole control strategy to be reasonable. One the other hand, the ability of the neural predictor to learn the system's dynamics and thus to backpropagate an error signal for the controller optimisation is a key element for this adaptive control approach. The results show that, if provided with suitable inputs from onboard sensors, neural networks are capable of real-time non-linear modelling and control. The neural predictor very precisely learned the dynamics of the UAV for a wide flight envelope and provided a stable error signal for a progressive controller optimisation. Further research will concentrate on validation of the online-learning architecture in flight tests and on further development of the ANN-architecure, especially regarding faster learning under highly non-linear flight conditions like gusts. To adress the problem of output stability of neural networks, the online learning phase will be expanded with a Lyapunov stability analysis of the optimisation method.

References

[1] H-W. Schulz, M. Buschmann, L. Krüger, S. Winkler, and P. Vörsmann. The Autonomous Micro and Mini UAVs of the Carolo-Family. In *Proceedings of the AIAA Infotech@Aerospace 2005*, Arlington, USA, September 2005. Paper 2005-7092.

[2] M. Buschmann, J. Bange, and P. Vörsmann. MMAV - A Miniature Unmanned Aerial Vehicle (MINI-UAV) for Meteorological Purposes. In *16th Symposium on Boundary Layers and Turbulence*, Portland/Maine, USA, 9 - 13 August 2004.

[3] T. Spiess, J. Bange, M. Buschmann, and P. Vörsmann. Unmanned meteorological measurements with the 'M2AV' aircraft - Comparison with remote sensing and the Helipod. In *European Geosciences Union - General Assembly 2006*, Vienna, Austria, 2006.

[4] C.-S. Wilkens, M. Buschmann, L. Johannes, L. Krüger, T. Krüger, and P. Vörsmann. Project Andromeda: Application of Drone Based Aerial Photographs. In 2^{nd} *International UAV World Conference*, Frankfurt a.M., Germany, 12-13 November 2008.

[5] N. El-Sheimy, K.-W. Chiang, and X. Niu. Performance Analysis of GPS/MEMS-Based IMU Data Fusion Utilising Artificial Neural Networks. *European Journal of Navigation*, 2(4):2–11, 2004.

[6] R. De Nardi, J. Togelius, O. Holland, and S. Lucas. Evolution of Neural Networks for Helicopter Control: Why Modularity Matters. In *IEEE Congress of Evolutionary Computation*, Vancouver, 2006. IEEE. 6690-6697.

[7] F.J. Gomez and R. Miikkulainen. Active Guidance for a Finless Rocket using Neuroevolution. In *Proceedings of the Genetic Evolutionary Computaion Conference*, Chicago, 2003. p. 2084-2095.

[8] H. Wu, D. Sun, H. Zhu, and Z. Zhou. An Autonomous Flight Control Strategy Study for a Small-Sized Unmanned Aerial Vehicle. In *IEICE Trans. Electron., Vol. E88-C, No.10*, Chicago, October 2005. pp. 2028-2035.

[9] K. Bickraj, T. Pamphile, A. Yenilmez, M. Li, and I.N. Tansel. Fuzzy Logic Based Integrated Controller for Unmanned Aerial Vehicles. *Florida Conference on Recent Advances in Robotics, FCRAR, Miami*, 2006.

[10] A.A. Pashilkar, N. Sundararajan, and P. Saratchandran. A fault-tolerant neural aided controller for aircraft autolanding. *Aerospace Science and Technology*, 10:49–61, 2006.

[11] S. Russell and P. Norvig. *Künstliche Intelligenz - Ein moderner Ansatz*. Pearson Education, ISBN 3-8273-7089-2, München, 2004.

[12] S.Winkler, M. Buschmann, and P. Vörsmann. An Integrated Navigation System for Unmanned Aerial Vehicles. *European Micro Air Vehicle Conference - Braunschweig*, 2006.

[13] T. Martin, S. Winkler, and P. Vörsmann. GPS/INS Integration for Autonomous Mini and Micro Aerial Vehicle Navigation Considering Time Correlated Sensor Noise. In *ION* 63rd *Annual Meeting*, Cambridge, Massachusetts, USA, 23 - 25 April 2007. p. 710-715.

[14] T. Krüger, L. Krüger, A. Kuhn, J. Axmann, and P. Vörsmann. Flight Control for Micro Aerial Vehicles Using a Modular Neural Network Approach. In *1st CEAS European Air and Space Conference*, Berlin, Germany, September 2007. CEAS-2007-301, pp. 3211-3217.

[15] H.-W. Schulz. *Ein rekonfigurierbares Bildverarbeitungssystem für unbemannte Kleinstflugzeuge - Entwicklung eines bildgestützten Landeverfahrens*. PhD thesis, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, 2008.

[16] U. Krogmann. *Beitrag zur Anwendung Neuronaler Netze in der Flugregelung*. PhD thesis, Faculty of Mechanical Engineering, Technische Universität Braunschweig, 1995.

[17] S. Omatu, M. Khalid, and R. Yusof. *Neuro-Control and its applications - Advances in industrial control*. Springer-Verlag, Berlin, 1996.

[18] R. Rojas. *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, 1996.