This paper addresses the issue of developing a computerized system for processing information in the construction of the trajectory of an unmanned aircraft vehicle (UAV), a remotely-piloted aviation system (RPAS), or another robotic system. Resolving this task involves the neural network learning algorithms based on the mathematical model of movement.

The construction of such a trajectory between two specified destinations has been considered that provides for the possibility of bypassing static and dynamic obstacles. The specified trajectory is divided into several smaller parts. The possibility of restructuring when changing the position of obstacles in space has been considered. A UAV flight control algorithm has been developed, which implies training a neural network for bypassing obstacles of different sizes.

To predict the development of the situation when an object moves between two specified points in space, it is proposed to use the Q-Learning algorithm. It has been shown that the smallest number of steps required for moving along a specified trajectory is 18, the largest is 273 steps. In case of distortion during data transmission, the training of the neural network makes it possible to reduce the possibility of collision with obstacles by improving the accuracy and speed of information transfer between the on-board computer and operator. A system of the video support to moving objects was modeled; dependence charts of the normalized frame size at different parameter values were built. Using the charts makes it possible to determine the function of the maneuver intensity. Existing neural network learning methods such as CNN and LSTM were compared. It has been proven that the success rate reaches 74 % when using CNN only, while it amounts to 92 % at the hybrid application of CNN+LSTM. The simulation results have demonstrated the high efficiency of the developed algorithm

Keywords: computerized system, information processing, motion trajectory, neural network

UDC 004
DOI: 10.15587/1729-4061.2021.225501

DESIGNING A COMPUTERIZED INFORMATION PROCESSING SYSTEM TO BUILD A MOVEMENT TRAJECTORY OF AN UNMANNED AIRCRAFT VEHICLE

V. Kvasnikov
Doctor of Technical Sciences, Professor, Head of Department*
E-mail: kvp@nau.edu.ua

D. Ornatskyi
Doctor of Technical Sciences, Professor*
E-mail: ornatskiyd@ukr.net

M. Graf
Senior Lecturer
Department of Computer Science
Zhytomyr Polytechnic State University
Chudnivska str., 103, Zhytomyr, Ukraine, 10005
E-mail: graf.maryna@gmail.com

O. Shelukha
Assistant*
E-mail: alexztshell@gmail.com

*Department of Computerized Electrotechnical Systems
and Technologies
National Aviation University
Liubomyra Huzara ave., 1, Kyiv, Ukraine, 03058

Received date 04.01.2021
Accepted date 05.02.2021
Published date 26.02.2021

Copyright © 2021, V. Kvasnikov, D. Ornatskyi, M. Graf, O. Shelukha
This is an open access article under the CC BY license
(http://creativecommons.org/licenses/by/4.0)

1. Introduction

The application of computer systems and components over recent years has been characterized by an increase in productivity, performance, and energy efficiency. At the same time, their mass-dimension parameters continue to decrease. New technologies and design algorithms are used to modernize and build new computerized systems, in particular for solving tasks of information transfer and processing, flight trajectory planning, video data processing, etc. Most of these tasks relate to the process of observing and bypassing fixed objects [1].

A special role in the operation of robotic systems belongs to the processes that track surrounding moving objects within the field of view of a given system's optical modules. When considering an unmanned aircraft vehicle (UAV) or a remotely-piloted aviation system (RPAS), those processes include the planning of a natural motion trajectory with respect to the observed objects.

To resolve such tasks, the most used tools are the neural network planning algorithms, the construction of algorithms on graphs [2, 3], as well as the application of random tree methods [4, 5]. When supporting a moving object and constructing a trajectory of its movement, the initial data employed are the coordinates of the observation objects.

The existing methods and algorithms do not properly enough highlight the choice of a motion trajectory, or a flight trajectory, considering the bypassing of obstacles because no information about obstacles is stored in the memory before the movement started. This is an absolutely relevant issue within a dynamic environment that requires the redevelopment of a motion trajectory when obstacles change their position without human intervention, that is, the automated operation of a computerized information processing system.

The process of highlighting an object in the image is shown in works [6, 7], which give algorithms for processing images and selecting moving objects on them. Papers [8–10] examine the methods of processing data about a moving object and designing tracking systems. However, these methods are meant for stationary surveillance systems. Studies [12–14] describe a mathematical apparatus of the computerized control over mobile complexes, use which could help develop a mobile system of video surveillance and support.

To build an algorithm to study a computerized system, it is necessary to collect, process data, as well as prepare, develop, and find a model based on the obtained data, conduct an experiment on the training of the resulting model. If we consider UAV, its control system resolves the task of achieving the specified coordinates. In [15], to guide UAV along the trajectory of the motion control, it is proposed to build an intelligent information processing system within the UAV control unit. Work [15] suggests its construction by solving the problem of calculating generalized coordinates and selecting a trajectory to achieve a specified position.

Therefore, there is a need to design a computerized information processing system for the construction of UAV movement trajectory with improved characteristics.

2. Literature review and problem statement

Existing mathematical models of information processing when controlling UAV are considered in study [16], which reports the analysis of factors taken into consideration in the development and application of UAV mathematical models; the equations for modeling UAV movement were analyzed. To solve the tasks of movement description, the construction of mathematical models involves different levels of models at different stages, described in [17]. The models' description is considered depending on the level of complexity and purpose. In general, there are four levels of models: initial models, linearized or nonlinear simplified models, linear models simplified for certain types of movement, and machine models. However, the factors proposed to be taken into consideration in this work do not provide for the possibility of restructuring the trajectory in the process of movement without the intervention of the operator.

Schematic representation of a model to control the UAV movement is given in [18]. The work describes the dynamic modeling of UAV equipped with a fixed wing. The mathematical model is considered for any mode of flight and type of movement, relying on a system of equations that include equations of the dynamics and kinematics of the center of masses, the dynamics and kinematics of angular motion.

Paper [19] shows that additional equations can be included in the initial system to describe the relationship between angles in different coordinate systems. In addition, expressions for the aerodynamic forces and moments, the acceleration of the gravitational field, as well as its dependence on altitude and air density, can be added to such a system.

Detailed analysis of the existing methods of information processing in the UAV control unit is reported in [20]. In the study, in order to process data when controlling UAV the greatest attention is paid to methods such as neural networks, genetic algorithms, and fuzzy logic. In addition to the theoretical methods based on the construction of mathematical models, much attention is paid to methods based on learning. It is clear that training is executed through commands or by reproducing certain actions. Most often, methods based on the use of artificial neural networks (ANN) or methods based on the use of expert systems are used in robotics to build computerized systems for image recognition and motion trajectory construction. When using ANN, the efficiency of operation is improved due to the rapid development of modern methods of parallel calculations [21]. The methods based on the use of ANN require the construction of an array of input data, the selection of network architecture, and a learning algorithm [22]. However, when applied, it is often not implied that in the event of obstacle and/or radio interference, the input data may be unknown and constantly changing.

To solve the tasks of planning a UAV movement trajectory, genetic algorithms are used that have a certain advantage over other technologies. These are advantages such as stability as a result of the possibility to encode parameters, the use of minimum information at random choice, as well as operations over populations [23]. However, such advantages are directly dependent on the solution coding technique while most mutations do not lead to improved results.

To determine the nature of body movement according to the specified forces, there are algorithms and methods involving the inverse problem. If a given task is considered from the point of view of control systems, then it comes down to finding generalized coordinates in the reference coordinate system [24]. Using inverse problems, it is possible to design control algorithms that provide the necessary dynamic properties of the system [25]. However, a given method is complicated for implementation and requires solving a system of differential equations, which leads to a decrease in the accuracy and performance of data transmission.

Recently, the method of specified synergy [26], which arises from the combined influence of factors, has become widespread. The described method is characterized by the fact that the combined action significantly exceeds the effect of each individual component and a simple sum, which is also described by the authors of work [27]. The synergistic method belongs to the class of semi-inverse methods; it boils down to the fact that some coordinates are specified explicitly while all others are determined from equations and relationships. However, when using methods with the predefined synergy, there is a difficulty in predicting the results, which complicates decision-making and leads to low accuracy and speed of data transmission.

For a long time, the main approach to solving the tasks of constructing a route was simultaneous location and cartography. That allows the use of UAV among trees and houses. A stereo camera, lidar, or other sensors are used as the monitoring systems. This approach makes it possible to partially resolve issues related to the construction of a motion trajectory [28]; however, there remain certain limitations when changing the position of objects along the trajectory.

The systematization of the identified subtasks, the construction of new algorithms and methods for building a UAV movement trajectory would make it possible to design a computerized system of information processing with improved characteristics.

3. The aim and objectives of the study

The aim of this study is to design a computerized information processing system to solve the task of planning a movement trajectory of UAV or RPAS taking into consideration the bypassing of obstacles within the static and dynamic space.

To accomplish the aim, the following tasks have been set:

– to devise a machine learning method based on the existing reinforcement learning (RL) algorithm in an unchanging environment in order to plan a UAV flight trajectory;

– to simulate a system for the video support to moving objects and calculate the parameters of obstacle movement;

– to build a structural scheme of the computerized information processing system for constructing a movement trajectory in a dynamic environment and a step-by-step algorithm for calculating the parameters of movement of the observed object;

– to suggest an algorithm for planning the trajectory of UAV movement in a dynamic environment.

4. The study materials and methods

4. 1. Devising a machine learning method

Consider machine reinforcement learning (RL) methods.

Typically, RL configuration consists of two components: the agent and the environment. Their interaction is schematically shown in Fig. 1.

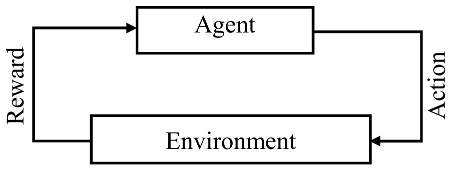


Fig. 1. Interaction of components in reinforcement learning

The environment refers to the object the agent acts on, while the agent represents the RL algorithm. From the environment, there is a dispatch of the state to the agent that takes actions in response to this state, based on the acquired, recorded, or obtained as a result of training, knowledge. After that, the next state data are received from the environment and the agent's reward is sent again. At the same time, the agent updates its knowledge taking into consideration the reward received, for the possibility of evaluating the latest actions. This loop repeats until the environment sends a status value to complete the phase. The goal is to learn how to act in such a way as to get the maximum reward.

The described actions are templated for most RL algorithms, including Q-Learning, SARSA, DQN (Deep Q Network), DDPG (Deep Deterministic Policy Gradient).

Q-Learning Algorithm.

This learning algorithm is based on the equation described in work [29]. In a given equation, the first value is the value of a certain state, the second is the sum of the value of this state and all possible actions from this state. That is, at Q-Learning, the value of Q is determined by the method of trial and error. To determine Q, one must initialize Q, select an action and perform it, conduct an assessment taking into consideration the reward received and update the Q value. The objective function takes the following form [30]:

$$Q_k(s_k,a_k)=Q(s,a)+$$
$$+\vartheta\left(r+\gamma\cdot\max Q_k(s_k,a_k)-Q(s,a)\right), \tag{1}$$

where $Q(s, a)$ is the current value of the function;

$Q_k(s_k, a_k)$ is the value of the function in the next step;

max $Q_k(s_k, a_k)$ is the choice of the maximum value of all possible in the next step;

s – the current position of the agent;

a – current state;

ϑ – the speed of training;

r – reward received in the current position;

γ – decrease in a reward;

s_k – the next selected position in accordance with the next selected action;

a_k – the next selected action.

The purpose of the Q-Learning algorithm is to maximize the Q value. Based on the received reward, the agent forms the utility function Q. Further, at each following step, the strategy of behavior is chosen taking into account the previous experience of interaction. That is, one needs to maximize the total reward using the utility function, rather than the reward in the current step.

4. 2. Simulation of a system for the video support to moving objects

Consider methods for calculating and extrapolation of obstacle movement parameters.

Bypassing the task of detecting an object in the frame of the tracking system, we shall consider the process of calculating and extrapolating the parameters of the trajectory of the obstacle as an object of observation.

The tasks of the filtration and extrapolation of the parameters of the trajectory of an observed object are stated as a problem of evaluating the state vector of a dynamic system, whose state equation corresponds to the nature of the movement. To solve this problem, [10] proposed using the recurrent algorithms, namely exponential smoothing, and Kalman filter. Given their simplicity, these tools are convenient to implement in computer systems.

In a general case, the sequence of values for bypassing the tracked object is represented as white noise with a mathematical expectation, equal to zero, and a variance of $\sigma^2_{(g_\theta)}$, $\theta=\{x,y,z\}$. In uniform, discrete measurements, the Kalman filter coefficients A_n, and B_n/T_0 are used as constant coefficients A=α and B=β, termed α, β filters. We shall represent a scheme of the filtration algorithm using the Kalman filter's constant coefficients, α, β-filters [7], for one separate coordinate in the form of the following formulae:

$$\hat{x}_n=\hat{x}_{n^*}+\alpha\left(x_n-\hat{x}_{n^*}\right),$$

$$\hat{\dot{x}}_n=\hat{\dot{x}}_{n-1}+\beta/T_0\left(x_n-\hat{x}_{n^*}\right),$$

$$\hat{\dot{x}}_{n^*}=\hat{x}_{n-1}+\hat{\dot{x}}_{n-1}/T_0, \tag{2}$$

where n is the measurement step, n–1 is the preliminary measurement step, n* is the data extrapolated (predicted) per step n, x_n is the coordinate of the observed object measured at step n*, $\hat{x}_n$ is the calculated (predicted) coordinate at step n, $\hat{\dot{x}}_n$ is the speed of change in a coordinate in one step (period) of measurement T_0.

The algorithm defined by formulae (2) is a discrete automatic control system with feedback and constant smoothing coefficients α and β. Such a system has certain characteristics: transient process, stability, random and dynamic errors under the established mode of operation. Having considered the study of the filtration algorithm [10], we can determine:

– the conditions of stability (required and sufficient) of the α, β-filter take the following form:

$$\alpha > 0,\ \ \beta \geq 0,\ \ 2\alpha + \beta < 4;$$

– for most of the stability region, the transition process has an oscillatory or slightly damping character;

– the variance of random parameter filtering errors under a stable operation mode is determined from the following expressions:

$$\sigma^2_{\hat{x}_n} = \frac{2\alpha^2 - 3\alpha\beta + 2\beta}{\alpha(4 - 2\alpha - \beta)}\sigma^2_{x_n}, \qquad (3)$$

$$\sigma^2_{\hat{\dot{x}}_n} = \frac{2\beta^2}{T_0\alpha(4 - 2\alpha - \beta)}\sigma^2_{x_n};$$

– the variance of a random error in the extrapolation of a coordinate by one step:

$$\sigma^2_{\hat{x}_{n*}} = \frac{2\alpha^2 + \alpha\beta + 2\beta}{\alpha(4 - 2\alpha - \beta)}\sigma^2_{x_n};$$

– a dynamic error in the extrapolation of a coordinate by one step, due to the constant acceleration of the observed object g_M, is:

$$\Delta_d x_{n*} = g_M T_0^2 / \beta.$$

4. 3. Designing a computerized information processing system for building a movement trajectory in a dynamic environment

Consider a system of automated support to moving objects in the form of a UAV structure or other remotely controlled mobile complex. The structure, in a general case, includes the following:

– an observed object;

– an operator's control panel;

– the platform and moving turret of the mobile complex;

– the units mounted on them, intended for acquiring data on the state of the system and the external environment (optical modules, accelerometers, gyroscopic devices, etc.);

– data processing and signal generation units (control unit);

– system state change units (displacement drives of the mobile complex in space: chassis, engines, turret suspensions, etc.).

The use of manual and semi-automated control in such systems requires the construction of a new algorithm for the automated calculation of movement parameters for an observed object employing methods of forecasting and extrapolation.

4. 4. Algorithms for planning a movement trajectory in a dynamic environment taking obstacles into consideration

Studies that model the process of training automatic agents often employ simulation methods. These methods make it possible to monitor the state and actions at every step. This approach can be used to process information in order to build an optimal movement trajectory, in order to accompany objects (targets), in order to bypass obstacles in the static or dynamic environment, etc. There is also the issue of cloning behavior so that the agent can make the right decision regarding its future actions. It is proposed to use neural networks and a modern trajectory planning algorithm, using cloning the behavior of the agent moving through obstacles to reach the final coordinate of the initial trajectory [31]. For a convenient choice of changing the position of the agent, it is proposed to split the movement trajectory into smaller sections [32]. To improve the ability to process information when controlling UAC, it is proposed to apply methods based on neural networks [33], in order to effectively avoid obstacles [34].

The hypothesis, put forward, assumes that the accuracy of forecasting and performance of the model could depend on losses during training. Therefore, it is proposed to use the convolutional neural network (CNN) models, as well as the combinations with the recurrent long short-term memory (LSTM) neural network, CNN+LSTM.

5. Implementing the construction of a computerized information processing system to solve the task of planning UAV movement trajectory

5. 1. Results of developing a machine learning method using the Q-Learning algorithm

The approach described above regarding the development of machine learning methods can be represented as a sequence of the following steps:

1) initialize a table of values for Q and the current value for the $Q(s,a)$ function;

2) monitor the current state, s;

3) select an action, a, for a given state based on one of the action selection policies;

4) execute the action taking into consideration the reward, r, as well as the new state, s_k;

5) update the Q value for the observed state, using the reward, as well as the maximum possible reward for the next state;

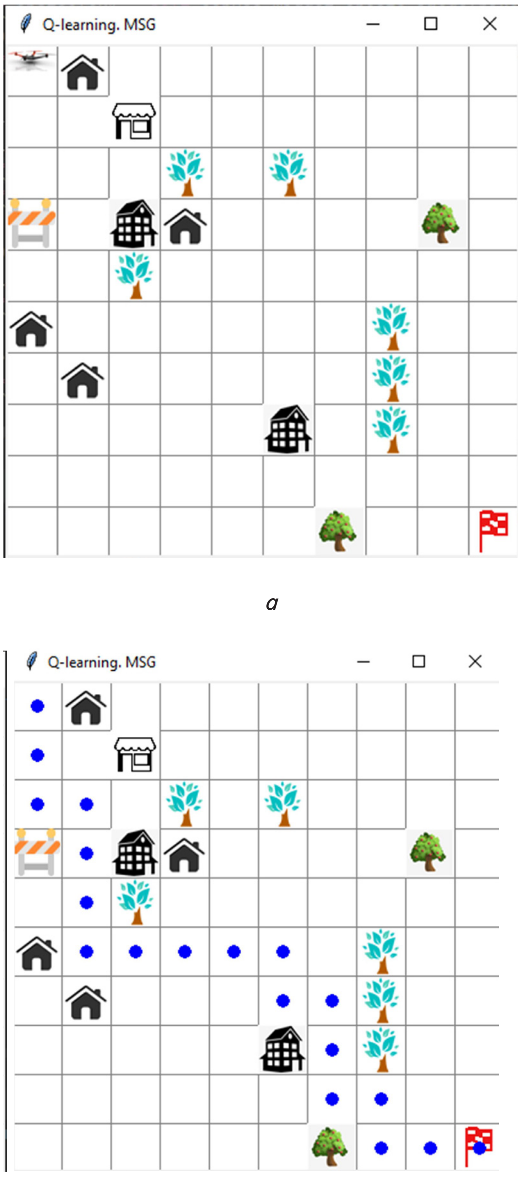
6) set a new state, and, if the final value is not reached, then return to step 2; if the final value is reached – complete the operation of the algorithm.

A function to select the agent's action is recorded using the Python programming language:

```
def choose_action(my, looks):
        my.check_state_exist(looks)
        if np.random.uniform() < my.epsilon:
                state_action=my.q_table.loc[looks, :]
                state_action=state_action.reindex (np.random.permutation(state_action.index))
                action = state_action.idxmax()
        else:
                action=np.random.choice(my.actions)
return action
```

*The results of training using the Q-Learning algorithm***.**

The results of executing a program to create an interference environment are shown in Fig. 2, *a***.** The environment, or a field, is divided into smaller areas, the movement trajectory is drawn from the coordinate of each intermediate area to the coordinate of the next intermediate area until the path is fully completed. The experimental result of training a neural network using the Q-Learning algorithm on a field of size 10 by 10 with the laying of a movement trajectory from the specified initial coordinate (an upper left angle) to the specified final coordinate (a flag) is shown in Fig. 2, *b*.



a

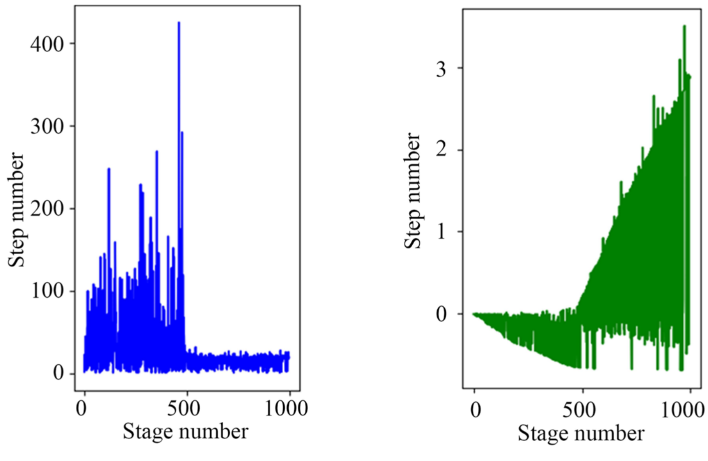
b

Fig. 2. Execution of training using the Q-Learning algorithm: a — creation of the environment; b — trajectory that bypasses obstacles

An important characteristic in the implementation of machine reinforcement learning methods is the construction of a chart illustrating the system state. Such a chart displays a set of all possible system states and the importance of the system's response to various actions. At each attempt to move along a specified trajectory through the created environment, UAV learns to bypass obstacles, and moves from one area to the next and, as a result, a trajectory is built taking into consideration obstacles to the destination coordinate.

The results of the efficient operation of the Q-Learning algorithm to train a neural network are shown in Fig. 3, ***a***. Training is considered depending on the number of steps required at a certain stage. In Fig. 3, *b*, each stage considered depends on receiving a reward at this stage.

To determine the shortest path, Table 1 was built by software to give the corresponding target values resulting from the execution of the program.



a

b

Fig. 3. The number of steps and rewards received at each stage of training a neural network using the Q-Learning algorithm: *a* — the dependence of stages on steps; *b* — the dependence of stages on a reward

Table 1

Value of the shortest movement trajectory for the created environment

Coordinate	Move up	Move down	Move right	Move left
[0.0, 40.0]	2.183082e-11	1.015156e-08	1.625024e	9.736734e-11
[0.0, 80.0]	7.407652e-11	–5.433903e-01	5.036295e-08	3.755632e-10
[40.0, 80.0]	1.392400e-11	2.563086e-07	1.347848e-11	2.886692e-10
[40.0, 120.0]	3.491871e-09	1.184092e-06	–3.573884e-01	3.888828e-01
[40.0, 160.0]	2.414627e-08	5.154729e-06	4.070336e-01	2.341310e-10
[40.0, 200.0]	3.652324e-08	–2.221786e-01	2.322062e-05	1.224790e-01
[80.0, 200.0]	–1.312542e-01	2.288132e-07	1.019095e-04	1.208706e-08
[120.0, 200.0]	1.406879e-10	4.066968e-04	1.705528e-06	1.021977e-06
[120.0, 240.0]	1.181061e-06	2.978240e-09	1.600272e-03	1.430772e-07
[160.0, 240.0]	5.288286e-07	1.057878e-07	5.549060e-03	2.735345e-05
[200.0, 240.0]	1.479461e-05	–1.224790e-01	1.612705e-02	9.514968e-05
[240.0, 240.0]	4.938648e-08	4.298275e-02	–1.136151e-01	2.199097e-05
[240.0, 280.0]	7.717672e-04	1.050327e-01	–4.900995e-02	–1.136151e-01
[240.0, 320.0]	1.659431e-03	-3.940399e-02	2.275248e-01	1.535019e-04
[280.0, 320.0]	–5.851985e-02	4.266747e-01	1.157103e-03	6.044271e-03
[280.0, 360.0]	2.321957e-02	3.268646e-02	6.795845e-01	–1.046617e-01
[320.0, 360.0]	3.129242e-03	5.094010e-02	9.323556e-01	2.482690e-02

Table 1 gives the summary values of the final trajectory: the coordinates of changing the position of the agent with the values of the shortest route in the movement environment. Based on the obtained values given in Table 1, one can see a decision made on the next action of the agent (UAV, RPAS). The sequence of movements as a result of training the neural network using the Q-Learning algorithm is as follows: down-right-down-down-down-down-right-right-right-down-down-right-

right-right. A given trajectory is the shortest path between two specified coordinates within the created environment. The result of this algorithm operation has established that for training a neural network in a specified environment, the smallest number of steps required to move along a specified trajectory is 18, the largest number is 273 steps. A given algorithm was implemented to work in the static (unchanging) environment but it is also possible to execute it in a dynamic environment if the position of observed objects and/or obstacles changes before the onset of training or after the completion of movement.

5. 2. Results of simulating a system for the video support to moving objects

After analyzing the above expressions, we can conclude that the effect of measurement errors can be reduced when selecting lower values for the coefficients α and β, but, to reduce the dynamic errors, the values for these coefficients should be chosen large enough. Accordingly, it is necessary to choose the best option that would ensure the proper smoothing of measurement errors and a sufficient speed of response to an observed object's maneuver.

It is possible to determine the optimal values for α and β according to the criterion of a minimum error of one step of coordinate extrapolation:

$$\sigma_{\hat{x}^*}^{(\alpha+\beta)} = \left(\sigma_{\hat{x}^*}^{(\alpha,\beta)} + \Delta_{g_M}^{(\beta)} \hat{x}_{n^*}\right) = \min_{\alpha,\beta}.$$

To minimize the dynamic error of coordinate extrapolation, the following ratio is proposed in [10]:

$$\beta = \alpha^2/(2-\alpha). \qquad (4)$$

The α and β values should be selected to meet the requirement for a specified probability at which a moving object is captured by the field of view of the video support system. In this case, the frame size should be chosen so that the number of false marks is minimal. The condition for the frame to be captured by the video support is written down in the following form:

$$g_M T_0^2/\beta + c\sigma_{\hat{x}} \le L/2, \qquad (5)$$

where L is the width of the frame side; c is the reliability factor, typically taken to be equal to 2–3.

By equating the left-hand and right-hand parts in expression (5) and by dividing them by an average quadratic measurement error σ_{x_n}, we obtain:

$$\frac{g_M T_0^2}{\sigma_{x_n}} = \left(\frac{L}{2\sigma_{x_n}} - \frac{c\sigma_{\hat{x}_n}}{\sigma_{x_n}}\right)\beta.$$

Considering (3) and (4), we obtain:

$$\frac{g_M T_0^2}{\sigma_{x_n}} = \left(\frac{L}{2\sigma_{x_n}} - c\sqrt{\frac{6\alpha-5\alpha^2}{8-8\alpha+\alpha^2}}\right)\frac{\alpha^2}{2-\alpha}.$$

Results of calculating the obstacles captured by the video support frame.

We modeled a system for the video support to moving objects by using Construct 2. Examples of the results from modeling and tracking moving objects are shown in Fig. 4.

Fig. 5 shows the $g_M T_0^2/\sigma_{x_n}$ dependence charts on the coefficient α for different $q = L/2\sigma_{x_n}$ parameter values at c=2 (Fig. 5, *a*) and c=3 (Fig. 5, *b*).

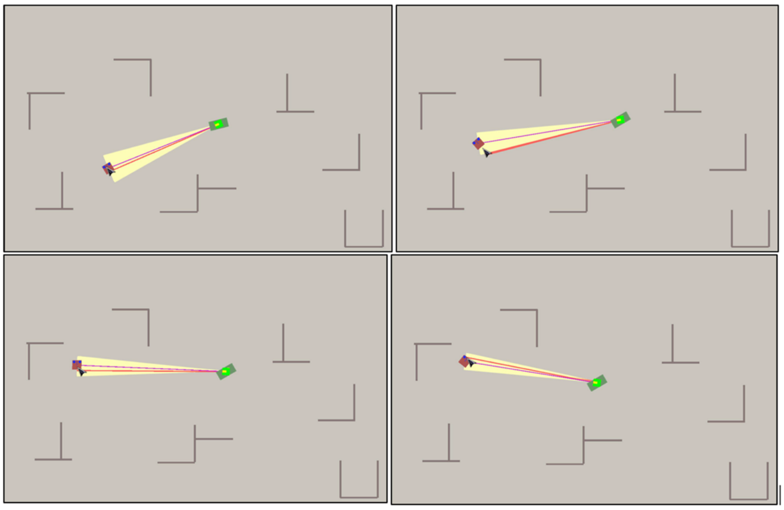
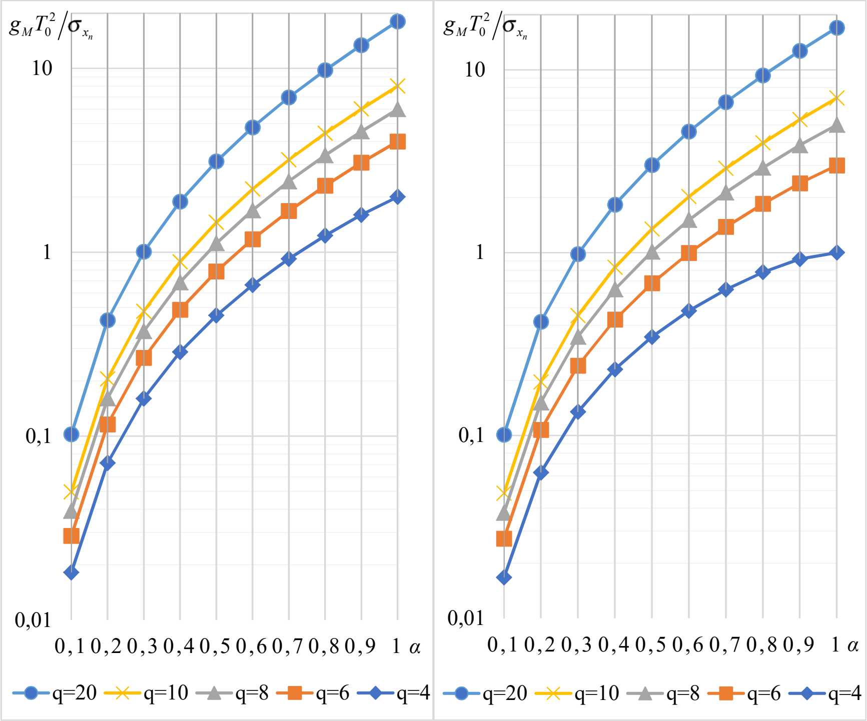


Fig. 4. Examples of the operation of a system for the video support to moving objects



a

b

Fig. 5. Dependence charts $g_M T_0^2/\sigma_{x_n}$ on the coefficient α for different values of $q = L/2\sigma_{x_n}$ parameter and the reliability coefficient *c*: *a* – reliability coefficient *c*=2; *b* – reliability coefficient *c*=3

Knowing the period of acquiring frames T_0 and the normalized frame size q, these charts make it possible to determine the value of α as a function of the maneuver intensity g_M, and to apply the selected coefficient to calculate the coordinate of the observed object in the next step.

5. 3. Results of developing a structural scheme of the computerized information processing system and a step-by-step algorithm for calculating the parameters of movement of the observed object

The results of developing a structural scheme are shown in Fig. 6. In this case, the object of observation is not only an obstacle that occurs along the movement trajectory but also other arbitrary targets.

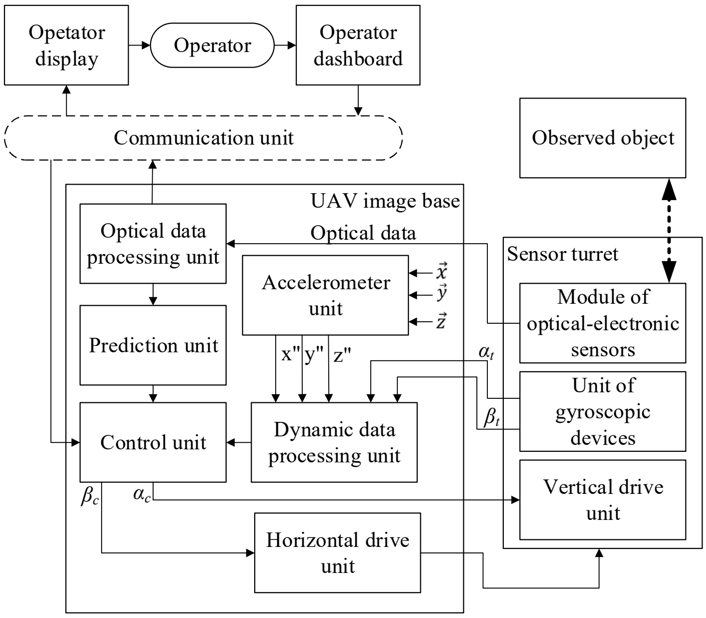


Fig. 6. System of automated support to moving objects based on a mobile complex

Fig. 6 shows that the complex includes an operator's control panel, the units and elements of a communication infrastructure, the turret itself. It shows the units of the computerized forecasting and control system, data processing units, the drive of the horizontal control for the observation system's turret. The turret hosts a vertical control drive unit and an optical-electronic detector module.

Under a semi-automatic mode, the line of vision is stabilized and maintained (compensation for swings and turns) using data acquired from gyroscopic devices and accelerometers while guidance is performed manually by the operator.

Under an automated mode, in addition to data from the unit of gyroscopic devices and accelerometers, data from the forecasting unit is used. Based on these data, the parameters of the movement of the observed object are calculated, used to correlate the coordinates of object (2).

Algorithm for calculating coordinates for a moving observed object.

Data on the displacement of the observed object or obstacle, over a single period of observations, is correlated with data from the forecasting unit. Based on the results of calculated deviations, rotation signals are generated for the horizontal and vertical turret drives for aligning the vision line.

The calculation of movement parameters of an observed object is carried out according to the following algorithm:

1) determine the coordinates of the required key points of the observed object in the frame of the optical module;

2) calculate the position of the specified points for further tracking of the object;

3) forecast the displacement of the object by one step and generate control signals for guiding the horizontal and vertical drives;

4) update data from the optical module, compare the offset of the selected points with the predicted indicators. When determining an observed object in the frame – proceed to point 2, in the absence – point 5;

5) extrapolate the predicted data to generate control signals for the horizontal and vertical drives for the next step;

6) increment the counter of loss of the observed object from the field of view, when reaching 0, proceed to point 7, otherwise – point 4;

7) generate the object loss signal and set the system to a semi-automated control mode.

5. 4. Results of developing an algorithm for planning the UAV movement trajectory in a dynamic environment

The process of data acquisition and algorithm construction using LSTM was reviewed and described in paper [15]. CNN consists of three main layers: a convolutional layer, a merged layer, and a fully bound layer with a linear activation function. Consider analyzing one-dimensional data, meaning CNN would need one-dimensional data arranged in order of consecutive moments of time. The hybrid network consists of CNN layers and LSTM layers. The source data are passed to the next level of LSTM.

To illustrate the generated path, write the following code:

```
import matplotlib.pyplot as plt
  plt.rcParams[“font.family”]=”Times New Roman”
  plt.rcParams[“font.size”]=14
  fig, ax=plt.subplots(figsize=(6,6))
  plt.xlim(0.15,1.1)
  plt.ylim(0.15,1.1)
  plt.scatter(starting_point[0],starting_point[1],
c=”r”)
  plt.scatter(ending_point[0],ending_point[1],
c=”b”)
  for i in center:
  ax.add_artist(plt.Circle((i[0],i[1]),obstacle_size,
color=”y”))

rrt=RRT(starting_point,ending_point,step_size=0.1,
velocity_constraint=True)
flag,p_archive=rrt.planning(1000,all_point=False)

print(flag)
p_archive=np.array(p_archive)
ax.scatter(p_archive[1:1,0],p_archive[1:1,1],s=3,c=”k”,z
order=10);
for i in range(p_archive.shape[0]-1):
plt.plot(p_archive[i:i+5,0],p_archive[i:i+5,1],c=”r”)
```

The result of the algorithm operation is shown in Fig. 7.

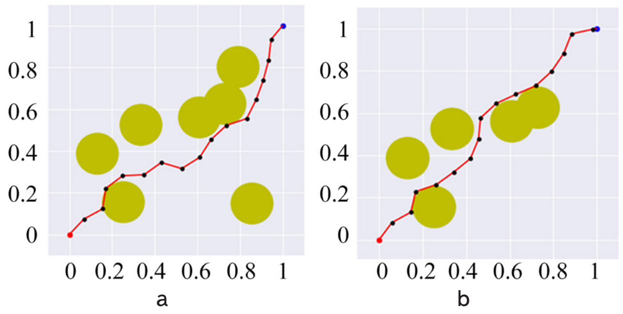


Fig. 7. Visualization of the redevelopment of movement trajectory when changing the coordinates of obstacles or in the presence of moving objects: *a* — initial trajectory; *b* — rebuilt trajectory

When using CNN only, the success rate reaches 74 %; during the hybrid application of CNN+LSTM, it amounts to 92 %.

It can be assumed that when using LSTM, productivity increases due to the fact that the agent can plan in advance and know how to avoid the most likely case of failure. And in a model without the use of LSTM, it stumbles upon an obstacle when it is very close to the final coordinate of the movement trajectory, but, at the same time, there is no collision with an obstacle.

6. Discussion of results of designing the components for a computerized information processing system

Fig. 2, *a* shows that the neural network is trained in the first 450 stages with a series of steps at one stage from several units to 270. The number of steps in the stage changes randomly. When training the network, after stage 450 and afterward the changes in the number of steps at one stage occur with a slight deviation. The same dependence can be traced in Fig. 2, *b*: after completing the first 450 stages of training, the reward increases dramatically. We can conclude that the neural network is trained for the selected environment in 450 stages with the smallest number of steps at one stage of 18, and the largest – 273.

Fig. 4 shows that the described and simulated model of the video support to moving object makes it possible not to lose the object of tracking from the video observation field. In other words, the described mathematical model involving the recurrent algorithms, exponential smoothing, and a Kalman filter allows the simulation of the process of observing a moving object. However, the effect of measurement errors can be reduced by selecting smaller coefficient values for the Kalman filter (Fig. 5, *a*, *b*), and, to reduce dynamic errors, the values of these coefficients should be large enough.

A new functional scheme has been developed, taking into consideration the additionally included functional forecasting unit, shown in Fig. 6, which makes it possible for the computerized system to independently decide on supporting moving objects.

The result of the program operation, shown in Fig. 7, *a*, *b*, demonstrates that the created agent successfully bypasses obstacles, automatically rebuilds the movement trajectory in case of changes in the position of obstacles. The success rate using LSTM is greater than that without LSTM: 92 % vs. 74 %. It can be assumed that the increased performance is due to the fact that LSTM can build an environment map implicitly and the agent can plan in advance and know how to avoid the most likely failure.

A computerized system has been developed that takes into consideration the influence of destabilizing factors, as well as increases the speed of information processing. That was achieved by using a combination of different algorithms to process information, the modern Python programming language, and the Construct 2 simulation environment.

Works [8, 9] examine data processing methods designed for stationary systems. When using ANN, for example, in work [22], it is not taken into consideration that the input data of obstacles may be unknown and constantly changing. Works [24, 25] describe the solution to a similar problem using an inverse problem, which leads to a decrease in accuracy in data transmission and the speed of information processing. Unlike the above, the computerized system proposed in this work can work in real time when the UAV or the robotic system itself move, with high accuracy and speed of information processing.

The proposed method refers to the processing of information based on the mathematical model of UAV motion control [18, 19] and the methods of processing information described in [20]. The input factors and a change in the coordinates of moving obstacles were taken into consideration. The proposed solution makes it possible to improve accuracy, performance, and maintain continuous communication in real time.

The limitations of this work are the unconsidered protection of information during transmission; the range of such a computerized system is limited to 15 km. In addition, the carrying capacity and natural factors were not taken into consideration: wind, icing, and others.

In the future, it is planned to add the consideration of wind strength and other weather conditions to the experiment. From a methodical point of view, it is planned to take into consideration the stability of movement.

7. Conclusions

1. It has been proposed to use machine reinforcement learning methods, in particular, the Q-Learning algorithm, to build a trajectory between two specified destination points with the possibility to bypass fixed obstacles. The result of the work is a program created to train a neural network, to calculate the UAV movement along a specified trajectory bypassing the obstacles. The effectiveness of the Q-Learning algorithm for training was also shown, depending on the number of steps in the stages of learning and on the rewards received. To determine the shortest path, a final table was compiled demonstrating the results of program execution when finding the optimal movement trajectory in a static environment. The constructed table makes it possible to determine the optimal solution for the next action of the agent.

It was determined that the smallest number of steps required to move along a specified trajectory is 18, and the largest – 273 steps.

2. We have simulated a system for the video support to moving objects. The obstacle motion parameters were calculated using the exponential smoothing and a Kalman filter. A hypothesis has been verified that the effect of errors can be reduced when choosing lower values of coefficients α

and β but, to reduce the dynamic errors, the values for these coefficients must be chosen large enough. Accordingly, it is necessary to select the best option, which would ensure sufficient smoothing of measurement errors, as well as the sufficient speed, heel, and acceleration of response to the observed object's maneuver.

Examples of the operation of the simulated system for the video support to moving objects have been provided. Dependence charts were built of the maneuver intensity function on the coefficient α for different parameter values of the normalized frame size and reliability coefficient.

The algorithm has been proposed, accounting for the coordinates of a moving object of observation, in particular obstacles.

3. A structural scheme of the computerized information processing system for the construction of a movement trajectory in a dynamic environment has been built. In addition to the typically accepted components, the developed scheme was supplemented with an optical data processing unit and a forecasting unit. The optical data processing unit is used to highlight key points in an image and determine the position of the observed object. Data from the unit are sent to the operator's screen and to the forecasting unit. The forecasting unit extrapolates the coordinates of moving objects and predict the displacement of the observed moving object. A step-by-step algorithm has been developed for calculating the parameters of movement of the observed object.

4. We have planned the UAV movement trajectory in a dynamic environment, taking into consideration obstacles, by using the CNN neural network, and the CNN+LSTM hybrid. A program for the proposed implementation has been created. When using CNN only, the success rate reaches 74 %, and, with the hybrid application of CNN+LSTM, it amounts to 92 %.

References

1. Lebedev, H. N., Mirzoian, L. A. (2005). Neiromerezheve planuvannia diy po oblotu nazemnykh obiektiv hrupoiu litalnykh aparativ. Aviakosmisne pryladobuduvannia, 12, 41–47.
2. Yakovlev, K. S., Baskin, E. S. (2013). Graph models for solving 2D path finding problems. Iskusstvennyi intellekt i prinyatie reshenii = Artificial intelligence and decision making, 1, 5–12.
3. De, L., Guglieri, G. (2012). Advanced Graph Search Algorithms for Path Planning of Flight Vehicles. Recent Advances in Aircraft Technology. doi: https://doi.org/10.5772/37033
4. LaValle, S. M. (2011). Motion Planning. IEEE Robotics & Automation Magazine, 18 (1), 79–89. doi: https://doi.org/10.1109/mra.2011.940276
5. Lee, D., Shim, D. H. (2014). RRT-based path planning for fixed-wing UAVs with arrival time and approach direction constraints. 2014 International Conference on Unmanned Aircraft Systems (ICUAS). doi: https://doi.org/10.1109/icuas.2014.6842270
6. Gonsales, R., Vuds, R. (2012). Tsifrovaya obrabotka izobrazheniy. Moscow: Tekhnosfera, 1104.
7. Alpatov, B. A., Babayan, P. V., Balashov, O. E., Stepashkin, A. I. (2008). Metody avtomaticheskogo obnaruzheniya i soprovozhdeniya obektov. Obrabotka izobrazheniy i upravlenie. Moscow: Radiotekhnika, 176.
8. Lakota, N. A. (Ed.) (1978). Osnovy proektirovaniya sledyashchih sistem. Moscow: Mashinostroenie, 392.
9. Malyshev, V. V., Krasil'shchikov, M. N., Karlov, V. I. (1989). Optimizatsiya nablyudeniya i upravleniya letatel'nyh apparatov. Moscow: Mashinostroenie, 312.
10. Kuz'min, S. Z. (2000). Tsifrovaya radiolokatsiya. Vvedenie v teoriyu. Kyiv: KVITS, 428.
11. Besekerskiy, V. A., Popov, E. P. (2003). Teoriya avtomaticheskogo upravleniya. Sankt-Peterburg: Professiya, 752.
12. Seydzh, E. P., Melsa, Dzh. L. (1974). Identifikatsiya sistem upravleniya. Moscow: Nauka, 248.
13. Taha, H. A. (2005). Vvedenie v issledovanie operatsiy. Moscow: Vil'yams, 912.
14. Dorf, R., Bishop, R. (2002). Sovremennye sistemy upravleniya. Moscow: Laboratoriya Bazovyh Znaniy, 832.
15. Graf, M., Kvasnikov, V. (2018). The Construction of the Algorithm Study Based on the Mathematical Model of Motion. ICTERI 2018, 235–242. Available at: http://ceur-ws.org/Vol-2105/10000235.pdf
16. Hraf, M. S., Ihnatenko, P. L. (2017). Analiz suchasnykh modelei obrobky informatsii ta keruvannia v bezpilotnomu povitrianomu sudni. VII Mizhnarodna naukovo-tekhnichna konferentsiya "Kompleksne zabezpechennia yakosti tekhnolohichnykh protsesiv ta system". Vol. 2. Chernihiv: ChNTU, 135–136. Available at: https://docplayer.net/70618517-Kompleksne-zabezpechennya-yakosti-tehnologichnih-procesiv-ta-sistem.html
17. Gordin, A. G., Bychkova, I. V., Kulik, A. S., Narozhniy, V. V. (2006). Problematika razrabotki perspektivnyh malogabaritnyh letayushchih robotov. Aerogidrodinamika: problemy i perspektivy, 2, 247–271.
18. Noth, A., Bouabdallah, S., Siegwart, R. (2006). Dynamic Modeling of Fixed-Wing UAVs. Swiss Federal institute of technology. Available at: http://www.sky-sailor.ethz.ch/docs/Dynamic_Modeling_of_Fixed-Wing_UAVs_(12.05.2006).pdf
19. Gordin, A. G. (2000). Bespilotnye letatel'nye apparaty kak obekty upravleniya. Kharkiv: Gos. aerokosm. un-t «Khark. aviats. in-t», 140.
20. Hraf, M. S. (2016). Analiz isnuiuchykh metodiv obrobky informatsiyi v blotsi keruvannia bezpilotnoho povitrianoho sudna. Bulletin of Engineering Academy of Ukraine, 4, 20–22.
21. Terekhov, V., Efimov, D., Tyukin, I. (2002). Neyrosetevye sistemy upravleniya. Moscow: Vysshaya shkola.
22. Haykin, S. (2008). Neural Networks: A Comprehensive Foundation. Moscow: Izdatel'skiy dom Vil'yams, 1104.

23. Goldberg, D. E. (1995). Algorytmy genetyczne i ich zastosowania. Warszawa: WNT.
24. Burdakov, S. F., D'yachenko, V. A., Timofeev, A. N. (1986). Proektirovanie manipulyatorov promyshlennyh robotov i robotizirovannyh kompleksov. Moscow: Vysshaya shkola, 264.
25. Pettré, J. (2013). Locomotion Synthesis for Digital Actors. Modeling, Simulation and Optimization of Bipedal Walking, 187–198. doi: https://doi.org/10.1007/978-3-642-36368-9_15
26. Vukobratovich, M., Shneyder, A. Yu., Gurfinkel', V. S. (1976). SHagayushchie roboty i antropomorfnye mekhanizmy. Moscow: Mir, 541.
27. Tertychniy-Dauri, V. Yu. (2012). Dinamika robototekhnicheskih sistem. Sankt-Peterburg: NIU ITMO, 128.
28. Lynen, S., Sattler, T., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R. (2015). Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. Robotics: Science and Systems XI. doi: https://doi.org/10.15607/rss.2015.xi.037
29. Barto, A. G. (1997). Reinforcement Learning. Neural Systems for Control, 7–30. doi: https://doi.org/10.1016/b978-012526430-3/50003-9
30. Sichkar, V. N. (2019). Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot. 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). doi: https://doi.org/10.1109/icieam.2019.8742915
31. Torabi, F., Warnell, G., Stone, P. (2018). Behavioral Cloning from Observation. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. doi: https://doi.org/10.24963/ijcai.2018/687
32. Graf, M. S., Kvasnikov, V. P. (2019). Information processing in the control system of an unmanned aerial vehicle. System Research and Information Technologies, 4, 59–65. doi: https://doi.org/10.20535/srit.2308-8893.2019.4.06
33. Carrio, A., Sampedro, C., Rodriguez-Ramos, A., Campoy, P. (2017). A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles. Journal of Sensors, 2017, 1–13. doi: https://doi.org/10.1155/2017/3296874
34. Gandhi, D., Pinto, L., Gupta, A. (2017). Learning to fly by crashing. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi: https://doi.org/10.1109/iros.2017.8206247