REINFORCEMENT LEARNING IMPLEMENTATION IN THE CONTROL SYSTEM FOR THE UNMANNED, COMPOUND HELICOPTER

Sara Waśniewska, Warsaw University of Technology, Warsaw, Poland
Sebastian Topczewski, Warsaw University of Technology, Warsaw, Poland

Abstract

The paper deals with the development of an automatic control system for an unmanned, small-scale compound helicopter, equipped with an additional pusher propeller located at the end of the tail boom. A linear quadratic regulator controls the classical part of the helicopter, while the pitch angle of the pusher propeller, at a given constant angular velocity, is controlled by two interchangeable controllers: a nonlinear controller based on the TD3 reinforcement algorithm and a linear, proportional controller. The latter is used to compare the performance of the control system with the nonlinear controller against the traditional, linear approach. Besides developing the control system, the paper demonstrates the possibility of applying the reinforcement learning to the control problem. Simulation results show that the preset forward velocity cannot be achieved without incorporating the thrust force of the pusher propeller. Furthermore, at higher flight velocities, the nonlinear controller better minimizes the velocity deviation by adjusting the pitch angle of the pusher propeller according to the current flight conditions.

1. INTRODUCTION

Unmanned aerial vehicles, due to their wide range of capabilities, have found applications not only in the military field, but also in an increasing number of scientific and industrial fields (Ref. 1). In particular, vertical take-off and landing (VTOL) aircraft, such as multirotorcraft and helicopters in classical and compound configurations, have attracted interest. The pivotal characteristics of rotorcraft are their nonlinear and complex dynamics, as well as the coupling between control channels, which significantly complicate the control of these objects. For small size and low mass aircraft, their sensitivity to disturbances such as wind gusts and control signals is increased, further complicating the control issue. In addition, for compound helicopters, compared to classical configurations, there is an additional need to consider the impact of added subsystems on aircraft performance and control approaches. Consequently, the effective design and development of control systems tailored to the characteristics of the control objects and the scope of their operations are indispensable elements in rotorcraft development.

Methods used in control systems can be categorized into linear and nonlinear techniques. Common linear approaches include PID and LQR controllers, as described in Ref. 2 and Ref. 3, respectively. Conversely, nonlinear methods can encompass the feedback linearization method shown in Ref. 4, the predictive control method proposed in Ref. 5, the gradient descent optimization method presented in Ref. 6, the adaptive Pi-Sigma neural network based method described in Ref. 7, and the fuzzy logic-based control method outlined in Ref. 8. An interesting approach is the use of reinforcement learning (RL) for the development of nonlinear, suboptimal controllers, as demonstrated in Ref. 9. A review of the literature indicates that much work has been done on the development of control systems for unmanned multirotor aircraft and classic helicopters, especially for hovering and for low-speed manoeuvre. However, hardly any work has been

encountered regarding the control of small, compound helicopters.

This paper is concerned with the development of a control system for a compound helicopter (classical configuration equipped with a pusher propeller), where the classical part of the aircraft is controlled using a linear quadratic regulator, and in parallel with which operates a dedicated pusher propeller controller. Compared to the conference paper from ERF 2023 (Ref. 10), which presented a proposal to implement a Deep Deterministic Policy Gradient (DDPG) reinforcement learning algorithm to control a whole unmanned helicopter, this paper not only provides a theoretical proposal for the use of one of the RL method in the control system for the compound helicopter's pusher propeller, but also describes the implementation of the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm, the development of the TD3 agent to control the pusher propeller, and the results of the simulations performed using the developed overall control system. To evaluate and compare the performance of the control system with the RL agent to the control system with proportional controller for the pusher propeller, numerical simulations are carried out. The test cases encompass the forward flight of the helicopter with different values of the preset velocities in the body coordinate system.

The following sections of the paper include description of the rotorcraft model used, the theoretical foundations of the LQR and RL algorithms. Next, the overall control system is introduced along with a detailed description of the implemented controllers. Finally, based on the results for the presented test cases, conclusions of the research are included.

2. ARCHER – COMPOUND HELICOPTER

The control object under consideration is the ARCHER compound helicopter. ARCHER is a small-scale, unmanned helicopter created at the Warsaw University of Technology as part of one of the projects (Ref. 11) that aimed to develop a modular lifting platform with the ability to easily attach various components in order to acquire different compound helicopter configurations and to serve as a test stand for future research.

In its physical version, the ARCHER helicopter was developed in a classical configuration, the basic parameters of which are shown in Table 1.

Table 1: Basic ARCHER Characteristics (Ref. 11).

Characteristic	Metric
Main rotor diameter	1,78 m
No. of MR blades	2
MR RPM	2'000
Tail rotor diameter	0,158 m
No. of TR blades	2
TR RPM	7'200
MR & TR rotorhub	Rigid, no flapping
TOW	8 kg
MR & TR blades airfoil	NACA23012
Propulsion	Electric

The numerical simulation model of the compound helicopter, developed in the Flightlab software, corresponds to a physical one in the classical configuration and additionally includes pusher propeller mounted on the tip of the horizontal tail boom. The two-blade main rotor is modeled using the blade element method. Its aerodynamic loads determined with a quasi-steady model are validated from ARCHER baseline flight data, and its induced velocity is determined with a Peters-He Six state model. The tail rotor and pusher propeller are modeled with two blades each, employing numerically more efficient disk theory to derive their loads. Moreover, due to the use of a rigid type for all rotors, the blade flapping is prevented. The fuselage is modeled as a rigid object with six degrees of freedom, with its aerodynamic characteristics determined by the CFD model. The control of the compound ARCHER model is executed by typical helicopter control variables for a classical configuration, including the main rotor longitudinal and lateral cyclic pitch angles, the main rotor collective pitch angle, the tail rotor collective pitch angle ($\mathrm{XB}, \mathrm{XA}, \mathrm{XC}$ and XP, respectively), and in the general case also the variable angular velocity and collective pitch angle of the pusher propeller ($\mathrm{X_{RPM}}$ and $\mathrm{X_{Theta}}$ respectively). The angular velocities of the main rotor and the tail rotor in the simulation model are fixed at 1000 RPM and 7200 RPM respectively. For the pusher propeller, it is assumed that the angular velocity is not controlled by the control system. $\mathrm{X_{RPM}}$ is predetermined and has a value of 7000 RPM.

THEORETICAL BACKGROUND OF CONTROL ALGORITHMS

2.1. Linear Quadratic Regulator

The linear quadratic regulator (LQR) is one of the basic methods of optimal control with feedback. LQR

requires knowledge of a linear model of the control object written in the form of a state equation (Eq. (1)):

$$\dot{X} = AX + Bu \tag{1}$$

where X is the state vector, u is the control vector, A is the state matrix, and B is the control matrix.

The operation of the LQR consist in determining the optimal control law, u, ensuring the minimization of the quadratic cost function, J (Eq. (2)):

$$J = \int_0^\infty ((X - X_{ref})^T Q\,(X - X_{ref}) + u^T R\,u)dt \tag{2}$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$ are respectively the state and control weighting matrices, and X_{ref} is the vector of the reference state variables.

The optimal control law, which is a linear feedback from the full state vector, is given by Eq. (3):

$$u = -K(X - X_{ref}) \tag{3}$$

The gain matrix, K, is definied as in Eq. (4):

$$K = R^{-1}B^T P \tag{4}$$

where P is the solution of the algebraic Riccati equation (Eq. (5)):

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{5}$$

Optimizing the value of J means bringing the deviation of the current state vector values, X, from the reference values, X_{ref}, to zero with minimum control effort. The LQR design process, mainly comes down to the selection of the weighting matrices Q i R.

2.2. Reinforcement Learning

Reinforcement learning, being one of the machine learning methods, is a computational approach to learning and decision-making related to a specific goal. RL is based on a dynamic interaction between the agent, which is the learning element in the RL algorithm, and the environment, which is the agent's operating space and the source of the data in the training process.

The general idea behind RL's working, visually shown in Figure 1, is as follows (Ref. 12): at each time step t, based on the observation of the current state of the environment, s_t, and the value of the immediate reward, R_t, for actions taken in the previous time step, a_{t-1}, the agent takes action according to the current policy, π, which is a relationship mapping observations to taken actions. The undertaken action changes the state of the environment that is observed by the agent at the next time step, $t + 1$, and, together with the next value of reward, R_{t+1}, constitute the basis for the subsequent action, a_{t+1}.

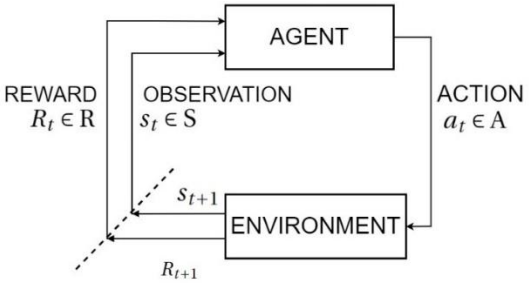

Figure 1: A scheme of the interaction between the agent and the environment in reinforcement learning

The immediate reward function, R, which needs to be maximized and which is defined by the designer of the RL algorithm adequately to the solved problem, takes into account the purpose of the agent's behavior. The value of the reward function provides a quantitative indication of the correctness of the performed actions and their impact on the state of the environment and is the only feedback to ensure that the learning process is correct.

An important aspect in the RL is the concept of the return, G, which is defined as the sum of consecutive, discounted, immediate rewards received from the environment: $G_t := \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$,. The discount factor, $0 \leq \gamma \leq 1$, ensures the trade-off between exploration and exploitation of the environment during the training phase. The goal of the agent is to learn the optimal policy, π^*, utilizing which the actions taken by the agent ensure that the maximum value of the expected return, G, is achieved.

More detailed description of the reinforcement learning, as well as the possibility of using RL as a control method, can be found in the paper presented at the ERF conference in 2023 (Ref. 10).

The specific RL method applied in this work is the TD3 algorithm, which belongs to the model-free, off-policy and actor-critic group of RL methods and uses neural networks as approximators of the TD3-specific strategies and value functions. The TD3 algorithm is an extension of the Deep Deterministic Policy Gradient (DDPG) algorithm and is developed to improve the performance of the DDPG algorithm, in particular to avoid overestimation of the value function and numerical instability that can occur in DDPG algorithm. A detailed description of the DDPG algorithm can be found in Ref. 10. The modifications introduced in TD3 compared to the DDPG are: the use of two Q-value

functions (and the use of the one with the smaller value in the critic updating rule), target policy smoothing, and less frequent updating of the strategy and target networks compared to the updating frequency of the value function approximator. A comprehensive explanation of the TD3 can be found in Ref. 13. The TD3 algorithm is shown in Table 2.

Table 2: TD3 algorithm (Ref. 13).

TD3 algorithm

1) Randomly initialize the neural network parameter vectors Θ_{Q1}, Θ_{Q2} and Θ_{π} for the critics' networks, Q_1 and Q_2, and actor's network, π, respectively

2) Initialize the target neural network parameter vectors $\Theta_{Q1}^{T} \leftarrow \Theta_{Q1}$, $\Theta_{Q2}^{T} \leftarrow \Theta_{Q2}$ and $\Theta_{\pi}^{T} \leftarrow \Theta_{\pi}$ for the critics' target networks, Q_1^{T} and Q_2^{T}, and actor's target network, π^{T}, respectively

3) Initialize experience buffer

for $t = 1 : T$ do

a) For the current observations s_t, select action $a_t \leftarrow \pi(s_t) + \mathcal{N}$, where $\mathcal{N}$ is exploration noise, $\mathcal{N} \sim N(0, \sigma)$

b) Execute action and observe new state s_{t+1} and reward R_{t+1}

c) Store transition $(s_t, a_t, R_{t+1}, s_{t+1})$ in replay buffer

d) Sample mini-batch of M transitions $(s_i, a_i, R_{i+1}, s_{i+1})$ from experience buffer

e) Compute:

$$\tilde{a_i} \leftarrow \pi^{T}(s_{i+1}) + \mathcal{N}$$
$$\mathcal{N} \leftarrow \mathrm{clip}(N(0, \sigma), -c, c)$$

f) If s_i is a terminal state, set $y_i = R_{i+1}$. Otherwise, set:

$$y_i \leftarrow R_{i+1} + \gamma \cdot \min\left\{Q_1^{T}(s_{i+1}, \tilde{a_i}; \Theta_{Q1}^{T}), Q_2^{T}(s_{i+1}, \tilde{a_i}; \Theta_{Q2}^{T})\right\} + \mathcal{N}$$

g) Update critics' neural network parameters Θ_{Qi}, Θ_{Q2} by minimizing the loss functions:

$$\Theta_{Q1} \leftarrow \arg\min_{\Theta_{Q1}} \frac{1}{M} \sum_{i=1}^{M} \left(y_i - Q_1(s_i, a_i)\right)^2$$
$$\Theta_{Q2} \leftarrow \arg\min_{\Theta_{Q2}} \frac{1}{M} \sum_{i=1}^{M} \left(y_i - Q_2(s_i, a_i)\right)^2$$

h) Every D_1 steps, where D_1 is determined by policy update frequency, update actor parameters Θ_{π} using the sampled policy gradient:

$$\nabla_{\Theta_{\pi}} J(\Theta_{\pi}) = \frac{1}{N} \sum_{i=1}^{M} \nabla_A \min\left\{Q_1(s_i, A; \Theta_{Q1}), Q_1(s_i, A; \Theta_{Q1})\right\} \cdot \nabla_{\Theta_{\pi}} \pi(s_i; \Theta_{\pi}), \qquad \text{where } A = \pi(s_i; \Theta_{\pi})$$

i) Every D_2 steps, where D_2 is determined by target update frequency, update the target networks:

$$\Theta_{\pi}^{T} \leftarrow \tau\Theta_{\pi} + (1 - \tau)\Theta_{\pi}^{T}$$
$$\Theta_{Q1}^{T} \leftarrow \tau\Theta_{Q1} + (1 - \tau)\Theta_{Q1}^{T}$$
$$\Theta_{Q2}^{T} \leftarrow \tau\Theta_{Q2} + (1 - \tau)\Theta_{Q2}^{T}$$

end

3. CONTROL SYSTEM DESCRIPTION

3.1. Description of the overall control system

The control system of the ARCHER compound helicopter model is developed in Matlab/Simulink and is schematically shown in Figure 2. Due to the fact that the rotorcraft nonlinear model is developed in Flightlab, the flow of signals between the two mentioned programs is ensured during the simulation at each time step. In the control system implemented in Simulink, the input signals to the block representing the nonlinear helicopter model are control signals calculated using the developed control laws. When the control is applied, the new state of the helicopter is calculated in Flightlab, and the state vector information is output from the block representing the ARCHER helicopter and is used to develop control signals in the next time step.

The used vector of state variables, $\mathrm{X} = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \varphi, \theta, \psi, \mathrm{Vx}, \mathrm{Vy}, \mathrm{Vz}, \mathrm{p}, \mathrm{q}, \mathrm{r}]$ consists of position coordinates in the inertial coordinate system $(\mathrm{x}, \mathrm{y}, \mathrm{z})$, orientation angles (φ, θ, ψ), linear velocities in the body system $(\mathrm{Vx}, \mathrm{Vy}, \mathrm{Vz})$ and angular rates $(\mathrm{p}, \mathrm{q}, \mathrm{r})$.

The overall vector of control variables, $\mathrm{u} = [\mathrm{XB}, \mathrm{XA}, \mathrm{XC}, \mathrm{XP}, \mathrm{X_{RPM}}, \mathrm{X_{Theta}}]$ consists of both the control typical of the classical helicopter configuration, that is, the percentage positions of the main rotor longitudinal cyclic pitch stick (XB), main rotor lateral cyclic pitch stick (XA), main rotor collective pitch stick (XC) and tail rotor collective cyclic pitch stick (XP), as well as the variable parameters of the pusher propeller: pitch angle $(\mathrm{X_{Theta}})$ and angular velocity $(\mathrm{X_{RPM}})$. The first four elements of the vector u are expressed as a percentage in the range from 0 to 100. The parameters of the pusher propeller are also limited: the pitch angle has values from -10^{o} to 25^{o}, while the angular velocity can take values from 100 to 10000 RPM.

In the overall control system two subsystems can be distinguished: the autopilot (AP) and the stability augmentation system (SAS).

The AP subsystem consists of the main controller, responsible for controlling the classical part of the helicopter, and the pusher propeller controller, which operates in parallel with the first one. The tasks of the AP subsystem include the determination of control signal, $\mathrm{u_{AP}} = [\mathrm{XB_{AP}}, \mathrm{XA_{AP}}, \mathrm{XC_{AP}}, \mathrm{XP_{AP}}, \mathrm{X_{RPM}}, \mathrm{X_{Theta}}]$, based on: the developed control laws, information about the full vector of state variables, the gain matrix K, the reference values of longitudinal, linear velocity $(\mathrm{Vx_{ref}})$ and lateral and vertical position $(\mathrm{y_{ref}}, \mathrm{z_{ref}})$.

The main controller is a LQR, which, based on the differences between the observed and reference values of the state variables, generates four control signals: $\mathrm{XB_{AP}}$, $\mathrm{XA_{AP}}$, $\mathrm{XC_{AP}}$, $\mathrm{XP_{AP}}$. To control the variable value of the pusher propeller pitch angle, two controllers are developed, for which the common and main input quantity is the difference between the reference and the current linear, longitudinal velocity in the body system, $\mathrm{Vx_{ref}} - \mathrm{Vx}$, and the output signal is the pitch angle of the pusher propeller, $\mathrm{X_{Theta}}$.

The purpose of using the SAS subsystem is to improve helicopter stabilization by damping angular velocity oscillations using three proportional controllers with zero reference angular rates. The output signal from the SAS subsystem is the vector $\mathrm{u_{SAS}} = [\Delta\mathrm{XB}, \Delta\mathrm{XA}, 0, \Delta\mathrm{XP}, 0, 0]$ composed of the increments of the control variables: $\Delta\mathrm{XB}$, $\Delta\mathrm{XA}$, $\Delta\mathrm{XP}$ corresponding to the rates p, q and r, respectively.

Additionally, a trim of the aircraft in the hover condition is performed prior to the start of the simulation and the resuting control $\mathrm{u_0} = [\mathrm{XB_0}, \mathrm{XA_0}, \mathrm{XC_0}, \mathrm{XP_0}, 0, 0]$ is included in the control system.

Finally, the overall control signal, u, that is the input signal to the block representing the helicopter model in Flightlab, is the sum of the control variables from the AP and SAS subsystems and the trim condition (Eq. 6):

$$\mathrm{u} = \mathrm{u_{AP}} + \mathrm{u_{SAS}} + \mathrm{u_0} \tag{6}$$

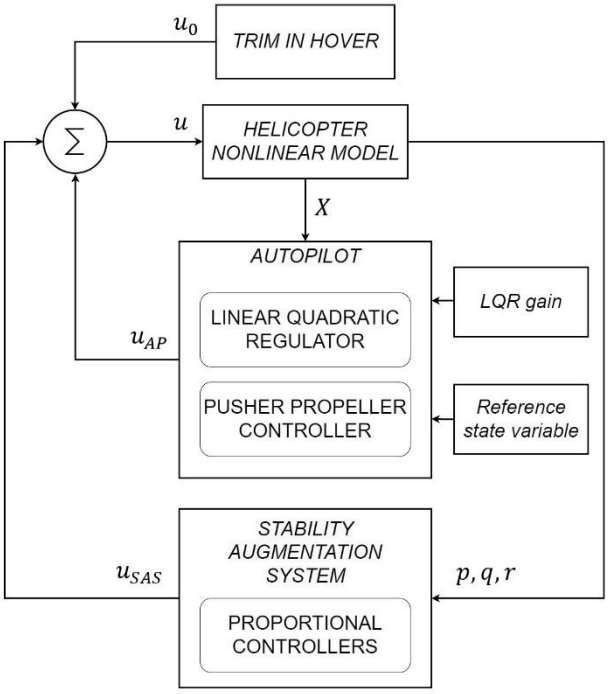

Figure 2: A scheme of the overall control system of the ARCHER compound helicopter

3.2. Implementation of the LQR controller

The linear quadratic regulator is made in Matlab, and the necessary linear model of the control object is obtained by linearizing the nonlinear model of the helicopter in classical configuration in hover state using Flightlab. The static gain matrix, K, used in the LQR control law is generated prior to the start of the simuation, is constant and the same for all performed simulation. The state vector used in the linearised model is identical to the state vector X of the overall control system. In turn, the vector of control variables associated with the LQR is four element and includes only the control variables corresponding to the classical ARCHER configuration.

3.3. Proportional controller implementation

The proportional pusher propeller controller is designed strictly to compare the performance of the control system using the main TD3 algorithm based nonlinear pusher propeller controller with the performance of the proportional, linear control method.

On the basis of the input signals to the controller, which are the deviations of the actual linear forward velocity from the reference values, there are generated control signals, X_{Theta} – the values of the pusher propeller pitch angle (Eq. 7). X_{Theta} is made up of two components expressed in degrees, where the first, $\mathrm{X_{Theta\ C}}$, is predetermined and depends only on $\mathrm{Vx_{ref}}$ (Eq. 8), and the second, $\mathrm{X_{Theta\ V}}$, is an adjustable increment of the control variable, proportional to the velocity deviation (Eq. 9).

(7) $$\mathrm{X_{Theta} = X_{Theta\ C} + X_{Theta\ V}}$$

(8) $$\mathrm{X_{Theta\ C}} = \begin{cases} 10, & \mathrm{Vx_{ref}} \geq 1\,\frac{\mathrm{ft}}{\mathrm{s}} \\ 10 \cdot \mathrm{Vx_{ref}}, & \mathrm{Vx_{ref}} < 1\,\frac{\mathrm{ft}}{\mathrm{s}} \end{cases}$$

(9) $$\mathrm{X_{Theta\ V}} = 1 \cdot (\mathrm{Vx_{ref}} - \mathrm{Vx})$$

3.4. Implementation of the RL algorithm-based pusher controller

The development of the TD3 algorithm and the execution of the RL agent learning process are carried out in Matlab/Simulink with the *Reinforcement Learning Toolbox* and *Deep Learning Toolbox*.

The RL agent is intended to serve as a pusher propeller controller, which is considered during the selection of observation and action signals.

The observation signal consists of six elements, where the inclusion of each element is intended to improve the agent's learning process and its subsequent use to control the additional helicopter component. The first three elements of the observation signal are based on the difference between the reference and current forward velocity in the body system (velocity deviation, $\mathrm{Vx_{ref}} - \mathrm{Vx}$). The three elements are, respectively, the value of the velocity deviation, the integral of the velocity deviation and the derivative of the velocity deviation. The presented choice is supposed to improve the quality of the agent's performance by allowing consideration of the dynamics changes of the velocity deviation. The fourth and fifth observed quantities are the current angular velocity and pitch angle of the pusher propeller, respectively. The last parameter is the preset, reference velocity of the helicopter in the longitudinal direction. The last three elements of the observation vector are intended to ensure that the controlled pitch of the pusher propeller is adjusted not only according to the velocity deviation, but also takes into account the actual values of the variable parameters of the pusher propeller and the actual value of the reference velocity.

In the case of the control system with the RL agent, the continuous actions taken by the agent within the range of -10^{o} to 25^{o} are equivalent to the control signal $\mathrm{X_{Theta}}$, i.e. the pusher propeller pitch angle.

The purpose of using the pusher propeller is to relieve the main rotor in the context of generating the required thrust force and to ensure the ability to achieve the desired forward velocity, which cannot be fully achieved without the additional thrust force. Moreover, it is necessary that the additional component control does not degrade the effectiveness of the overall control system and does not generate excessive control effort. Therefore, the immediate reward function (Eq. (10)) used and maximized in the training process consists of the following elements given by equations (11), (12) and (13):

(10) $$\mathrm{R_t = R_{Vx,t} + R_{X,t} + R_{Theta,t}}$$

(11) $$\mathrm{R_{Vx,t} = -50 \cdot \left|Vx_{ref,t-1} - Vx_{t-1}\right|}$$

(12) $$\begin{aligned} \mathrm{R_{X,t}} = &-\left|\mathrm{y_{ref,t-1} - y_{t-1}}\right| - \left|\mathrm{z_{ref,t-1} - z_{t-1}}\right| \\ &- \left|\mathrm{Vx_{ref,t-1} - Vx_{t-1}}\right| \\ &- \left|\mathrm{Vy_{ref\,,t-1} - Vy_{t-1}}\right| \\ &- \left|\mathrm{Vz_{ref,t-1} - Vz_{t-1}}\right| - \left|\mathrm{p_{ref,t-1} - p_{t-1}}\right| \\ &- \left|\mathrm{q_{ref,t-1} - q_{t-1}}\right| - \left|\mathrm{r_{ref,t-1} - r_{t-1}}\right| \end{aligned}$$

(13) $$\mathrm{R_{Theta,t} = -\left|X_{Theta,t-1} - X_{Theta,t-2}\right|}$$

where $\mathrm{R_{Vx}}$ is the penalty for velocity deviations, $\mathrm{R_X}$ is the penalty for deviations of the selected state variables (controlled by the LQR) from their reference values, $\mathrm{R_{Theta}}$ includes the difference of the control signal from the two previous time steps to limit unrealistically large and rapid increments in the pitch angle of the pusher propeller, and the subscripts t indicate the time step, where t is the current time step.

For the approximation of the policy and value functions, artificial neural networks are used, the structures of which are shown in Figure 3. The values of the hyperparameters and assumed settings selected during the development of the TD3 agent are included in Table 3.

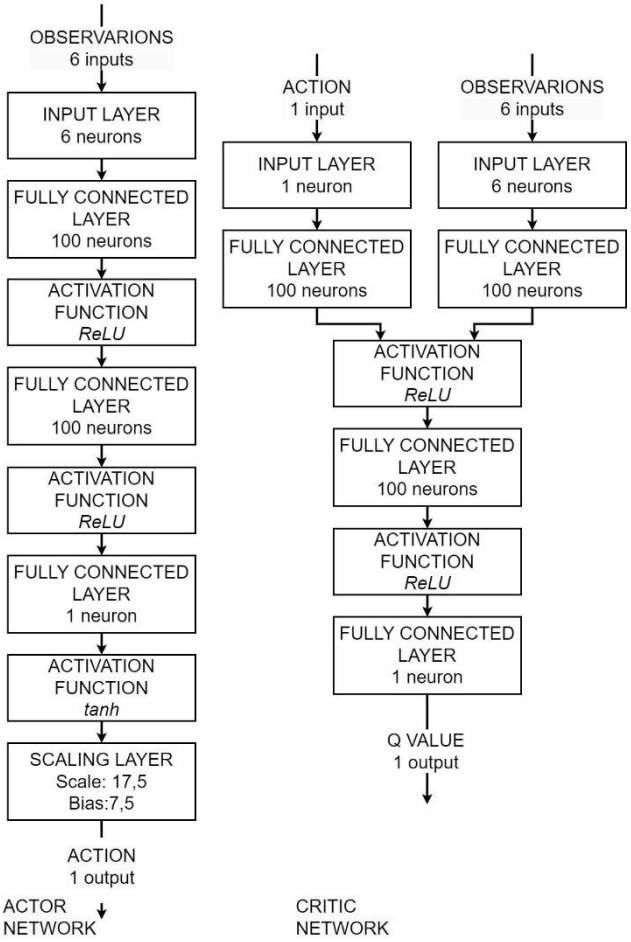

Figure 3: Neural network of the actor (on the left) and the critic (on the right)

Table 3: Hyperparameters of the TD3 algorithm.

Hyperparameter	Value
Sample time	$8{,}3333 \cdot 10^{-4}$ s
Actor learn rate	10^{-3}
Learn rate for both critics	10^{-3}
Experience buffer length	10^{5}
Mini batch size, M	256
Discount factor, γ	$0{,}99$
Target smooth factor, τ	$0{,}005$
Actor and critics optimizer	adam
Variance in target policy smooth model	$3{,}4641$
Variance in exploration model	$3{,}4641$
Policy update frequency	2
Target update frequency	2

The agent's learning process is divided into seven stages, where each stage differs in the number of executed episodes and the values of the set reference velocity. In each episode, the reference velocity is generated using a reset function as a random value from an assumed range of variation dependent on the learning stage. The training phase is designed to start with the simplest possible tasks, so in the first stages of the learning process the reference velocities are close to zero values and therefore correspond to the hovering state – the state mainly for which the primary LQR controller was developed. As the training progress, the values of the reference velocity are increased to ensure that the pusher propeller controller operates effectively over a wider range of flight velocities. Exact information about the successive training stages (number of episodes, ranges of random reference velocities) is presented in Table 4. During training, each simulation lasts for $\mathrm{T_f} = 10$ seconds, which, with an assumed time step of $\mathrm{T_s} = 8{,}333 \cdot 10^{-4}$ seconds (a small value caused by the time step in the helicopter's numerical model), results in $\mathrm{N} = 12000$ time steps performed in each episode. The terminal state occurs when the maximum number of time steps is completed. The results of the computation, in the form of pretrained agents, are saved after every 10 episodes. The performance of the agent is evaluated based on the values of the instantaneous reward and value function as a function of the number of episodes generated during the training, as well as through flight simulations and using predefined quality indicators.

Table 4: Stages of the training process.

Stage	Number of episodes in each stage	Summary number of episodes	Range of $\mathrm{Vx_{ref}}$ $\left[\frac{\mathrm{ft}}{\mathrm{s}}\right]$
1	10	10	0÷1
2	10	20	0÷2
3	10	30	0÷5
4	20	50	0÷10
5	30	80	0÷20
6	30	110	0÷30
7	30	140	0÷40

In the study, the training process was terminated after completing the maximal, presented number of 140 episodes. That was since after the seventh stage, attempts to further training the agent at velocities equal to or greater that those set in seventh stage resulted in the deterioration of the controller performance. The best results are achieved for the last two agents, namely agents after 130 and 140 episodes of training.

4. CONTROL SYSTEM PERFORMANCE

4.1. Test cases

The test cases are provided to enable the comparison of the effectiveness of different modes of the control system, both in terms of the effect of the additional pusher propeller control on reaching the desired velocity (comparison of the control system without and with additional element controller), as well as the performance of both pusher propeller controllers at different flight speeds. The considered modes of the control system are denoted as follows:

- mode a) means operation only of the LQR, and the pusher propeller parameters are set to produce zero thrust,

- mode b) indicates parallel operation of the LQR with proportional controller of the pusher propeller,

- mode c) denotes parallel working of the LQR with the RL agent.

In each test case, the goal of using the control system is to achieve and maintain a constant, reference forward velocity, $\mathrm{Vx_{ref}}$, while keeping the other state variables at preset values: altitude at $\mathrm{z_{ref}} = -20$ ft (the negative value is due to the aircraft coordinate system used in Flightlab), lateral coordinate at $\mathrm{y_{ref}} = 0$ ft, linear velocities at $\mathrm{Vy}_{ref} = 0$ ft/s and $\mathrm{Vz}_{ref} = 0$ ft/s, and angular rates at $\mathrm{p}_{ref} = 0^o$/s, $\mathrm{q_{ref}} = 0^o$/s and $\mathrm{r_{ref}} = 0^o$/s. The simulations start from the hovering state at the altitude with the absolute value of 20 ft. Each simulation is performed for 20 seconds and for the last learned agent, so obtained in the stage seven and after a total of 140 episodes of the learning process. In test cases two values of $\mathrm{Vx_{ref}}$ are assumed: 25 ft/s and 50 ft/s. Information on the test cases is summarized in Table 5.

The additional modification included during the testing phase in mode c) of the control system phase is the application of a low-pass filter to the actions taken by the agent. The performed procedure is made because the time step used in both the plant model and the control system is very small, which does not typically match the time steps used in real systems and which may result in unrealistic changes in the control signal, such as too frequent variations of the control signal or excessively large increments of the control signal. The used in the study filter's cutoff frequency is equal to $\mathrm{f_g} = 1/(2\pi \cdot \mathrm{T_s} \cdot 12)$ Hz, which corresponds to a new sampling period of $\mathrm{T_s^*} = 12 \cdot \mathrm{T_s} \approx 0{,}01$ s. It should be noted that in the presented study the low-pass filter is not used during the training process, but during the testing phase and might be implemented in the destined operation of the controller.

Table 5: Test cases.

Test case No./ Figure	Control system mode	$Vx_{ref}\ \left[\frac{ft}{s}\right]$
1 / Fig. 4	a)	25
2 / Fig. 5	b)	25
3 / Fig. 6	c)	25
4 / Fig. 7	a)	50
5 / Fig. 8	b)	50
6 / Fig. 9	c)	50

4.2. Quality indicators

To quantitatively compare the results for different test cases, two quantity indicators are used: $\mathrm{J_{Vx}}$ and $\mathrm{J_X}$. (Eq. (14) and (15)) The former includes only the deviation of actual velocity from the reference value. The latter takes into account also the deviations of other state variables from their reference values. This provides insight into the impact of the controller's operation on the overall performance of the aircraft, rather than only acquiring the information on achieving a desired velocity.

(14) $$\mathrm{J_{Vx}} = \sum_{\mathrm{t=1}}^{\mathrm{N}} \left|\mathrm{Vx_{ref,t}} - \mathrm{Vx_t}\right|$$

(15)

$$\mathrm{J_X} = \sum_{\mathrm{t=1}}^{\mathrm{N}} \left(\left|\mathrm{y_{ref,t}} - \mathrm{y_t}\right| + \left|\mathrm{z_{ref,t}} - \mathrm{z_t}\right| + \left|\mathrm{Vx_{ref,t}} - \mathrm{Vx_t}\right| + \left|\mathrm{Vy_{ref\,,t}} - \mathrm{Vy_t}\right| + \left|\mathrm{Vz_{ref,t}} - \mathrm{Vz_t}\right| + \left|\mathrm{p_{ref,t}} - \mathrm{p_t}\right| + \left|\mathrm{q_{ref,t}} - \mathrm{q_t}\right| + \left|\mathrm{r_{ref,t}} - \mathrm{r_t}\right|\right)$$

It is desirable to minimize the values of $\mathrm{J_{Vx}}$ and $\mathrm{J_X}$, which means approaching the current state variables to the reference values.

4.3. Simulation results

The simulation results in the form of plots of the state and control variables as a function of time for the selected test cases are presented in Figures 4 – 9, where the black curves correspond to the helicopter actual state and control variables, and the red lines in the Vx plots represent the reference velocity, $\mathrm{Vx_{ref}}$. Additionally, the calculated values of the quality indicators $\mathrm{J_{Vx}}$ and $\mathrm{J_X}$ are listed in Table 6.

Table 6: Values of the quality indicators.

Test case No.	Test case description	J_{Vx}	J_X
1	25 ft/s, mode a)	144,70	198,70
2	25 ft/s, mode b)	38,62	109,40
3	25 ft/s, mode c)	27,26	120,10
4	50 ft/s, mode a)	419,30	569,70
5	50 ft/s, mode b)	113,60	220,60
6	50 ft/s, mode c)	56,68	203,70

From the provided performances, it is apparent that for both proposed reference velocities, the lack of an additional pusher propeller (test cases 1 and 4) results in not reaching the exact preset forward velocity. The thrust force generated solely by the LQR-controlled main rotor is insufficient to achieve the desired flight velocity. The addition of an operating pusher propeller improves performance in terms of approaching the desired $\mathrm{Vx}_{\mathrm{ref}}$, with better minimization of the velocity deviation occurring in mode c) of the control system, which includes the RL agent.

The quality indicators calculated for the test cases with the lower velocity of 25 ft/s show similar effectiveness for modes b) and c). The linear velocity is slightly better controlled with the RL algorithm (test case 3), whereas in a more holistic view and taking into account the other state variables, the proportional controller seems to be more effective (test case 2). For the higher velocity of 50 ft/s, both quality indicators reveal the superiority of the nonlinear controller (test case 6), which seems to select the values of the pusher propeller parameter more appropriately according to the current flight state over a wider range of forward velocities than the linear controller. However, it is important to remember that the quality of the RL agent's performance is strongly influenced by the selected hyperparameter values and the way the learning process is carried out, including the reference values selection, the training phase duration, etc.

In the present study, no optimization of the parameters used in the RL algorithm is carried out, however, the execution of such optimization could further improve the performance of the developed nonlinear controller.

In each of the presented modes of the control system operation, such state variables as linear velocities $\mathrm{V}x$ and $\mathrm{V}y$, lateral and vertical position coordinates, y and z, are not kept exactly at their reference values, but are set to constant values slightly deviating from the reference values. However, it should be noted that in modes b) and c), where additional control of the pusher propeller is included, the simulation results do not show any deterioration in the deviations of the above state variables compared to mode a) of the control system. Thus, extending the control approach by using the LQR and pusher propeller control in parallel not only does not degrade the performance of the main control system, but also allows an increase in the range of flight velocities achieved.

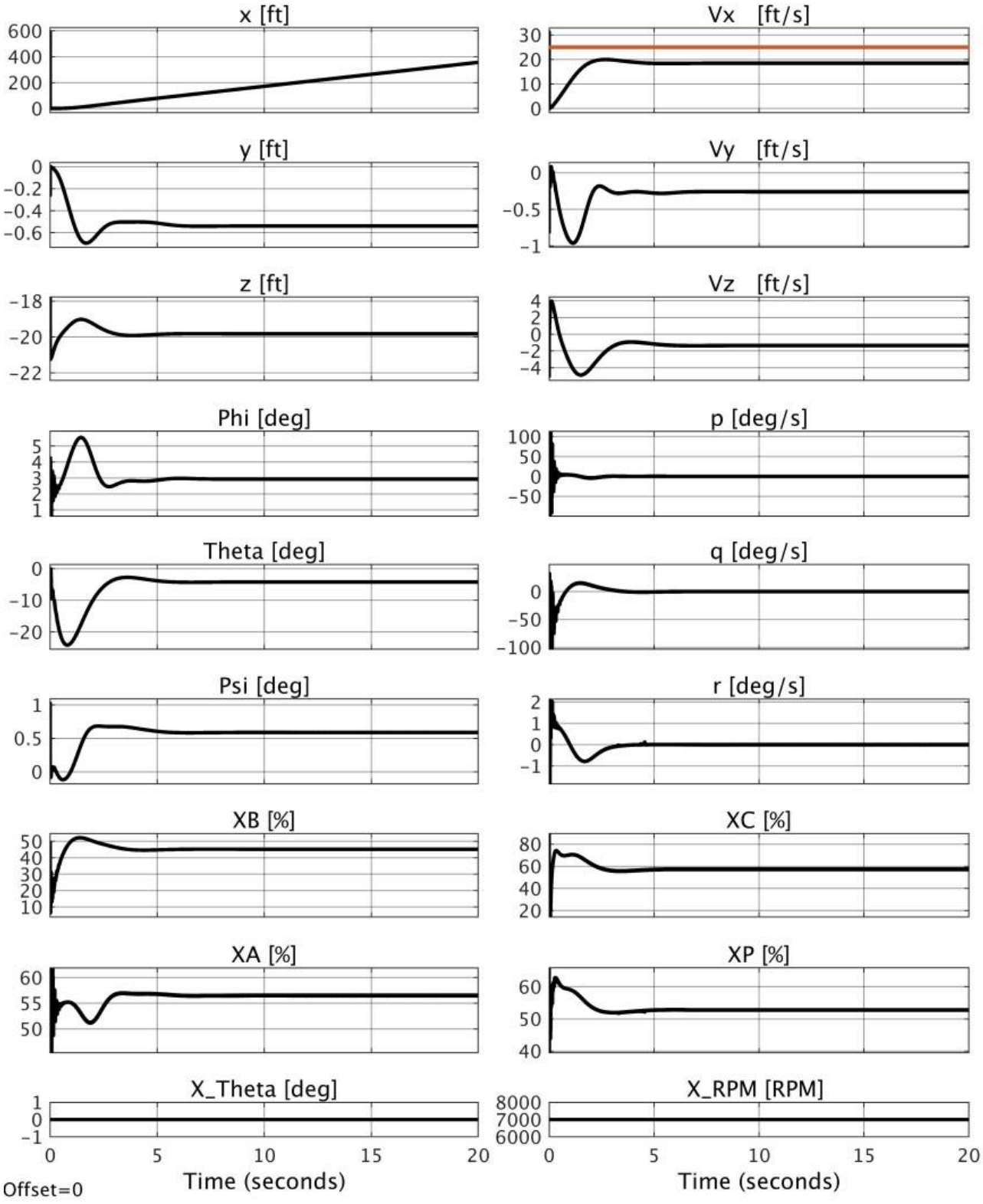

Figure 4: State and control variables for test case 1: mode a) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 25$ ft/s

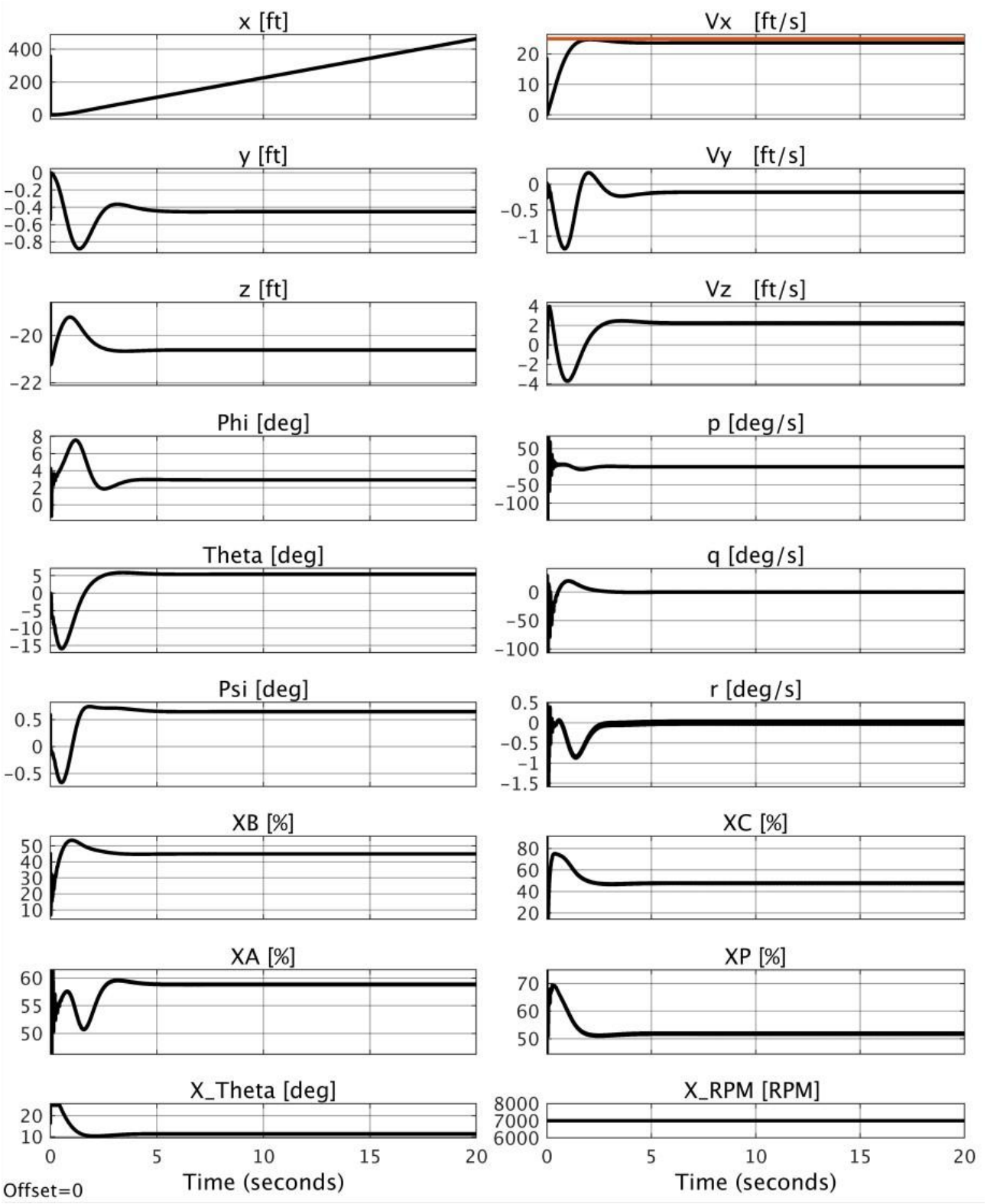

Figure 5: State and control variables for test case 2: mode b) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 25$ ft/s

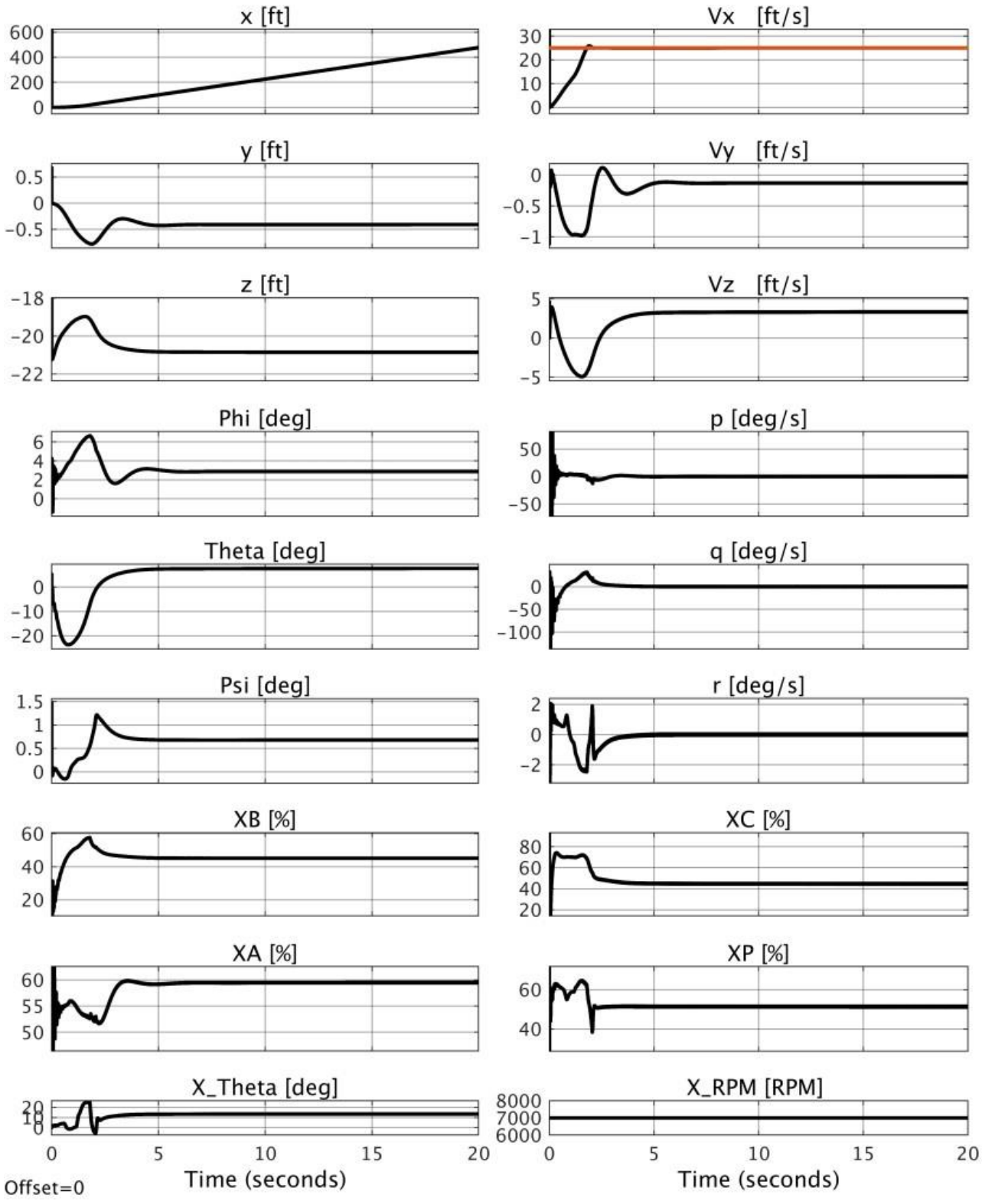

Figure 6: State and control variables for test case 3: mode c) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 25$ ft/s

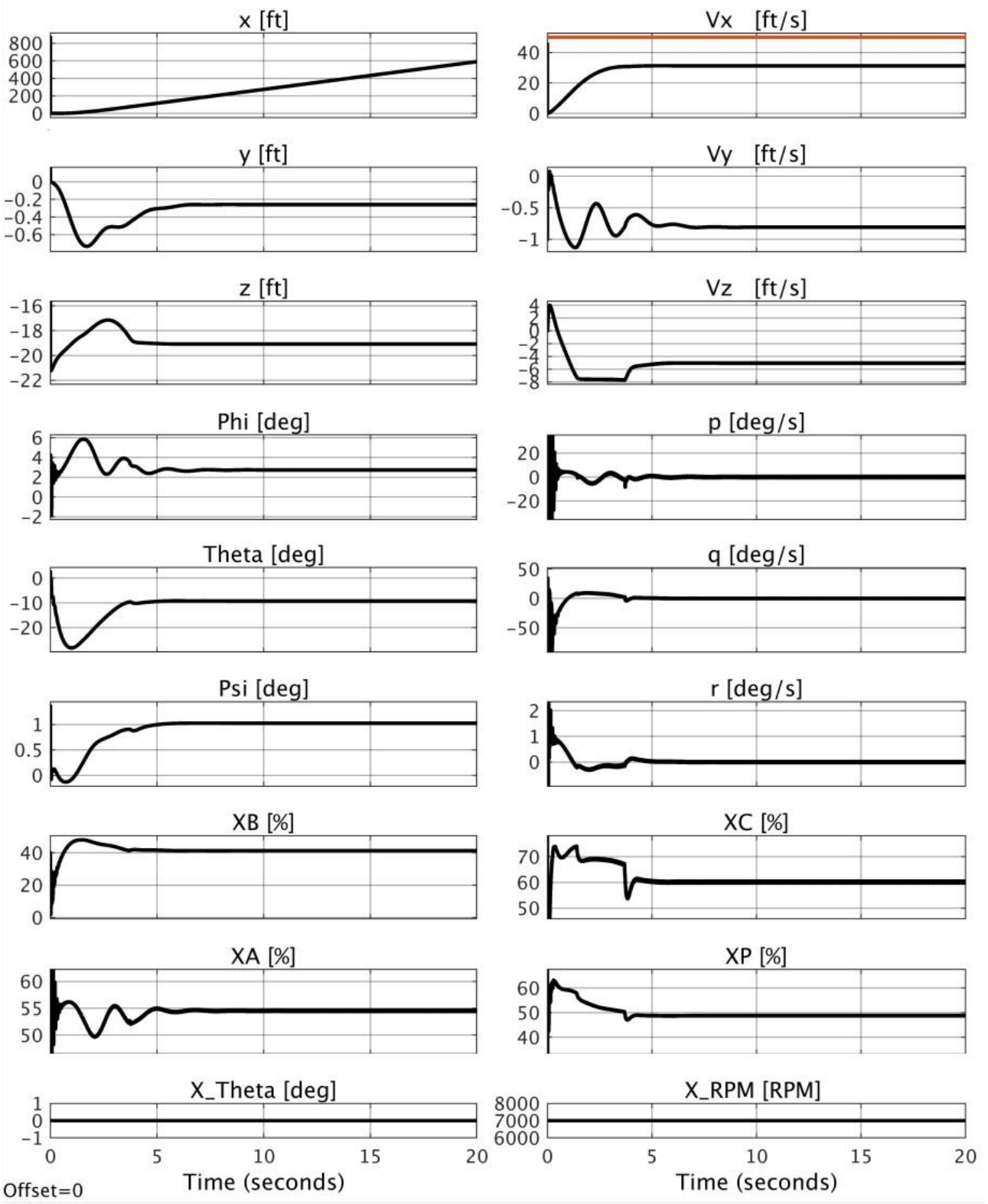

Figure 7: State and control variables for test case 4: mode a) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 50$ ft/s

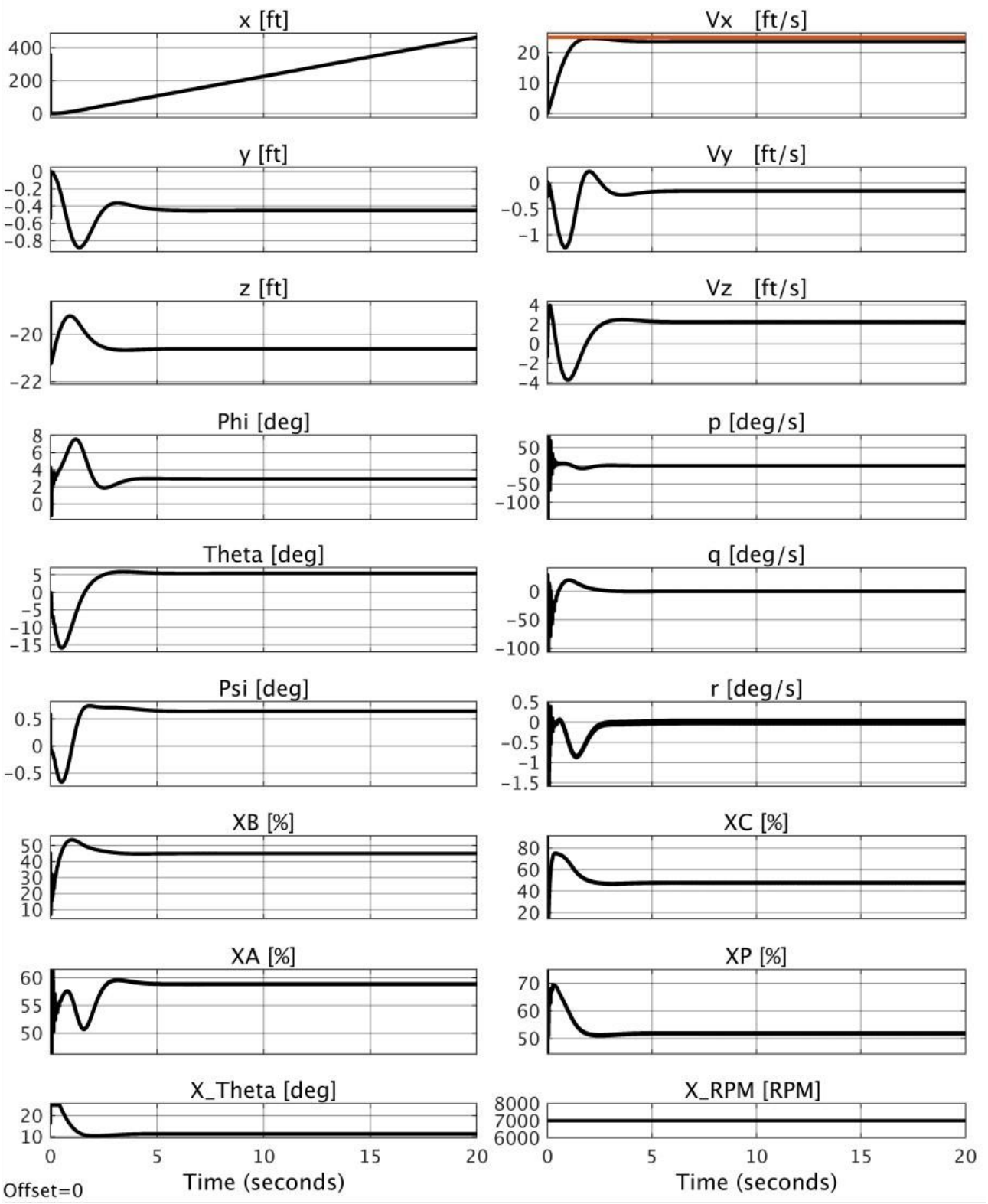

Figure 8: State and control variables for test case 5: mode b) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 50$ ft/s

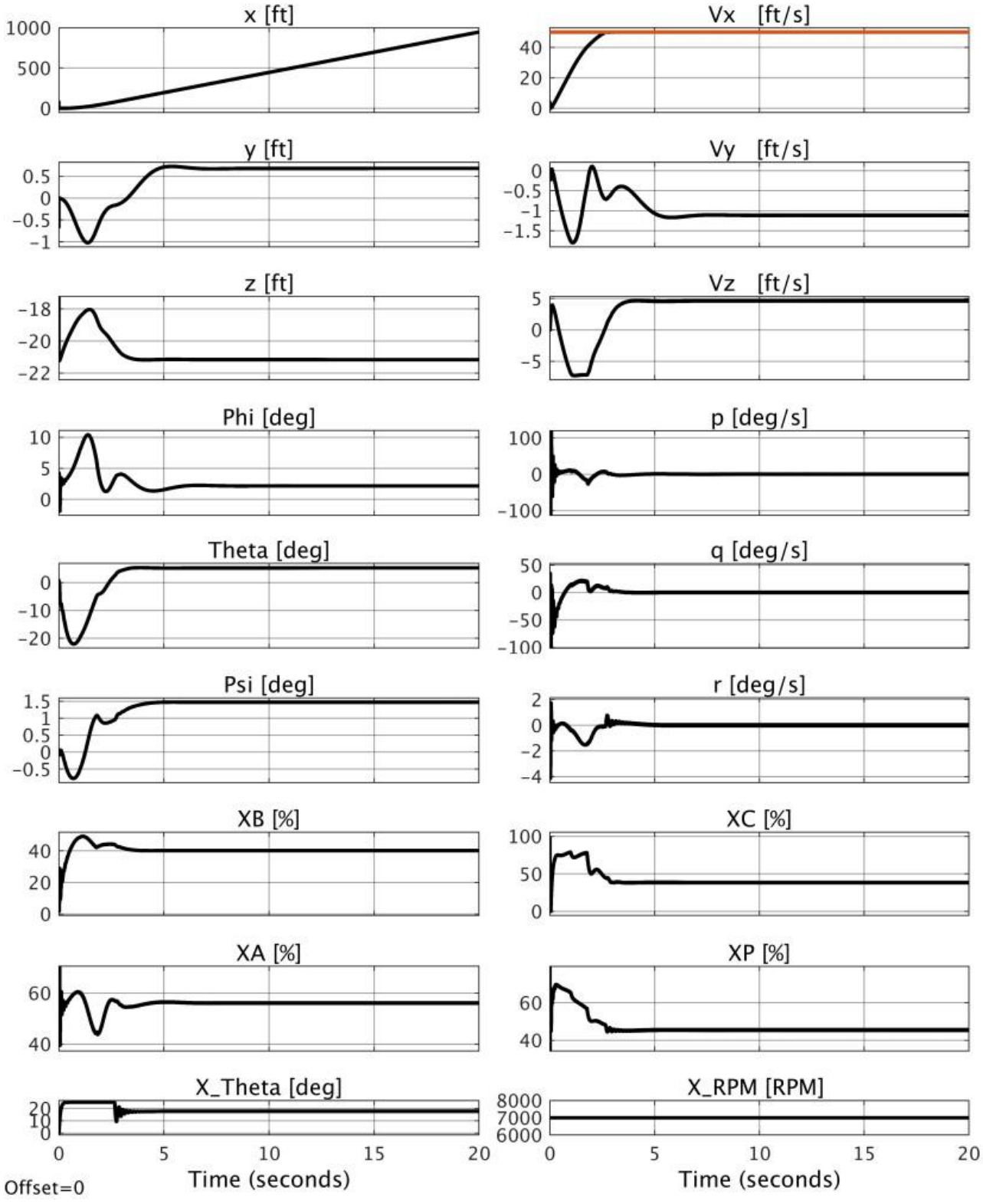

Figure 9: State and control variables for test case 6: mode c) of the control system, $\mathrm{Vx}_{\mathrm{ref}} = 50$ ft/s

5. CONCLUSIONS

The paper deals with the development of the control system for the compound helicopter. The primary controller, responsible for the classical part of the aircraft, uses the LQR algorithm. For the additional pusher propeller, two controllers are designed: the linear, proportional controller and the target, nonlinear controller, based on the TD3 reinforcement learning algorithm. The main objectives of the presented studies and test case results are to demonstrate the effect of the added component control on the overall control system, to show how the pusher propeller aids the main part of the helicopter in the classical configuration in realizing the given task – obtaining the preset forward velocity, and to compare two proposed pusher propeller controllers, highlighting the possibility of using RL algorithm in control tasks.

The main conclusions of the paper are as follows:

1. The presented control system provides the stabilization of the compound helicopter and control that allows the aircraft to fly up to $\mathrm{Vx}_{\mathrm{ref}} = 50$ ft/s.
2. The use of the LQR controller alone, which is mainly designed for the hover condition, and without the thrust force from the additional pusher propeller, proves to be insufficient to achieve the specified flight velocities.
3. The addition of the pusher propeller and appropriate control of its parameters not only does not adversely affect the performance of the aircraft (in comparison with the results obtained under exclusive LQR control), but also allows to increase the range of flight velocities achieved.
4. At lower velocities, both proposed additional component controllers give similar results, however, as $\mathrm{Vx}_{\mathrm{ref}}$ increases and thus moves away from the hover state, the nonlinear controller starts to be more preferable.
5. The presented results and the comparison of two pusher propeller controllers show the importance of the correct selection of a reinforcement learning method for the considered application, the optimization of the used hyperparameters and the appropriate implementation of the agent learning process in the development of effective RL-based controllers that have the desired characteristics and give better results (in a wide range of flight envelopes) than the commonly used linear controllers.

Author contact:

Sara Waśniewska,
sara.wasniewska.dokt@pw.edu.pl

Sebastian Topczewski,
sebastian.topczewski@pw.edu.pl

6. REFERENCES

1. Tijani, I. B., Akmeliawati, R., Legowo, A., Budiyono, A. and Muthalif, A. A. G., “H∞ robust controller for autonomous helicopter hovering control,” *Aircraft Engineering and Aerospace Technology*, Vol. 83, (6), Oct. 2011, pp. 363-374, DOI: 10.1108/00022661111173243.
2. Bayisa, A. T., “Controlling Quadcopter Altitude using PID-Control System,” International Journal of Engineering Research & Technology (IJERT), Vol. 8, (12), Dec. 2019, DOI: 10.17577/IJERTV8IS120118.
3. Budiyono, A. and Wibowo, S. S., “Optimal Tracking Controller Design for a Small Scale Helicopter,” Journal of Bionic Engineering, Vol. 4, (4), pp. 271-280, Apr. 2008, DOI: 10.1016/S1672-6529(07)60041-9.
4. Huang, H., Hoffmann, G. M., Waslander, S. L. and Tomlin, C. J., “Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering,” ICRA’09. IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12-17, 2009.
5. Guerreiro, B., Silvestre, C. and Cunha, R., “Terrain Avoidance model Predictive Control for Autonomous Rotorcraft,” Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008.
6. Kadmiry, B., Palm, R. and Driankov, D., “Autonomous Helicopter Control Using Gradient Descent Optimization Method,” 4th Asian Conference on Robotics and its Applications (ACRA 2001), Singapore, June 6-8, 2001.
7. Zheng, F., Xiong, B., Zhang, J., Zhen, Z. and Wang, F., “Improved neural network adaptive control for compound helicopter with uncertain cross-coupling in multimodal maneuver,” Nonlinear Dynamics, Vol. 108, (3), Apr. 2022, DOI: 10.1007/s11071-022-07382-x.
8. Kadmiry, B. and Driankov, D., “Fuzzy Control of an Autonomous Helicopter,” IFSA World

Congress and 20th NAFIPS International Conference, Vancouver, Canada, July 25-28, 2001.

9. Ammar, H. B., Voos, H. and Ertel, W., “Controller Design for Quadrotor UAVs using Reinforcement Learning,” Proceedings of the IEEE International Conference on Control Application, CCA 2010, Yokohama, Japan, September 8-10, 2010.

10. Waśniewska, S. and Topczewski, S., “Non-linear Approach for Unmanned, Compound Helicopter Control,” 49th European Rotorcraft Forum 2023, Bückeburg, Germany, September 5-7, 2023.

11. Topczewski, S., Bibik, P., Waśniewska, S. and Cioć, T., “Automatic Flight Control System for the Small-scale Compound Helicopter,” 48th European Rotorcraft Forum, Winterthur, Switzerland, September 6-8, 2022.

12. Sutton, R. S. and Barto, A. G., *“*Reinforcement Learning: An Introduction,” The MIT Press, Cambridge, 2015, Chapters 1, 3.

13. Fujimoto, S., van Hoof, H., Meger, D., “Addressing Function Approximation Error in Actor-Critic Methods,” 35th International Conference on Machine Learning, Stockholm, Sweden, July 10-15, 2018.