International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12474-12482

Direct Neural Network Based UAV Control Algorithm for Moving Landing Site Approach Trajectory Following Using Computer-Vision Based Tracking

[1]Pavel Vyacheslavovich Skribtsov, [2]Aleksey Vladimirovich Dolgopolov
[3]Denis Vladimirovich Tyulyaev, [5]Vladimir Vasilevich Smirnov

[1]Ph.D, CEO, PAWLIN Technologies Ltd, Lenina St., Dubna, Moscow Reg., Russia.
[2]Senior developer, PAWLIN Technologies Ltd, Dubna, Moscow Reg., Russia.
[3]Senior developer, PAWLIN Technologies Ltd, Dubna, Moscow Reg., Russia.
[4]Federal State Higher Military Educational Establishment "Nakhimov Black Sea Higher Naval School", Sevastopol, Russia.
[1,2]Orcid: 0000-0002-1139-168X, 0000-0002-5713-0880

Abstract

Landing site tracking using on-board UAV computer vision system provides angular coordinates of the landing site. However, the traditional on-board approach course control system assumes multiple steps of consequent computations necessary to issue the final directives for aircraft actuators that include current position estimation, intercept trajectory generation and finally low-level UAV control to follow the computed trajectory. Direct neural network-based algorithm is proposed to convert 2D angular measurements of the target and INS readings into control signals for fixed-wing UAV actuators that would lead the aircraft to landing site using a specified approach pattern without any intermediate computation steps. The algorithm is robust, fast and simple and easy to implement in hardware.

Keywords: direct neutral network, UAV control algorithm, computer vision, landing site

INTRODUCTION

The importance of the automatic systems for UAV landing on ships has been confirmed by the multiple researchers in this field and deserve fundamental, thesis type of works [1, 2]. This area is related to more generic area – automatic UAV landing systems [3] and automatic UAV and aircraft control [4]. Components of these systems typically include computer vision, machine learning, automatic control elements (e.g. PID regulators, etc.). Computer vision is usually involved because GPS positional info may either not always be available or fails to provide enough information on the aircraft orientation relative to the static or moving landing site [5]. Neural networks are typically involved to handle uncertainties that are too difficult to express mathematically or strict mathematical solution is too complex to be computed in real-time operation by on-board CPU. Different research projects from this area typically investigate one particular aspect of intellectual UAV control concentrating on a certain element of the automatic landing procedure. Some representative examples of these studies include the early work of Saripalli et al. [6] published in 2003 aimed at automatic helicopter landing using computer vision to estimate height and horizonal deviation of the linearly moving landing site so it covers only final top-down descend part of the landing. In that work no artificial intelligence methods were involved and the control was indirect – output of the vision system was the high level command of the standard on-board low-level control contour. Some recent works, e.g. Hu et al [7], propose direct control of the flight actuators based on computer vision input for the quadrotor landing on the oscillating platform. Another popular sub-area of research in the UAV flight control is neurocontrol where neural network serves as a system identification model trained online or offline. The work of Heryanto et al. [8] is a good example of studies in this area; however it does not cover high-level operation, so trajectory following is done manually by a human pilot. In [1, 2, 9, 10, 11, 12, 13] as in many other similar publications the specific aspects of neural network application in a low-level control contour has been studied within a more generic trajectory – generation-and-following based traditional high-level control loop.

Simultaneous high- and low-level control of the UAV landing or maneuvering by means of neural network is a relatively new line of research, represented for example in [14] where human pilot trains neural network to solve 2D navigation task to fly aircraft to a waypoint via heading control. The work differs greatly from the neural network studies mentioned above, because neural network learns from human and provides trajectory guidance together with low level aircraft control up to the final aileron and elevator surface commands and thus combines outer and inner loops of control just as it would be done by the pilot using the steering stick. The control task is split among several neural networks. The work assumes however that accurate 6DOF estimation is already performed with high degree of accuracy on board and does not use computer vision, which might require quite expensive navigation hardware typically available only in civil aircraft or expensive, heavy UAVs. A very interesting approach is proposed in [15] where *single* neural network learns from the optimal control rules generated by reinforcement learning techniques or Mixed Integer Linear Programming (MILP) that would require too much computation to handle them directly on-board in real time. In this case neural network implements 2D navigation task and provides control of the bank angle.

This means, however, that movements of the ailerons still have to be handled in the additional control loop unit. As in [14] it is assumed that full 6DOF estimation of aircraft state is available, so necessary bearing angles and distance to waypoint can be calculated as the inputs to the network.

This work proposes a method to train neural network to achieve full high- and low- level control of the UAV with inputs from computer vision and IMU only. Instead of the task of navigating to a single waypoint or directly to a landing site the authors research the possibility for the neural network to fulfill the desired approach schema with multiple waypoints encoded implicitly in a single neural network. Thus, the neural network guides aircraft through downwind, base and final legs of approach. It means that once the visual contact with the ship is established the neural network implements landing procedure just by taking computer vision and IMU data as inputs and providing direct control signals for the flight actuators, therefore, nothing else is required. In addition, such method is also operational in GPS-denied environments. Advanced requirements for neural network necessitated more advanced approach to training methods and involved *active learning* [16]. In this case the procedure of neural network training resembles newbie pilots training with an instructor. The summary difference of the proposed method from similar works is expressed in Tabl

Table 1: Comparison of the representative relevant publications with this work

	Saripalli et al [6]	Hu et al [7]	Heryanto et al [8]	Baomar et al [14]	Julian et al [15]	This work
Year of publication	2003	2015	2015	2016	2016	2017
UAV type	Helicopter	Quadrotor	Quadrotor	Fixed wing	Fixed wing	Fixed wing
Neural Net model	-	-	+	+	+	+
Training schema	N/A	N/A	Neural system identification model	1-pass function approx.	1-pass function approx.	Active learning based function approx.
Model Training Data	N/A	N/A	Online IMU	Human Pilot / XPlane Simulator	Reinfor-cement learning or MILP	PID regulator based path following
Dynamic control	-	+	+	+	+	+
Explicit trajectory generation for control	-	+	N/A	-	-	-
Trajectory complexity	1D- Top-down descend	1D- Top-down descend	N/A - trajectory control is done manually by the pilot	1D - Heading control only	2D - single waypoint approach	3D -landing site approach trajectory following, including series of turns and descend
Input signals	Vision (Hue Moments), IMU	Vision, IMU (measuring height above the platform)	Pilot commands (3 axis)	Speed, Altitude, Pitch, Roll Heading	Distance, Bearing angle from heading,Angle to waypoint heading from heading, Bank angle, Vehicle speed	Vision, IMU (altimeter, airspeed, accelerometer, magnetometer, gyro)
Control signal	Indirect: altitude setting which then has to be converted into control signals by on-board control system (Collective Cyclic x2 Rotor)	Direct thrust control	Propellers speed control	Direct: elevator, ailerons, rudder, gear, throttle	Bank angle	Direct : Ailerons & Elevator control
Disturbance	N/A	N/A	N/A	Calm / Stormy weather	wind	Turbule-nce (additive random center mass force & moments)
Target motion	Linear	Sinusoidal	N/A	No	No	Linear
An explicit estimation of the 6DOF state is required	-	-	+	+	+	-

METHOD

The context of application

The proposed method is assumed to be deployed in the following scenario. The UAV uses GPS to reach the area of the last known coordinates of the ship. Then onboard computer vision is deployed to detect the designated ship as proposed in [17]. Once the reliable visual contact with the ship is establishedб the control unit must carry out the approach pattern including necessary steps of passing downwind, base and final legs. This step is the main subject of the present research. After brining the UAV to the close vicinity of the ship, typically landing site landmarks become visible, therefore the final step of the flight (flare and touchdown or getting into net) can be implemented using the techniques similar to [6, 7, 18, 19], which is left out of scope of this work, although similar approach to the one proposed in this paper can be derived for this end step as well.

General schema of the study

Overall approach used in this work is depicted in Figure 1.

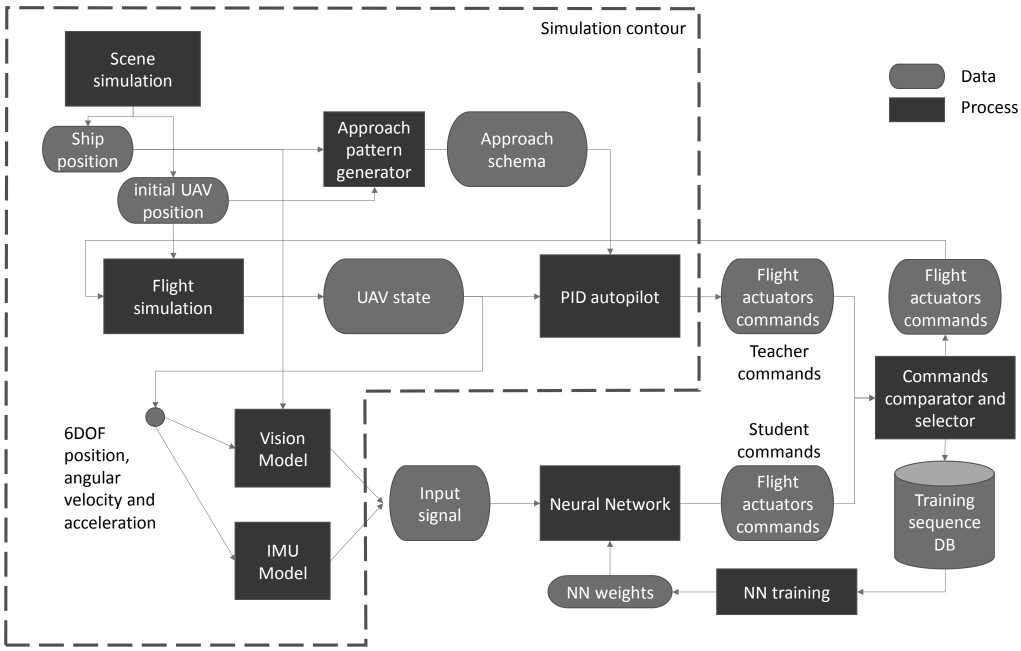

Figure 1: Overall schema of the neural network controller synthesis

The diagram consists of a simulation contour (dashed area) and a neural network training contour (rest of the diagram), which in turn consist of the following components:

- The scene simulation is used to produce the initial ship and UAV positions;
- The approach pattern generator outputs landing trajectory as a set of waypoints using predefined approach pattern that depends on the relative location of the UAV with respect to the ship and typically consists of downwind, base and final legs;
- The Flight Simulation generates the UAV state based on the initial state, the UAV control commands and atmospheric disturbance (turbulence);
- The autopilot generates the UAV control commands to follow the generated landing trajectory taking into account the current UAV 6DOF position & orientation;
- The vision model emulates output of the on-board computer vision system that produces relative angular information of the ship landmarks with respect to UAV coordinate system;
- The IMU model emulates the UAV MEMS-sensors information;
- The Neural Network transforms IMU and Vision data into the final UAV flight actuators – ailerons and elevator desired setting;
- The commands comparator and selector choose which control signal to pass and populate the training database. This is done differently in 3 operation modes which will be described in more details below;
- The NN Training process finds optimal neural network weights according to the training database collected.

Note that nothing prevents the approach from changing the simulation contour to the actual UAV, autopilot, vision and IMU modules, provided appropriate data collection pipes would be introduced in the UAV controller software that would capture autopilot commands, vision and IMU measurements. Real flight experiments are planned in future works, whereas at this stage the concept has been confirmed using simulation of the corresponding units.

Scene simulation and approach pattern generator

Scene simulation algorithm generates a set of random ship positions, whereas the approach pattern generator generates an approach path (including the final approach) depending on the relative position of the ship and UAV. Figure 2 illustrates the approach pattern used in the experiments, although its particular schema may vary. Figure 3 shows a set of generated ships and a subset of landing approach trajectories that the UAVs fly using a traditional autopilot module

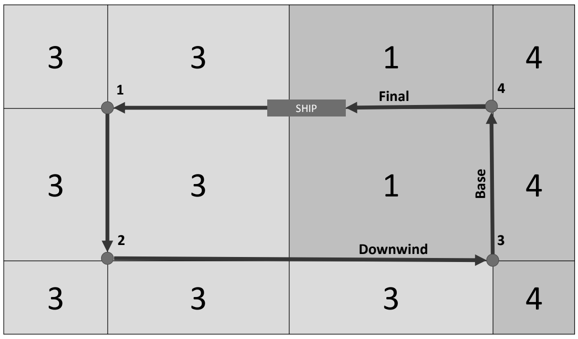

Figure 2: The approach schema used. One of the four turns that connect legs of the approach pattern are used as the first waypoint depending on the initial location of the aircraft relative to the ship. Then approach continues using the standard order

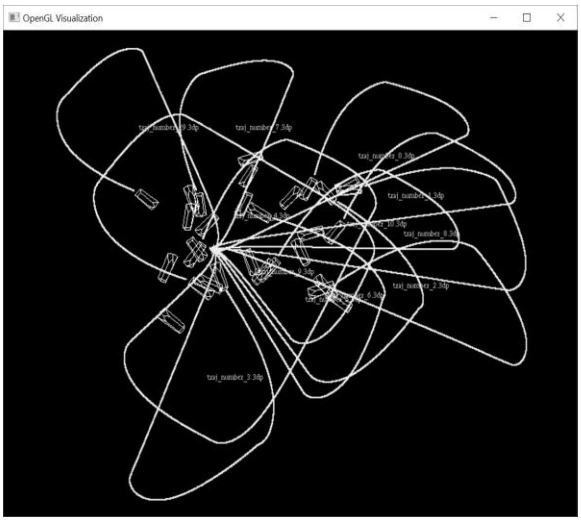

Figure 3: Visualization of the random subset of ships and UAV approach trajectories flown by the UAV using a regular autopilot into which the approach pattern has been loaded.

Flight Simulation

Lightweight custom simulation software has been created using C++ language to facilitate the optimized (much faster than real-time) flight modelling according to the generic flight dynamics used in most flight computer simulators [20, 21]. The model realism was tested using actual UAV pilots. Visualization was done using OpenGL(see Figure 4). The simulation models air turbulence by appending disturbances to total force and total moment

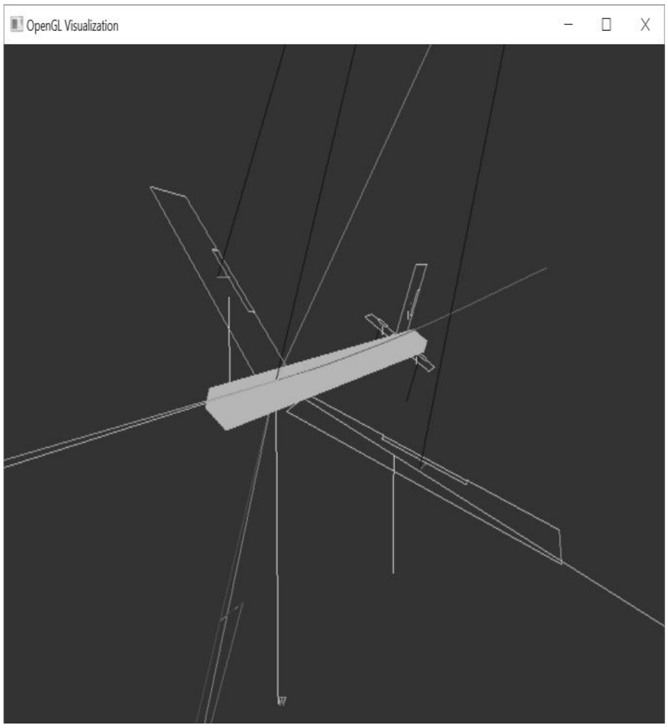

Figure 4: Screenshot from the flight simulation software. Computed velocities, forces and moments, position of control surfaces are visualized.

Autopilot

The autopilot used in experiments was a simple hierarchical PID-regulator based autopilot. Note that, unlike the neural network controller, the autopilot module leads the UAV taking into account its full state information including 6DOF position & orientation, whereas neural network controller has only implicit information about the aircraft position (see below).

Computer vision setup

It is assumed that angular coordinates of the front and back edges of the ship relative to the UAV coordinate system can be received from the computer vision working in the continuous target tracking mode, as shown in [18, 19, 22]. Angular coordinates of the target (see Figure 5) are used to drive the pan-tilt-zoom camera unit, and taking into account camera optical axis position the angular coordinates to the ship edges can be calculated in real time. Supposing the field of view in a zoom mode of the UAV on-board camera unit is 5 degrees, then with resolution of 1200x800 the accuracy of the angular coordinates measurement can be better than 0.004 degrees or 0.00007 rad.

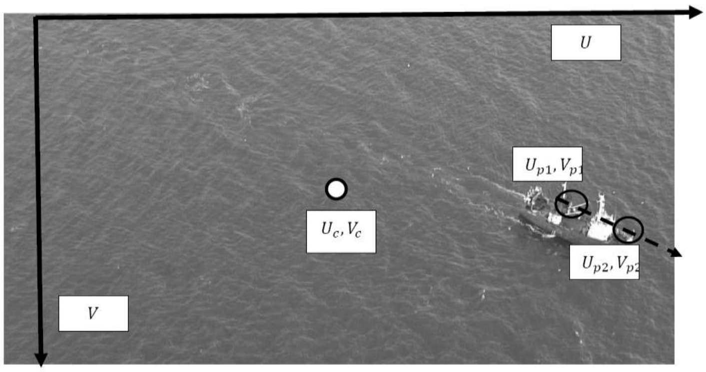

Figure 5: Camera angular coordinate system (U, V), point "c" – optical center, "p_1", "p_2" – front and rear edges of the ship.

Determining 6DOF position and orientation of the UAV using vision and IMU measurements is possible in theory and practice, but it is not a simple task and resembles a separate line of research that can be represented by such work as [23]. In this work the authors eliminated the necessity of direct computation of 6DOF state assuming that necessary computations would be done implicitly inside the neural network "black box". The research shows that the proposed method is another method of IMU and vision data fusion, though implicit as it does not compute 6DOF state, but allows necessary intrinsic values to be computed within the neural network enabling necessary subsequent calculations just as human pilot "knows where to steer the plane" on visual approach without explicit manipulations with the coordinates or angles.

IMU model

The following sensors have been emulated:

- 3-axis accelerometer
- 3-axis magnetometer
- 3-axis gyro
- Air speed
- Altitude

Note that combination of accelerometer and magnetometer typically does not allow computing orientation of the aircraft directly because of the centripetal accelerations in rolls of the fixed wing. Most probable accelerometer reading is directed towards the bottom of the UAV independently of its bank angle, therefore determining orientation of the UAV from these measurements is a non-trivial task and in general cannot be solved without involving computer vision.

Neural Network controller

Neural network controller used is designed as follows:

Input signal

The neural controller works with this 18-dimensional input space, characterized in Table 2.

Table 2: Input signal description

#	Variable	Dimension	Description	Data source
1	p_1	3	Unit vector aiming at the ship front end in the local airplane system of coordinates	Computer Vision Unit
2	p_2	3	Unit vector aiming at the ship rear end in the local airplane system of coordinates	Computer Vision Unit
3	s_1	3	same as p_1, but at the beginning of the approach, as the pattern may depend on initial relative position of the UAV and ship	memory
4	s_2	3	same as p_2 at the beginning of the approach (see above comment)	memory
5	acc	3	Accelerometer reading	IMU
6	gyr	3	Gyro reading	IMU
7	mag	3	Magnetometer reading	IMU
8	h	1	Altitude	Altimeter reading, assumed to be corrected for local pressure
9	v_x	1	Indicated Air speed	Pito air speed sensor reading
	TOTAL input space dimensions	22		

Target (desired) output signal

In our case the teacher of the neural network is the hierarchical PID regulator based controller and potential advantage of using it as the teacher for a neural network control model is shown in [13]. The approach trajectories used as input for the PID-regulator controller for the random combination of UAV and ship positions were generated using an approach similar to [24].

Neural network (NN) architecture

The MLP neural network model [25] is used to approximate the control decision rules learnt from a teacher using RPROP algorithm [26]. The network used symmetrical sigmoid activation function with an output in the range of [-0.5;+0.5].

NN Training process

Training has been done using proprietary software [27] allowing GPU acceleration using NVIDIA CUDA backend. NVIDIA GeForce 1000 series GPU has been deployed providing more than x10 times acceleration of neural computations compared to the CPU version powered by Intel MKL Library. For better approach phase identification, the network outputs contained 4 phase id recognition outputs, in addition to the 2 main "elevator" and "ailerons" command outputs. Since NN training results depend on random initial weights, the NN benchmark procedure (see below) was used to select the best network of a few trials.

Commands comparator and selector

The input dimension state for the neural network is so big, that it is not possible to generate training dataset that would populate all this space with sufficient resolution (curse of dimensionality). Therefore an original approach, inspired by the active learning paradigm, has been deployed. Imagine autopilot being an instructor for a newbie pilot (NN). The instructor allows the newbie pilot to rule the stick until the commands of the student are too contradictive to what the instructor would do. In such moments, the additional training records were added to the training database. After the training flight was over, the NN training with the updated database was involved. Since simulation allowed safety of experiments in all conditions, the learnt student (NN) was allowed to make a series of approaches without the instructor and the success rate was measured, depending on the results of the operation, which could be one of the following:

- Success, meaning that the aircraft reached the final TGT within a designated threshold;
- "Too long" flight, meaning that the aircraft has been put to an infinite loop or fly-away path that did not end it with a successful landing within a certain time period threshold;
- Crash (hitting the ground/sea).

After the benchmark flights by the NN alone were over, the NN was put to fly with the "instructor" again, populating more records in the training DB, and so on, hopefully improving skills after each next training flight.

To summarize, the Commands comparator and selector works in 3 modes:

- "instructor only" – fly by autopilot (mainly used to check the autopilot),
- "NN-only" – exam flight, no autopilot used
- "Hybrid" – training flight using the principles explained above.

RESULTS

The typical flow of the active learning path is shown in Table 3 for the case with 64 hidden units used.

Table 3: Example of the active learning "register"

iteration	success rate	"too long" rate	crash rate	tgtAvgDist
0	0.03	0.00	0.97	581.7
1	0.13	0.10	0.77	215.0
2	0.10	0.33	0.57	314.1
3	0.33	0.37	0.30	150.1
4	0.37	0.37	0.27	135.9
5	0.27	0.23	0.50	252.2
6	0.07	0.80	0.13	341.6
7	0.27	0.50	0.23	274.3
8	0.60	0.37	0.03	163.3
9	0.30	0.70	0.00	281.0
10	0.20	0.67	0.13	222.1
11	0.67	0.17	0.17	91.9

As one can see, the results are fluctuating but the NN improves result almost each next flight.

The success rate and crash rates graphs are represented in Figure 6.

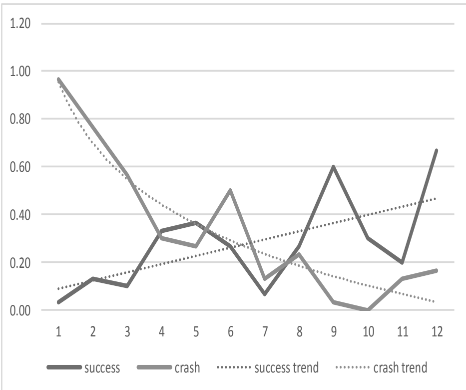

Figure 6: Increasing success and decreasing crash rate over "training" flights. Trendlines are shown to smooth the fluctuating results.

Figure 7 shows the rate of "trust" of an utopilot/"instructor" to the NN/"student

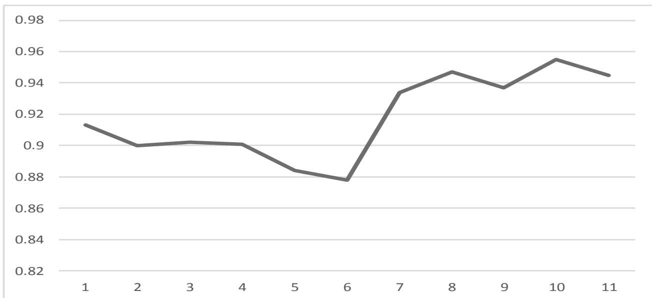

Figure 7: Increasing rate of "trust" – the percentage of cases where command selection model allowed the trained neural network to control the UAV depending on the number of lessons/training flights.

Figure 8 shows the unsuccessful ("crash" or "too long") flight trajectories implemented by the neural network after different number of training flights.

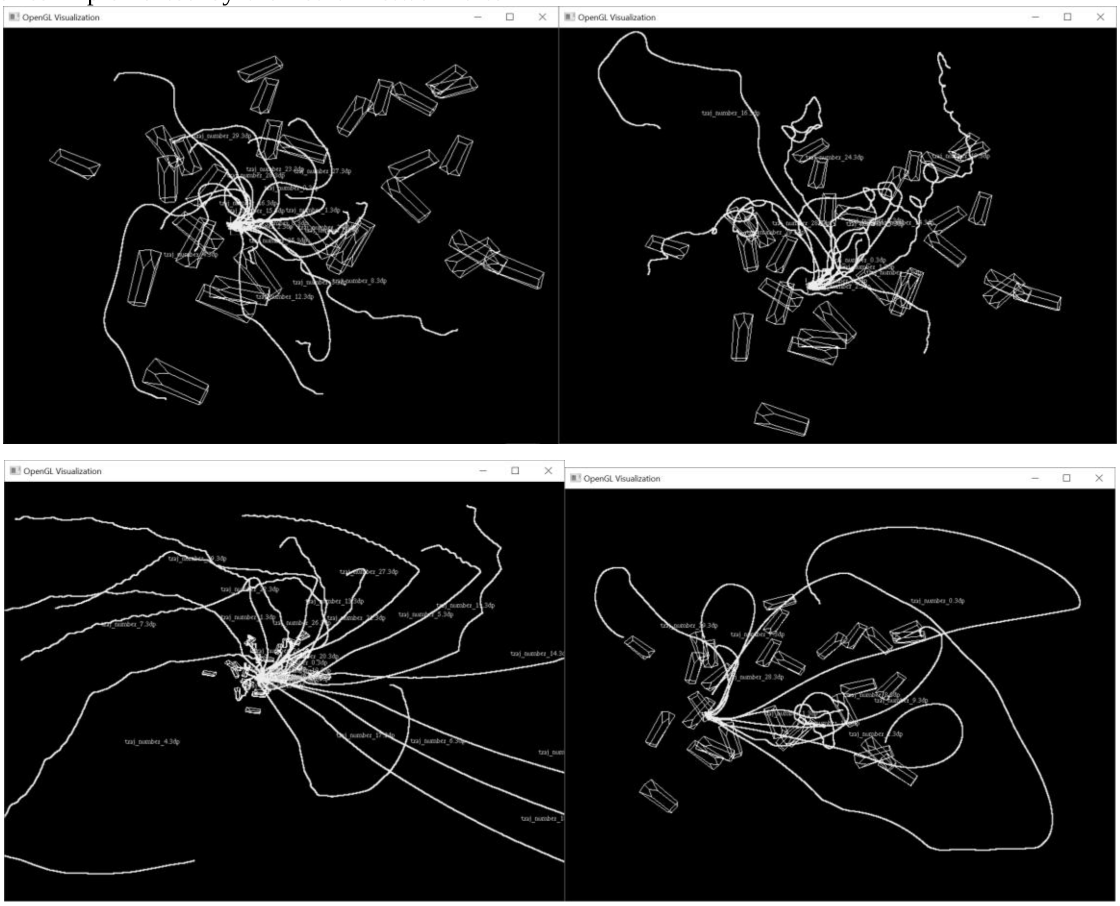

Figure 8: Examples of non-successful flight trajectories implemented by the undertrained neural network. Overcompensation leads to oscilations and crashes and under-control leads to "fly-away" curves.

Increasing the number of neurons enables to increase the feasible success rate. The number of the required hidden neurons is dependent on the complexity of the approach path and variaty of the possible initial locations of the UAV relative to the ship. For the approach pattern descirbed above, 100% success rate was reached after the 3rd round of experiments by gradually increasing the number of hidden neurons in the network (see Table 4)

Table 4: Maximum achievable success rate depending on the number of hidden neurons in the neural network controller

Hidden neurons	Success rate
32	0.50
64	0.67
256	1.0

Example successful trajectories of the UAV controlled only by the neural network in an "exam" flight are shown in Figure 9

Figure 9: An example subset of successful trajectories for random ship dispositions during the neural network ship landing approach

DISCUSSION

The computer simulations demonstrated the proof-of-concept for the possibility to train the neural network to control the ship landing approach procedure of the UAV using computer vision and IMU readings. Since the decisions implemented by the neural network depend on the relative location of the ship and aircraft and the ship movement is slow relative to the UAV, the concept may be applied to approaching the moving ships as well.

Further research must study the effects of increasing variation of the initial conditions, wind, and sensor errors. In terms of the neural network model development – recurrent architectures could be considered that would allow for some way of the internal state representation. The neural network complexity must be kept in mind to allow for on-board real time computation of the neural network output signal. Finally transition from simulation to real flights study is required.

The input of this work can be best described by comparing our findings with those of [15] provided in Table 5.

Table 5: Comparison of this research with the recent state-of-art work (most important differences)

Parameter	Julian et al	This work
Trajectory complexity	single waypoint	multi-waypoint landing approach schema
Horizontal control	Yes	Yes
Vertical control	No	Yes
Need in 6DOF UAV state estimate	Yes	No
Ready to work in GPS denied environment using Vision+IMU data fusion	No	Yes
Further low-level controller needed	Yes (to control the set value of the bank angle using Ailerons)	No (neural network outputs Ailerons & Elevator commands directly)
Simulation verification	+	+
Real flight verification	Yes	No

CONCLUSION

The concept of the neural network based controller for the ship landing approach procedure of the UAV using computer vision and IMU readings was offered along with the methodology of overcoming the curse-of-dimensionality problem by means of Active Learning technique that resembles a lot the process of human pilots' training by the instructor. The final control algorithm deploying 256-hidden units MLP is robust, fast and simple and easy to implement in low-size, low-cost hardware and allows for autonomous operation in GPS denied environment. Computer simulations confirm the proof-of-the concept and open new directions for additional studies required to deploy the method in the real flying UAV. Current research should be treated as the next step towards a fully neural-network UAV flight and landing controller.

ACKNOWLEDGEMENT

The authors thank the Ministry of Education and Science of the Russian Federation for supporting this research. The research was financed by the Ministry of Education and Science of the Russian Federation under the Grant Contract dated 27 October, 2015, (grant number 14.607.21.0127, Contract Unique Identifier RFMEFI60715X0127), the grant was assigned for the research into "Development and implementation of autonomous light-weight aircraft UAV ship deck landing using machine vision system".

REFERENCES

[1] Frølich, M., 2015, "Automatic Ship Landing System for Fixed-Wing UAV", MS thesis, Norwegian University of Science and Technology, Trondheim, Norway.

[2] Nevstad, S. O., 2016, "Autonomous Landing of Fixed-Wing UAV in Net suspended by Multirotor UAVs-A Fixed-Wing Landing System", MS thesis, Norwegian University of Science and Technology, Trondheim, Norway.

[3] Gautam, A., Sujit, P. B., Saripalli, S., 2014, "A survey of autonomous landing techniques for UAVs", Proc. International Conference on Unmanned Aircraft Systems, Institute of Electrical and Electronics Engineers, Bonnet Creek - Orlando, FL, pp. 1210-1218. DOI: 10.1109/ICUAS.2014.6842377

[4] Adami, T. M., Zhu, J. J., 2011, "6DOF flight control of fixed-wing aircraft by trajectory linearization", Proc. American Control Conference, Institute of Electrical and Electronics Engineers, the Hilton San Francisco, California, pp. 1610-1617. DOI: 10.1109/ACC.2011.5991521

[5] Fitzgerald, D., Walker, R., Campbell, D., 2005, "A vision based forced landing site selection system for an autonomous UAV", Proc. Intelligent Sensors, Sensor Networks and Information Processing Conference, Institute of Electrical and Electronics Engineers, Melbourne, Australia, pp. 397-402. DOI: 10.1109/ISSNIP.2005.1595612

[6] Saripalli, S., Sukhatme, G. S., 2003, "Landing on a moving target using an autonomous helicopter", Field and Service Robotics, pp. 277-286. DOI: 10.1007/10991459_27

[7] Hu, B., Lu, L., Mishra, S., 2015, "Fast, safe and precise landing of a quadrotor on an oscillating platform", Proc. American Control Conference, Institute of Electrical and Electronics Engineers, Chicago, IL, pp. 3836-3841. DOI: 10.1109/ACC.2015.7171928

[8] Heryanto, M. A., Wahab, W., Kusumoputro, B., 2015, “Optimization of a neural network based direct inverse control for controlling a quadrotor unmanned aerial vehicle”, MATEC Web of Conferences, 34, pp. 1-4. DOI: 10.1051/matecconf/20153404003

[9] Kaneshige, J., Bull, J., Totah, J. J., 2000, “Generic neural flight control and autopilot system”, Proc. AIAA Guidance, Navigation, and Control Conference, Denver, CO, USA, 4281, pp. 1-11. DOI: 10.2514/6.2000-4281

[10] Lee, T., Kim, Y., 2001, “Nonlinear adaptive flight control using backstepping and neural networks controller”, Journal of Guidance, Control, and Dynamics, 24(4), pp. 675-682. DOI: /10.2514/2.4794

[11] Ambati, P. R., Padhi, R., 2017, “Robust auto-landing of fixed-wing UAVs using neuro-adaptive design”, Control Engineering Practice, 60, pp. 218-232. DOI: 10.1016/j.conengprac.2016.03.017

[12] Kayacan, E., Khanesar, M. A., Rubio-Hervas, J., Reyhanoglu, M., 2017, “Learning Control of Fixed-Wing Unmanned Aerial Vehicles Using Fuzzy Neural Networks”, International Journal of Aerospace Engineering, 2017, pp. 1-12. DOI: 10.1155/2017/5402809

[13] Artale, V., Colotta, M., Milazzo, C., Pau, G., Ricciardello, A., 2016, “Real-time system based on a neural network and PID flight control”, Applied Mathematics & Information Sciences, 10(2), pp. 395-402. DOI: 10.18576/amis/100202

[14] Baomar, H., Bentley, P. J., 2016, “An Intelligent Autopilot System that learns piloting skills from human pilots by imitation”, Proc. International Conference on Unmanned Aircraft Systems, Institute of Electrical and Electronics Engineers, Arlington, VA, pp. 1023-1031. DOI: 10.1109/ICUAS.2016.7502578

[15] Julian, K. D., Kochenderfer, M. J., 2017, “Neural Network Guidance for UAVs”, Proc. AIAA Guidance, Navigation, and Control Conference, American Institute of Aeronautics and Astronautics, Grapevine, Texas, pp. 1-14. DOI: 10.2514/6.2017-1743

[16] Hasenjäger, M., and Ritter, H., 2002, “Active learning in neural networks”, New Learning Paradigms in Soft Computing, L. C. Jain, and J. Kacprzyk, eds., Physica, Heidelberg, Germany, 84, pp. 137-169. DOI: 10.1007/978-3-7908-1803-1_5

[17] Dolgopolov, A. V., Kazantsev, P. A., Bezuhliy, N. N., 2016, “Ship Detection in Images Obtained from the Unmanned Aerial Vehicle (UAV)”, Indian Journal of Science and Technology, 9(46), pp. 1-7. DOI: 10.17485/ijst/2016/v9i46/107562

[18] Moriarty, P., Sheehy, R., Doody, P., 2017, “Neural networks to aid the autonomous landing of a UAV on a ship”, Proc. 28th Irish Signals and Systems Conference, Institute of Electrical and Electronics Engineers, Killarney, Ireland, pp. 1-4. DOI: 10.1109/ISSC.2017.7983613

[19] Yakimenko, O. A., Kaminer, I. I., Lentz, W. J., and Ghyzel, P. A., 2002, “Unmanned Aircraft Navigation for Shipboard Landing Using Infrared Vision”, IEEE Transactions on Aerospace and Electronic Systems, 38(4), pp. 1181-1200. DOI: 10.1109/TAES.2002.1145742

[20] Sinha, N. K., Ananthkrishnan, N., 2013, Elementary Flight Dynamics with an Introduction to Bifurcation and Continuation Methods, CRC Press, Boca Raton, Florida, USA.

[21] Schmidt, L. V., 1998, Introduction to Aircraft Flight Dynamics, American Institute of Aeronautics and Astronautics, Reston, VA, USA.

[22] Kazantsev, P. A., Sadakov, V. A., Chupakov, M. V., 2016, “Maritime Vessels Real-time Tracking-by-detection in UAV Videos”, Indian Journal of Science and Technology, 9(48), pp. 1-9. DOI: 10.17485/ijst/2016/v9i48/107490

[23] Webb, T. P., Prazenica, R. J., Kurdila, A. J., Lind, R., 2007, “Vision-based state estimation for autonomous micro air vehicles”, Journal of Guidance, Control, and Dynamics, 30(3), pp. 816-826.

[24] Dubins, L. E., 1957, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents”, American Journal of Mathematics, 79(3), pp. 497-516. DOI: 10.2307/2372560

[25] Haykin, S., 1999, Neural Networks: A Comprehensive Foundation, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

[26] Riedmiller, M., and Braun, H., 1993, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm”, Proc. IEEE International Conference on Neural Networks, San Francisco, CA, 1, pp. 586-591. DOI: 10.1109/ICNN.1993.298623

[27] Skribtsov, P. V., Dolgopolov, A. V., 2007, “Comparison of performance graphics accelerators and CPU in the calculations for large volumes of processed data”, Neurocomputers: Development, Application, 9, pp. 421-425.