Russian Quantum Center

as a manuscript

Sorokin Dmitrii Igorevich

Development of reinforcement learning methods to control robotic and virtual agents

PhD Dissertation Summary

for the purpose of obtaining academic degree
Doctor of Philosophy in Computer Science

Moscow — 2023

The PhD dissertation was prepared at Russian Quantum Center, RQC.

Academic Supervisor: Alexander Isayevich Lvovsky, PhD in Physics, Professor of the Department of Physics, University of Oxford and Principal Investigator at the Russian Quantum Center.

DISSERTATION TOPIC

Modern deep reinforcement learning methods (RL) can solve optimal control and planning tasks without prior information about the problem. Learning is performed through trial and error as the agent interacts with the environment and learns to optimize its actions to maximize the expected reward. This formulation closely resembles how humans learn, and can be considered as one of the approaches to creating artificial general intelligence [1]. Deep reinforcement learning was applied for the first time to control a virtual agent in the Atari 2600 game-based environment in 2013[2]. It has been shown that RL can control the agent based on visual information at the human level. Since then, the topic of reinforcement learning has received more and more attention. Reinforcement learning is effective in many complicated tasks such as playing chess [3], Go [4], and StarCraft II [5]. These impressive results have become possible due to the development of both RL methods and computer technology since successful training of deep RL agents requires a huge number of interactions with the environment. The research carried out in this work is mainly focused on the development of reinforcement learning methods applied to control real robotic devices. The creation of algorithms capable of making decisions in the real world, self-learning, and not requiring large amounts of human-labeled data, can lead to explosive growth in collaborative robotics, self-driving cars, and the field of virtual assistants.

At present, the scope of application of RL methods is limited. The main limiting factor is the large amount of data needed to train the agent. To leverage this problem, the agent is often trained in a virtual environment — a simulator and then transferred to a physical installation. This setting leads to a second problem — a difference between observations and the dynamics of the environment in the simulation and the reality, which results in a significant decrease in the performance of the agent in the real world. Another problem arises when an agent is trained directly on a physical robot — the ability of the agent to perform actions that could break the robot should be limited during the learning process without a large restriction on the agent's ability to explore the environment. Moreover, many practical tasks can be viewed as a hierarchy of subtasks. While one part of the subtasks can be effectively solved by classical methods, the other part requires more complex learning methods. This leads to the need to combine neural networks and classical approaches within a single hierarchical agent.

Actuality. One of the most promising applications for reinforcement learning methods is robotics. Modern robots are already replacing humans in strictly controlled environments such as manufacturing lines because it is possible to develop a program that takes into account all possible environment states. In contrast, in the everyday life, it is important to act in conditions of uncertainty. In such cases, the optimal behavior is difficult to program, but it

can be learned through interaction with the environment. However, in order to apply reinforcement learning methods to control physical robots, one needs to solve a number of challenges such as reducing the number of examples needed for training, increasing the efficiency of transferring models trained in simulation to the real world, and developing algorithms capable of solving problems in the absence or sparse reward function.

In the present dissertation, we developed methods with would mitigate the limitations of reinforcement learning algorithms in real-world tasks. We evaluate the performance of our algorithms on a set of important practical applications and propose a novel application for reinforcement learning methods — alignment of an optical interferometer. For the first time, we developed a reinforcement learning method for the alignment of an optical Mach-Zehnder interferometer [6; 7]. To align an interferometer, the agent must be capable of sampling actions with different amplitudes — large actions at the beginning and small, precise actions at the end of the alignment process. The size of the actions at the beginning of an episode can be more than 100 times bigger than at the end. This problem is not frequent in usual RL benchmarks but is important for practical applications. Also, the agent trained in the simulation must be robust to optical noises in the input images when deployed at the experimental setup. Interferometers are a common part of most setups in experimental optics. Alignment of the experimental setup is one of the most labor-intensive phases in experimental optics. Fine-tuning hundreds of optical elements such as lenses, mirrors, and attenuators requires a lot of experimental experience and takes many hours even for an experienced specialist. Automation of the alignment process will greatly increase the performance of optical experiments. The developed method uses images from cameras and adapts to the parameters of a particular setup. The proposed method is significantly superior to a human in terms of quality and speed of interferometer alignment.

A similar task of continuous control arises in quadruped robots. We considered the task of controlling the movement of a quadruped Unitree A1 robot[8]. For successful learning of RL algorithms, it is important to specify the reward function. With a carelessly given reward function, an agent can learn a suboptimal policy. We propose a reward function with penalties for the non-optimality of the agent's policy increasing according to the schedule. The proposed reward function was applied to train the RL agent to control the movement of the Unitree A1 [9] robot with given parameters such as angular and linear velocities. The proposed reward function encourages the agent to learn safe and smooth movement at a given speed. In this case of a robot with many degrees of freedom, the use of reinforcement learning has significant advantages over classical algorithms, since the agent can independently learn the optimal policy based only on the scalar reward function. The optimal policy should filter out the noise in the observations well, not require an excessive amount of data for training, and not perform actions that could damage the

robot. The results of testing the trained agent showed that it can perform well in the simulation.

The next challenge in the application of reinforcement learning methods in the real life is to combine them with algorithmic approaches to solve complex problems. As part of this work, we propose a hierarchical algorithm that combines reinforcement learning, classical algorithms on graphs, and expert knowledge. The policy is built from basic skills designed to solve specific problems, and the choice of skill is based on the current state. The developed algorithm was applied to control a virtual agent in the NetHack [10] environment. This environment is one of the most difficult test environments for reinforcement learning algorithms. The average episode length in NetHack is 100,000 steps, which is 50 times longer than in StarCraft II. NetHack is also a procedurally generated environment, which means that an agent can rarely be in the same state more than once. The large space of actions and different states of the environment means that most of the methods used in reinforcement learning to explore the environment do not work in this setting. The developed method made it possible to effectively apply reinforcement learning to this problem and won first place in the competition held by Google DeepMind and Facebook AI Research as part of the NeurIPS Competition track 2021 conference. This approach can be used to design systems that combine machine learning and classical algorithms.

The aim of this work is the development of machine learning methods for the control of robots in real-world tasks.

We formulate the following **goals of this research**:

1. Development of a simulator of Mach-Zehnder interferometer.
2. Formulation of an optical interferometer alignment in terms of Markov decision process. Selection of the reward function, definition of action and observation spaces, and choice of hyperparameters.
3. Development of reinforcement learning algorithm capable of operating with actions of different magnitude and robust to optical noises.
4. Development of software complex to align an optical interferometer.
5. Development of a curriculum-based method with penalties increasing according to a schedule and training an RL algorithm to control the speed of a quadruped robot.
6. Development of a hierarchical method that combines algorithmic and neural approaches and its application to the NetHack environment.

Scientific novelty:

1. We proposed a reinforcement learning method that can operate actions of different magnitudes and is robust to optical noises. This method was for the first time applied to align the optical interferometer.
2. For the first time, we created a hardware-software complex for tuning an optical interferometer based on images from a camera based on reinforcement learning.

3. We developed an original method for learning a policy to control the movement of a quadruped robot with the desired speed.
4. We performed an original study on the applicability of a hierarchical algorithm combining neural and algorithmic policies for the NetHack game.

Theoretical and practical significance of the work is:

1. The automated approach to the alignment of the optical interferometer proposed in the dissertation will significantly speed up the conduct of physical experiments and reduce the need for manual labor.
2. The developed algorithms for controlling virtual agents can be applied in robotics, self-driving cars, and virtual assistants.

Research methods. In the work we applied methods of machine learning, computer vision, deep reinforcement learning, software development, linear algebra, general physics, and optics.

KEY RESULTS

The main defense points:

1. A reinforcement learning method capable of sampling actions of different magnitudes and resistant to optical noise. The developed method makes it possible to adjust the optical interferometer without human intervention, based solely on images of the interference pattern. The proposed method does not use a priori information and is able to adapt to a specific setup.
2. Software-hardware complex for automatic alignment of an optical interferometer. The speed and accuracy of alignment using the developed method significantly exceed manual tuning.
3. A reinforcement learning method for policy learning for controlling the movement of a quadruped robot with the target linear and angular velocity.
4. Hierarchical algorithm combining algorithmic and neural approaches. The algorithm was tested in the NetHack environment.

Author's contribution to the study. The author personally developed a simulator of the optical interferometer, a method for training an RL agent with discrete action space to align the interferometer, and a software-hardware complex for testing the trained agent on the experimental setup. The author has been actively involved in the development of a machine-learning method for the alignment of the interferometer using a continuous action space. The author personally proposed and implemented a reward function that allows learning a policy to control the movement of a walking robot at a given speed. The author personally proposed and implemented a hierarchical agent that combines reinforcement learning and algorithmic approach for playing NetHack.

PUBLICATIONS AND APPROBATION OF RESEARCH

First-tier publications

1. Interferobot: aligning an optical interferometer by a reinforcement learning agent [Текст] / D. Sorokin [et al.] // Advances in Neural Information Processing Systems. Vol. 33 / ed. by H. Larochelle [et al.]. — Curran Associates, Inc., 2020. — P. 13238—13248. — URL: https://proceedings.neurips.cc/paper/2020/file/99ba5c4097c6b8fef5ed774a1a6714b8-Paper.pdf. (**CORE A***)

Second-tier publications

1. *Sorokin, D. I.* Learning Various Locomotion Skills from Scratch with Deep Reinforcement Learning [Текст] / D. I. Sorokin, D. L. Babaev // Advances in Neural Computation, Machine Learning, and Cognitive Research VI / ed. by B. Kryzhanovsky [et al.]. — Cham : Springer International Publishing, 2023. — P. 322—329. (**Scopus Q4**)

Other publications

1. Aligning an optical interferometer with beam divergence control and continuous action space [Текст] / S. Makarenko [et al.] // Proceedings of the 5th Conference on Robot Learning. Vol. 164 / ed. by A. Faust, D. Hsu, G. Neumann. — PMLR, 2022. — P. 918—927. — (Proceedings of Machine Learning Research). — URL: https://proceedings.mlr.press/v164/makarenko22a.html.
2. Insights From the NeurIPS 2021 NetHack Challenge [Текст] / E. Hambro [et al.] // Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track. Vol. 176 / ed. by D. Kiela, M. Ciccone, B. Caputo. — PMLR, 2022. — P. 41—52. — (Proceedings of Machine Learning Research). — URL: https://proceedings.mlr.press/v176/hambro22a.html.
3. *Certificate of state registration of the program for PC.* Software package for automatic tuning of an optical interferometer based on machine learning (Interferobot) / D. I. Sorokin ; RQC. — № 2021616685 ; заявл. 26.04.2021 ; опубл. 15.04.2021, 2021616685 (Russian Federation).

Reports at conferences and seminars

1. 34-th annual international conference Neural Information Processing Systems (NeurIPS 2020) (**spotlight talk**).
2. 29-th annual international conference on laser physics LPHYS 2021.
3. 5-th international Conference on Robot Learning (CoRL) 2021.

4. 35-th international conference Neural Information Processing Systems (NeurIPS 2021, Competition track).
5. International conference on quantum technology ICQT 2021.
6. International conference on artificial neural networks Neuroinformatics 2022.

Participation in scientific projects

1. Grant UMNIK «Development of an automatic alignment system for an optical interferometer based on machine learning» № 120ГУЦ-ЭС8-D3/56352 от 21.12.2019

The reliability of obtained results is ensured by complex testing of the proposed method for automated tuning optical interferometer, carried out at RQC. According to the results of the competition held by Google DeepMind and Facebook AI Research, the developed method of managing an agent in the NetHack environment has surpassed the rest approaches using neural networks.

CONTENTS

The introduction substantiates the actuality of the research conducted in the present dissertation. We present an overview of the literature which shows the great potential of deep reinforcement learning methods in decision-making tasks. We formulate goals and tasks for the present work. We present the scientific novelty and practical significance of the results of the work. The following chapters first describe the general principles of reinforcement learning methods. Then we discuss modern reinforcement learning methods which achieve state-of-the-art results in various benchmarks. In the second chapter, we describe the task of alignment of an optical interferometer in terms of reinforcement learning. After, we present our reinforcement learning methods to align the interferometer using discrete and continuous action spaces, and the results of the evaluation of trained agents on a physical interferometer setup. The third chapter introduces the task of control of a quadruped robot. We present the proposed method for learning a policy for locomotion with target speed. The forth chapter formulates the need for complex test problems to develop and test the generalization ability of reinforcement learning methods. We formulate challenges, which makes the game of NetHack one of the most difficult test environments for reinforcement learning methods. Then we present the developed algorithm and evaluate its performance. In conclusion, the main results obtained during the preparation of the dissertation research are summarized, indicating their novelty and practical significance.

The **first chapter** is devoted to an overview of reinforcement learning methods (RL) and their applications in robotics. It formulates the RL problem — maximization of the expected total discounted reward obtained in interaction with the environment. Schematically, this interaction is shown in fig. 1. In fig. 1 an agent interacts with the environment through actions and receives from the environment a tuple of the next state, reward, and termination flag.

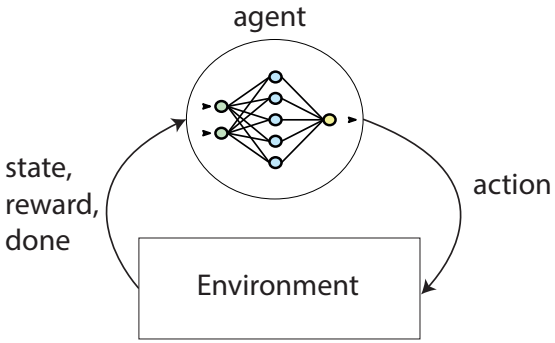

Figure 1 — Interaction of RL agent and environment.

The expectation of the total discounted reward is computed along trajectories sampled from the current policy of the agent:

$$E_{\tau\sim\pi(\theta)}[G(\tau)] = E_{\tau\sim\pi(\theta)}[R_0 + \gamma R_1 + \gamma^2 R_2 + ...] = E_{\tau\sim\pi(\theta)}[\sum_{t=0}^{T-1} \gamma^t R_t]$$

The interaction between the agent and the environment is considered under the assumption of a Markov decision process. The Markov property says that the next state of the environment and the reward received by the agent depend only on the previous state of the environment and the action taken by the agent in this state. Given this assumption, the optimal policy of an agent depends only on the current state of the environment, which allows using the Bellman optimality equation. The Bellman optimality equation expresses the relationship between the optimal policy in two successive states of the environment s and s'

$$V^*(s) = \max_{a\in\mathcal{A}} E(r_{t+1}(s, a, s') + \gamma V^*(s_{t+1})),$$

where $V^*(s)$ (V-function) - expectation of total discounted reward in state s under the condition of the optimal policy, γ is the discounting factor. Equivalently, the Bellman equation can be expressed as follows:

$$Q^*(s, a) = E(r_{t+1}(s, s', a) + \max_{a\in\mathcal{A}} \gamma Q^*(s_{t+1}, a)),$$

where $Q^*(s, a)$ (Q-function) - expectation of total discounted reward in state s under the condition of action a and following the optimal policy in the next state s'.

Next, we will consider the main reinforcement learning algorithms. Conventionally, RL algorithms can be divided into two large classes off-policy methods and on-policy methods. Off-policy methods are based on the maximization of the Q-function using the time difference method. Such methods can use data obtained by an arbitrary policy. Due to this, off-policy methods usually require fewer data, but may be unstable during training. On the other hand, on-policy methods which can use only data obtained by the current policy are more stable but require significantly more training data.

Then we consider tasks with **sparse reward**. In such environments, the probability that an agent following random policy will find a non-zero reward and thus receive a positive learning signal is negligible. In this case, approaches called intrinsic motivation are used. The general idea of these methods is to reward the agent for discovering new states, which would motivate it to efficiently explore the environment.

The **meta-learning** algorithms are discussed next. Generally, reinforcement learning methods are designed to solve a concrete task and cannot be easily generalized to similar problems. In order to train an agent to solve a similar task, its training should be started from the beginning. Meta-learning algorithms allow the training of an agent that can adapt to a distribution of tasks.

Next, we discuss **hierarchical reinforcement learning**. The main idea of hierarchical reinforcement learning is to construct a policy as a hierarchy of skills, in which the lower-level policies solve sub-tasks that are used by the upper-level policy to solve more complex problems. Such a decomposition can be useful in a wide range of problems. For example, in the task of controlling a walking robot, the lower-level policy can be trained to walk in a target direction, while the upper-level policy uses this skill to move the robot to a given point.

At the end of the chapter, we discuss the application of reinforcement learning methods in the problems of controlling robots. Recently, RL methods have achieved outstanding results, such as solving a Rubik's Cube with a robotic arm [11] and even controlling a fusion reactor [12]. Compared to methods based on classical control algorithms such as inverse kinematics, machine learning algorithms are able to adapt to the parameters of the robot and, consequently, work in conditions where these parameters are not precisely known.

In the **second chapter** we consider an environment in which the optimal agent must be able to handle actions of various scales, be noise-resistant to be transferred to a physical setup, and work with high-dimensional states. Based on the method developed for an environment with such properties, we created a software and hardware complex for aligning an optical interferometer.

At the beginning of the chapter, we discuss the physical principles of an optical interferometer. The optical interferometer utilizes the principle of light interference - when two coherent light waves are superimposed, light oscillations of different amplitudes occur at different points in space. Depending on whether the waves are superimposed in phase or in anti-phase, the interference leads to an increase or decrease in the total amplitude of the oscillations. Within the framework of this work, we consider the interference of two light beams emitted by one laser light source. In this case, the interference pattern looks like a series of bright and dark fringes visible to the naked eye. Schematically, the interference pattern obtained by superimposing two light beams with directions given by the wave vectors $k1$ and $k2$ is shown in fig. 2.

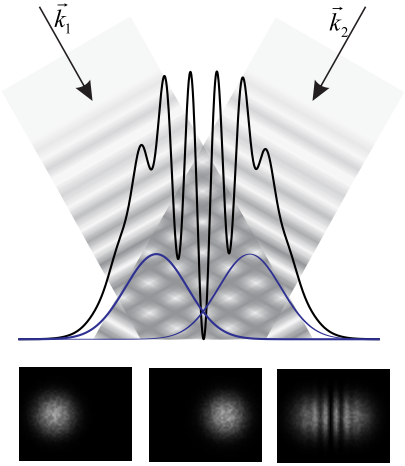

Figure 2 — One-dimensional slice of the interference pattern obtained with an incomplete overlap of two coherent laser beams with wave vectors denoted by k1 and k2. The wavefront is shown using a gray gradient. Blue lines show the intensities of each beam (not to scale). The black line shows the intensity of the interference pattern. The 2D interference patterns corresponding to the 1D case are shown below.

The interferometer uses the principle of interference to accurately measure the relative phase difference between two coherent laser beams. The interferometer is one of the essential instruments used in optical experiments. For instance, the Fabry-Pierrot interferometer is used in spectroscopy [13]. Modern gravitational wave detectors LIGO and VIRGO [14; 15] use a Michelson interferometer. Sagnac interferometer is used in navigation systems [16]. The Mach-Zehnder interferometer is the main tool for conducting modern experiments in quantum optics [17; 18].

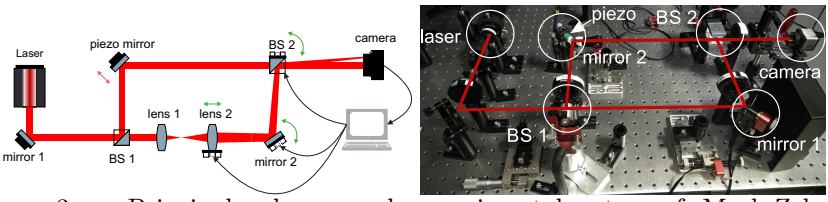

Figure 3 — Principal scheme and experimental setup of Mach-Zehnder interferometer.

Next, we consider the Mach-Zenhder interferometer shown in fig. 3. In the scheme shown in fig. 3 the laser beam is divided into two by the first beam splitter (BS 1). Then the two beams propagate along the two arms of the interferometer. Finally, they coincide at the second beam splitter (BS 2). The resulting interference pattern is observed by a digital camera. To control the mirror (mirror 1), beam splitter (BS 2), and lens (lens 2), we use optomechanical shifts. The mirror and the beam splitter shifts can rotate along two axes while the lens shift moves linearly. The control is performed by a PC using an actuator. The goal of interferometer alignment is to precisely match the two beams after passing through the interferometer arms so that their center positions, k-vector directions, and wavefront curvatures match. The interferometer is aligned based on the interference pattern observed on the camera. An important part of the observations is the temporal dynamics of the interference fringes observed due to the piezo mirror (mirror 2), which moves periodically with an amplitude of the order of a wavelength. In our experiment, the time of the forward pass of the piezo mirror is longer than the time of the backward pass, which makes it possible to guess not only the absolute value of the angle between the k-vectors but also its sign.

Despite the apparent simplicity, the process of alignment of the interferometer is laborious. Firstly, this is due to the fact that each movement of the mirror (mirror 1) and the beam splitter (BS 2) leads to a simultaneous change in both the position of the lower beam on the camera and its direction. Thus, when the beams are moved together, their parallelism is violated and vice versa. Secondly, the interference pattern observed by the camera may contain optical noise, aberrations, and dust particles as shown in 4(a). In the third, optomechanical shifts that control the positions of optical elements have a significant hysteresis and limited sensitivity.

After, we present a computer model of the Mach-Zehnder interferometer constructed on the basis of interference principles, which can simulate images obtained on the camera of an optical interferometer with an arbitrary arrangement of optical elements - mirrors and lenses. In our model, laser beams in cross-sections are described by the Gaussian function. The vector of the electric field strength at the point (x,y,z) for the beam propagating along the upper arm of the interferometer is given by the expression:

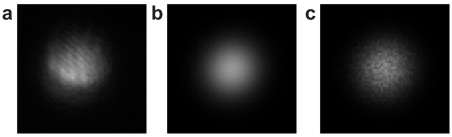

Figure 4 — Laser beams: (a) experimental. (b) simulation with Gaussian amplitude profile. (c) simulation with noise. In fig. (a), fringes are visible due to the resonator effect.

$$E_u = \mathrm{Re}\left[\exp\left(-\frac{x^2+y^2}{r_u^2(z)}\right)\exp\left(-i\left(k_z z + k\frac{x^2+y^2}{2\rho_u^2(z)} + \phi_{\mathrm{piezo}}(t)\right)\right)\right] \quad (1)$$

The vector of the electric field strength for the beam propagating along the lower arm of the interferometer is given by:

$$E_l = \mathrm{Re}\Bigg[\exp\left(-\frac{(x-x_0)^2+(y-y_0)^2}{r_l^2(z)}\right)\cdot \exp\left(-i\left(k_x x + k_y y + k_z z + k\frac{x^2+y^2}{2\rho_l^2(z)}z\right)\right)\Bigg], \quad (2)$$

where (x_0, y_0) is the position of the center of the lower beam [the center of the upper beam is assumed to be $(x,y) = (0,0)$], the beam propagates along the z axis, $r(z)$ — beam radius, $\rho(z)$ — wavefront curvature radius, (k_x,k_y,k_z) — wave vector with $k = \sqrt{k_x^2+k_y^2+k_z^2} = 2\pi/\lambda$, and $\phi_{\mathrm{piezo}}(t)$ is the phase shift due to the periodic movement of the piezo mirror. We use the paraxial approximation $k_z \gg k_x, k_y$.

Prior to the beam splitter (BS 1), the two beams have identical parameters. To calculate the beam parameters after passing through the lens system we use the formalism of ABCD matrices. In the ABCD method, the beam is characterized by the complex-valued parameter $\frac{1}{q} = \frac{1}{\rho} - \frac{i\lambda}{\pi r^2}$ and the change in the beam parameters when passing through the system lenses is written as follows $q' = \frac{Aq+B}{Cq+D}$, where the resulting system matrix M_{total} is calculated as the product of the matrices $M_{total} = M_{lens1} \cdot M_{fs} \cdot M_{lens2}$ corresponding to the propagation of a beam in empty space at a distance d, $M_{fs} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix}$ and a thin lens with focal length f $M_{lens} = \begin{bmatrix} 1 & 0 \\ -1/f & 1 \end{bmatrix}$.

The main quality metric for interferometer alignment is the visibility of the interference pattern:

$$V = \frac{\max_t(I_{\mathrm{tot}}) - \min_t(I_{\mathrm{tot}})}{\max_t(I_{\mathrm{tot}}) + \min_t(I_{\mathrm{tot}})}, \quad (3)$$

where $I_{tot}(t) = \iint_{-\infty}^{+\infty} I(x, y, t)\mathrm{d}x\mathrm{d}y$ total optical flux incident on the camera; the maximum and minimum are calculated in one pass of the piezo mirror. Visibility, by definition, lies between 0 and 1. For a fully aligned interferometer [figure 5(a)], $\min_t(I_{tot}) = 0$, thus $V = 1$, for a fully misaligned interferometer [figure 5(c)], $\min_t(I_{tot}) \approx \max_t(I_{tot})$, thus $V \approx 0$.

We model the camera matrix as an equidistant grid of 64×64 pixels and calculate the intensity of light in each grid cell. The camera is located at the point $(x,y) = (0,0)$, so the beam propagating through the upper arm of the interferometer is located in the center of the camera. For each pass of the piezo mirror, we simulate 16 interference images corresponding to different $\phi_{\text{piezo}}(t)$. We calculate the visibility of the interference pattern according to the equation 3. Since successful training of RL agents requires millions of interactions with the environment, we implemented the simulator in C ++, and we perform intensity calculations in parallel. The final speed of the simulator on the intel core i7 processor was more than 200 frames, consisting of 16 images, per second. For training agents, the simulator has a standard gym interface.

Examples of interference images obtained using the developed program are shown in fig. 5.

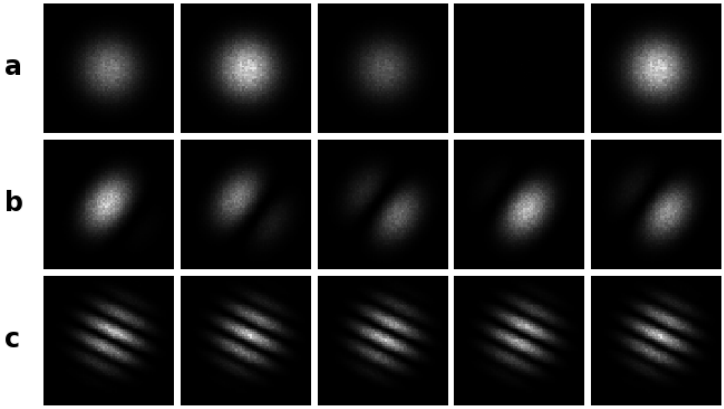

Figure 5 — Interference images for various mirror positions were obtained using a simulation program. (a) Perfectly aligned interferometer, visibility = 1. (b) Weakly misaligned interferometer, visibility = 0.3. (c) Highly misaligned interferometer, visibility = 0.0026. Images from left to right correspond to different points in time.

The geometric dimensions of the interferometer used in the simulation and in the experiments are given in table 1. The focal lengths of lenses lens1 and lens2 are 50 mm. In the tuned state of the interferometer, the distance

between the lenses is equal to the sum of the focal lengths. Lens lens1 is located at a distance of 50 mm from the beam splitter BS1.

Table 1 — Parameters of the interferometer, mm.

parameter	length	width	distance to camera	beam radius
value	200	300	100	0.71

The maximum offsets from the aligned position for optical elements are given in the 2 table. The values were chosen to reproduce the entire spectrum of interference patterns observed in the experiment.

Table 2 — The maximum offset value of each optical element. Mirror angles are in radians, lens positions are in millimeters.

parameter	mirror 2, x	mirror 2, y	BS 2, x	BS 2, y	lens 2
value	$2.6 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	$0.9 \cdot 10^{-3}$	7.5

In the following sections, we formulate the task of alignment of the interferometer in terms of reinforcement learning, present two agents trained with continuous and discrete action spaces, and show evaluation results for the agents in the experimental setup.

We view the interferometer alignment as a Markov decision process. The state of the environment s is a sequence of 16 interference images obtained in one pass of the piezo mirror. The actions of an agent must control the position of the optical elements of the interferometer. We implement the action space in two ways — as discrete and continuous. In the case of the discrete action space, the agent samples actions from a predetermined set of steps with different sizes. This allows the agent to use both large and small amplitude steps with equal confidence. In our implementation, the agent can change the positions of the optical elements independently and only in one direction at a time. The agent was trained using the DQN [19] algorithm.

This method is simple, but it limits the number of optical elements that the agent can operate simultaneously, as well as the speed of the interferometer alignment. To level these restrictions, we train the second agent to align the interferometer using a continuous action space. In this case, the agent can operate simultaneously with all optical elements and change their position by an arbitrary value. The action of the agent a is an N-dimensional vector $a \in (-1, 1)^N$. We use TD3 [20] algorithm to train the agent. Since at the end of the interferometer alignment, the agent's actions become two orders of magnitude less than at the beginning we need to rescale them:

$$a = \begin{cases} sign(a) \cdot 1000^{|a|-1} & \text{if } |a| > 0.17 \\ 0 & \text{if } |a| \leq 0.17 \end{cases} \tag{4}$$

Such a renormalization leads to actions in the interval $|a| \in \{0\} \cup [2.5 \cdot 10^{-3}, 1]$.

Next, we substantiate the choice of the reward function. The visibility of the interference pattern V is not a good reward, as it does not allow the agent to differentiate states with V close to 1, which is important in experiments. We chose the reward function $r = V - \log(1 - V)$. This reward function allows the agent to distinguish between the states close to the aligned position.

In order to transfer the trained agent on the experimental setup without substantial performance loss we trained it in simulation with environment randomizations based on the uncertainty of the setup parameters. At the beginning of each episode, we varied the beam radius by $\pm 20\%$, since this parameter is difficult to measure accurately. Changing the radius also helps to overcome the deviation of the experimental beam profile from the Gaussian distribution. The following randomizations were used at each step. First, we varied the brightness of the interference pattern by $\pm 30\%$, which simulates different camera exposure times. Second, we've added 20% white noise to each pixel, which can be seen as camera noise or dust on the camera 4. Thirdly, we added a cyclic shift of the interference images obtained in one pass of the piezo mirror and randomized the ratio of the times of forward and backward passes of the piezo mirror. The ratio of the number of images obtained during the forward pass was always greater than 50%.

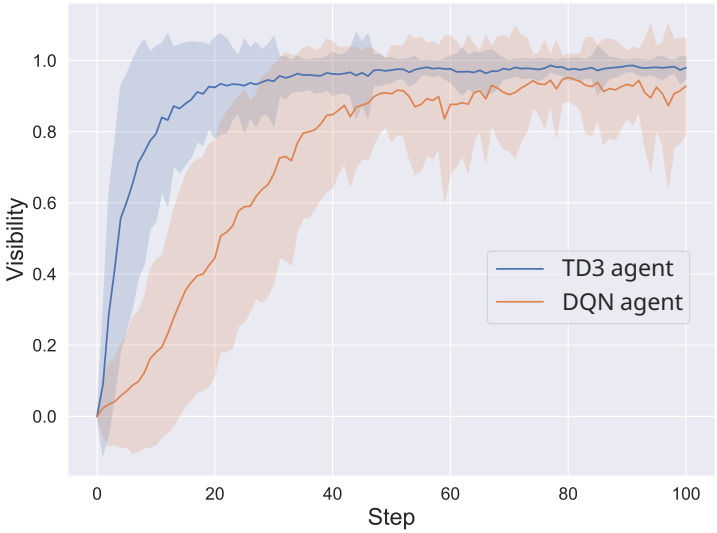

Figure 6 — Evaluation of the agents on experimental setup.

After, we describe the main characteristics of the developed hardware-software complex and present the results of the evaluation of the agents on the

experimental setup. We compare the quality of the interferometer alignment of our RL agents with an expert. Next we analyze the policy used by the agents when aligning the interferometer. The evaluation results are shown in fig. 6. It presents the achieved visibility as a function of the step number. Results are averaged over 100 episodes. It is seen from fig. 6 that both methods achieve a good value for the visibility of the interference pattern. TD3 agent, which uses the continuous action space, is faster and reaches better visibility than the DQN agent, which operates in the discrete action space.

The evaluation results of the interferometer alignment by the developed methods are presented in the table 3. It can be seen that DQN agent works at the human level in terms of the alignment quality and speed. TD3 agent aligns the interferometer much better than a human.

Table 3 — Comparison of the developed methods with an expert. The table shows the time required to achieve interference visibility of 0.92, 0.95, and 0.98. The percentage of episodes where this visibility was not achieved is shown in brackets.

	$V \geq 0.92$	$V \geq 0.95$	$V \geq 0.98$
Expert	93.9 (**0%**)	103.6 (**0%**)	129.6 (10%)
TD3	**56.16** (**0%**)	**75.06** (**0%**)	**120.1** (**4%**)
DQN	98.7 (7.6%)	116.1 (7.6%)	156.4 (10.6%)

The results of the work were published in the articles [21; 22] at the leading scientific conferences «Neural information processing systems (NeurIPS)» and «Conference on Robot Learning (CoRL)» also the developed program was patented [23].

The third chapter is devoted to the problem of determining the reward function for a multitasking agent with a continuous action space. We choose Unitree A1 [9] robot model as the environment. We developed a reward function that encourages the agent to learn a safe and smooth movement policy at a target speed. The agent was trained and tested in a simulation using the Raisim [24] simulator. The agent was trained using the PPO [25] algorithm as a backbone. We added noise to the observations and applied a force to the torso of the robot in a random direction to force the agent to learn a stable moving policy during training. The observation space contains information about the orientation, speed of the robot, and the position and speed of its joints. The action space was continuous, and each action corresponded to the desired position of the robot joints. Evaluation results are shown in tab. 4 and in fig. 7.

The table 4 presents the evaluation results for the tasks “Move forward”, “Move backward”, “Turn clockwise”, and “Rotate counterclockwise”. It can be seen, that for the tasks “Move forward” and “Move backward”, the learned policy is more stable, and the number of steps is close to the maximum length

of the episode. In the "Turn Clockwise" and "Turn Counterclockwise" tasks, the learned policy is less stable, resulting in lower overall rewards and a lower average number of steps. The reason for this may be that the random force, which destabilizes a turning agent more than one moving straight.

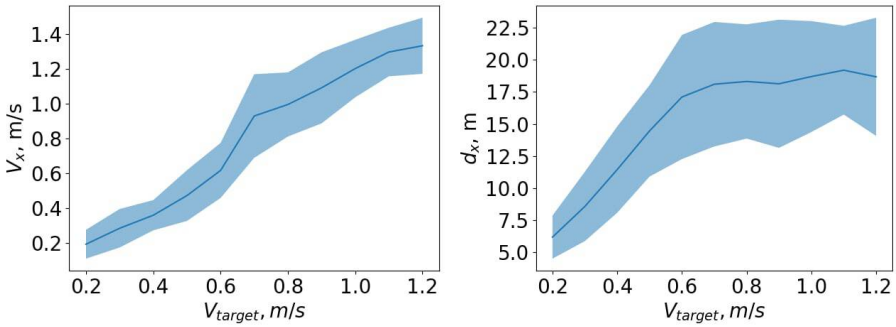

Figure 7 — Evaluation of the task "Move forward at a given speed". (a) Speed of the agent V_x as a function of target speed V_{target}. (b) Distance traveled by the agent by the end of episode d_x as a function of the target speed V_{target}.

Figure 7 shows the evaluation results in the "Move forward at a given speed" task. It can be seen from fig. 7a that the trained agent is able to move at different speeds, and the velocity is close to the target speed. The distance traveled by the agent along the x axis is shown in fig. 7b. At the target speed $V_{target} \in (0,\ 0.6)$ m/s, the distance traveled by the end of the episode increases. When the target speed is greater than 0.6 m/s, the distance traveled is almost constant and equal to the distance from the agent to the edge of the environment.

Table 4 — Evaluation results for the Unitree A1 robot in simulation.

Task	Move forward	Move backward	Turn clockwise	Turn counterclockwise
Return	529 ± 124	508 ± 98	85 ± 219	165 ± 155
Number of steps	3263 ± 633	3136 ± 545	2057 ± 1208	2095 ± 1325

The obtained results show that the linear and angular speed of the Unitree A1 robot can be controlled by the proposed reinforcement learning method with good quality.

The fourth chapter is devoted to the study of hierarchical reinforcement learning methods in procedurally generated environments. We choose the game of NetHack as our environment. The NetHack environment was proposed as a test for machine learning algorithms in 2020 [10]. In this

game, an agent travels through a procedurally generated dungeon. The game has an ASCII interface shown in fig. 8.

Figure 8 — The game of NetHack.

This task proposes several challenges for the reinforcement learning methods. The first challenge is a multi-modal observation space which consists of a dungeon represented as an image, text message in the natural language, and hero stats in tabular form. Also, the reward provided by the environment is sparse. Finally, the actions of an agent are keystrokes, so the action space is combinatorially large. To overcome these challenges, we developed a method that combines reinforcement learning algorithms, graph navigation algorithms, and algorithms based on expert knowledge 1. We implemented a separate policy for each subtask that arises during the game, such as fighting monsters, exploring the dungeon, and handling objects. At each step, we select the best policy in accordance with its priority. We decided to train a monster combat policy with reinforcement learning since it is responsible for the most of the reward and requires complex behavior. We trained the combat policy with the IMPALA [26] algorithm and discrete action space. The remaining policies were implemented using graph navigation algorithms. This approach allowed us to outperform other methods in this task. The results were published at one of the leading machine learning conferences NeurIPS (competition track) [27].

Algorithm 1: RAPH agent

```
Data: view_distance, agent, hard_coded_skills
state, done ← env.reset(), False;
while not done do
    action_queue = parse_message(state);
    if action_queue then
        state, reward, done, info = env.step(action_queue);      /* We
         have a prompt to response */
        continue
    end
    monster_distance, preprocessed_state = parse_dungeon(state);
    if monster_distance <view_distance then
        action_queue = agent.act(preprocessed_state);
    else
        action_queue = first_fit(hard_coded_skills,
         preprocessed_state);          /* Select non-rl action on
         first-fit basis */
    end
    state, reward, done, info = env.step(action_queue);
end
```

CONCLUSION

Conclusion presents the main results of the work, which are as follows:

1. We have developed a computer model of the optical Mach-Zehnder interferometer. Based on the computer model, we developed an environment for training reinforcement learning agents to align an optical interferometer. The environment can reproduce the interference patterns obtained in a Mach-Zehnder interferometer with an arbitrary arrangement of mirrors and lenses with high fidelity.
2. The process interferometer alignment was formulated in terms of a Markov decision process. We determined the reward function, observation, and action spaces. The proposed reward function makes it possible to distinguish the states with high visibility. We have proposed a transformation of the action space that allows the agent to operate with continuous actions of different orders of magnitude.
3. We have developed and implemented reinforcement learning algorithms capable of operating on actions of various magnitudes and are resistant to optical noises.
4. The developed algorithms were trained and successfully evaluated on the experimental setup. The evaluation results show that the quality of the interferometer alignment for the agent with continuous actions was significantly higher than that of a skilled professional.

5. We have developed a policy learning method to control the walking robot with target linear and angular velocities. Our designed reward function enforces the agent to learn a smooth policy. Evaluation results show that the trained agent can perform well in a simulated environment.
6. We have developed a hybrid neuro-symbolic method to control a virtual agent in the NetHack environment. The proposed method proved its effectiveness during the NeurIPS NetHack Challenge and took first place.

References

1. Reward is enough [Текст] / D. Silver [и др.] // Artificial Intelligence. — 2021. — Окт. — Т. 299. — С. 103535. — URL: https://doi.org/10.1016/j.artint.2021.103535.
2. Human-level control through deep reinforcement learning [Текст] / V. Mnih [и др.] // Nature. — 2015. — Т. 518, № 7540. — С. 529—533. — URL: https://doi.org/10.1038%2Fnature14236.
3. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm / D. Silver [и др.] // ArXiv. — 2017. — Т. abs/1712.01815.
4. Mastering the Game of Go with Deep Neural Networks and Tree Search / D. Silver [и др.] // Nature. — 2016. — Янв. — Т. 529, № 7587. — С. 484—489.
5. Grandmaster level in StarCraft II using multi-agent reinforcement learning [Текст] / O. Vinyals [и др.] // Nature. — 2019. — Окт. — Т. 575, № 7782. — С. 350—354. — URL: https://doi.org/10.1038/s41586-019-1724-z.
6. Interferobot: aligning an optical interferometer by a reinforcement learning agent [Текст] / D. Sorokin [et al.] // Advances in Neural Information Processing Systems. Vol. 33 / ed. by H. Larochelle [et al.]. — Curran Associates, Inc., 2020. — P. 13238—13248. — URL: https://proceedings.neurips.cc/paper/2020/file/99ba5c4097c6b8fef5ed774a1a6714b8-Paper.pdf.
7. Aligning an optical interferometer with beam divergence control and continuous action space [Текст] / S. Makarenko [и др.] // Proceedings of the 5th Conference on Robot Learning. Т. 164 / под ред. A. Faust, D. Hsu, G. Neumann. — PMLR, 2022. — С. 918—927. — (Proceedings of Machine Learning Research). — URL: https://proceedings.mlr.press/v164/makarenko22a.html.
8. *Sorokin, D. I.* Learning Various Locomotion Skills from Scratch with Deep Reinforcement Learning [Текст] / D. I. Sorokin, D. L. Babaev // Advances in Neural Computation, Machine Learning, and Cognitive Research VI / под ред. B. Kryzhanovsky [и др.]. — Cham : Springer International Publishing, 2023. — С. 322—329.
9. https://www.unitree.com/products/a1/.
10. The NetHack Learning Environment / H. Küttler [и др.] // Advances in Neural Information Processing Systems. Т. 33 / под ред. H. Larochelle [и др.]. — Curran Associates, Inc., 2020. — С. 7671—7684. — URL: https://proceedings.neurips.cc/paper/2020/file/569ff987c643b4bedf504efda8f786c2-Paper.pdf.

11. Solving Rubik's Cube with a Robot Hand [Текст] / OpenAI [и др.] // arXiv e-prints. — 2019. — Окт. — arXiv:1910.07113. — arXiv: 1910.07113 [cs.LG].

12. Magnetic control of tokamak plasmas through deep reinforcement learning [Текст] / J. Degrave [и др.] // Nature. — 2022. — Февр. — Т. 602, № 7897. — С. 414—419. — URL: https://doi.org/10.1038/s41586-021-04301-9.

13. *Fabri, C.* On the Application of Interference Phenomena to the Solution of Various Problems of Spectroscopy and Metrology [Текст] / C. Fabri, A. Pérot // Astrophysical Journal. — 1899. — Т. 9. — С. 87.

14. LIGO: the Laser Interferometer Gravitational-Wave Observatory [Текст] / B. P. Abbott [и др.] // Reports on Progress in Physics. — 2009. — Июнь. — Т. 72, № 7. — С. 076901. — URL: https://doi.org/10.1088/0034-4885/72/7/076901.

15. Virgo: a laser interferometer to detect gravitational waves [Текст] / T. Accadia [и др.] // Journal of Instrumentation. — 2012. — Март. — Т. 7, № 03. — P03012—P03012. — URL: https://doi.org/10.1088/1748-0221/7/03/p03012.

16. *Kandpal, H.* Simple method for measurement of surface roughness using spectral interferometry [Текст] / H. Kandpal, D. Mehta, J. Vaishya // Optics and Lasers in Engineering. — 2000. — Сент. — Т. 34, № 3. — С. 139—148. — URL: https://doi.org/10.1016/s0143-8166(00)00098-1.

17. Generating optical Schrödinger kittens for quantum information processing. [Текст] / A. Ourjoumtsev [и др.] // Science. — 2006. — Т. 312, № 5770. — С. 83—86.

18. Enlargement of optical Schrödinger's cat states [Текст] / D. V. Sychev [и др.] // Nat. Photonics. — 2017. — Т. 11. — С. 379—382.

19. Human-level control through deep reinforcement learning [Текст] / V. Mnih [и др.] // Nature. — 2015. — Февр. — Т. 518, № 7540. — С. 529—533. — URL: https://doi.org/10.1038/nature14236.

20. Continuous control with deep reinforcement learning [Текст] / T. P. Lillicrap [и др.] // arXiv preprint arXiv:1509.02971. — 2015.

24. *Hwangbo, J.* Per-Contact Iteration Method for Solving Contact Dynamics [Текст] / J. Hwangbo, J. Lee, M. Hutter // IEEE Robotics and Automation Letters. — 2018. — Т. 3, № 2. — С. 895—902.

25. Proximal Policy Optimization Algorithms [Текст] / J. Schulman [и др.] // ArXiv. — 2017. — Т. abs/1707.06347.

26. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures [Текст] / L. Espeholt [и др.] // arXiv e-prints. — 2018. — Февр. — arXiv:1802.01561. — arXiv: 1802.01561 [cs.LG].