PEQUENO ROBÔ UNIAXIAL DE DUAS RODAS PARA LABORATÓRIO EDUCACIONAL

SMALL-SIZED UNIAXIAL TWO-WHEEL ROBOT FOR EDUCATIONAL LABORATORY

МАЛОГАБАРИТНЫЙ ОДНООСНЫЙ ДВУХКОЛЁСНЫЙ РОБОТ ДЛЯ УЧЕБНОЙ
ЛАБОРАТОРИИ

LELKOV, Konsnantin S.^{1*}; KURIS, Eduard D.²; STOLYAROV, Yan V.³; NIKITENKO, Ilya A.⁴;
KRAPIVIN, Sergei V.^{5,6};

^{1,2,3,4,5} Moscow Aviation Institute (National Research University), Department of Automated Complexes of
Orientation and Navigation Systems, 4 Volokolamskoe Highway, zip code 125993, Moscow – Russian
Federation

⁶ K.G. Razumovsky Moscow State University of Technologies and Management (the First Cossack University),
Department of Robotics Center, 73, Zemlyanoy Val Str., zip code 109004, Moscow, Russian Federation

* Correspondence author
e-mail: pec-orange@mail.ru

Received 01 March 2019; received in revised form 28 March 2019; accepted 17 April 2019

RESUMO

Este artigo é dedicado ao desenvolvimento do robô uniaxial de duas rodas (UTR) construído como o pêndulo superior. Este robô destina-se a ser usado como um modelo simples e barato para aulas de laboratório de engenharia robótica. Esse robô pode ser usado para fins educacionais em instituições de ensino superior, bem como em cursos de engenharia robótica em escolas e instituições de ensino suplementar. A unidade experimental UTR e suas especificações gerais foram desenvolvidas. O artigo descreve a síntese do modelo matemático, sistema de controle e composição de hardware simplificado da UTR. Está provado que as equações simplificadas estão corretas. Além disso, apresenta os algoritmos de formação de torque de motores de acionamento baseados em medições de sensores com diferentes naturezas físicas e diferentes métodos de filtragem. O possível diagrama de variantes de fusão de dados é apresentado. No processo de modelagem em meia escala, a pesquisa de movimentos de robôs foi realizada simultaneamente com o registro de informações primárias de todos os sistemas de medição. Como um movimento definido, a estabilização em relação ao horizonte a partir da posição quando o ângulo de rotação de uma plataforma foi de 40 graus foi escolhida. Tem sido determinante que os motores do robô podem aumentar o torque variando de 0,05 a 0,22 Nm. A combinação do modelo matemático descrito, algoritmos de controle e características específicas de determinados componentes estruturais garantem a eficiência da aplicação de UTR no processo educacional. Diversas variantes do sistema de controle UTR são descritas e os resultados da simulação são apresentados. O artigo também inclui os resultados da simulação de hardware.

Palavras-chave: robô uniaxial de laboratório de duas rodas, processo de ensino e aprendizagem, algoritmos, controle de movimento ao longo de uma trajetória, navegação.

ABSTRACT

This article is dedicated to the development of the uniaxial two-wheel robot (UTR) constructed as the upper pendulum. This robot is intended to be used as a simple and inexpensive model for robotic engineering laboratory classes. Such robot can be used for educational purposes in higher education institutions, as well as in robotic engineering courses in schools and supplementary education institutions. Experimental UTR unit and its general specifications have been developed. The article describes the synthesis of UTR's simplified mathematical model, control system and hardware composition. It is proven that the simplified equations are correct. In addition, it presents the algorithms of driving engines torque formation based on measurements of sensors with various physical nature and different filtration methods. Possible data merging variants' diagram is presented. In the process of the half-full scale modelling, the researching of robot movements was conducted simultaneously with the recording of primary information from all measuring systems. As a defined motion, the stabilisation concerning the horizon from the position when the rotation angle of a platform was 40 degrees has been chosen. It has been determining that the robot's engines can enhance torque ranging from 0.05 to 0.22 Nm.

The combination of the described mathematical model, control algorithms and specific features of certain structural components ensure the efficiency of UTR application in the educational process. Several UTR control system variants are described, and the simulation results are presented. The article also includes the results of hardware simulation.

Keywords: *uniaxial two-wheel laboratory robot, teaching and learning process, algorithms, control of motion along a trajectory, navigation.*

АННОТАЦИЯ

Статья посвящена разработке обладающего верхней маятниковостью одноосного двухколёсного робота (ОДР). Этот робот предназначен для использования в качестве простой и недорогой модели для лабораторных занятий по робототехнике. Такой робот может использоваться в образовательных целях в высших учебных заведениях, а также на курсах робототехники в школах и учреждениях дополнительного образования. Получена упрощённая математическая модель ОДР. В статье описывается синтез упрощённой математической модели UTR, системы управления и аппаратного состава. Доказано, что упрощённые уравнения верны. Кроме того, представлены алгоритмы формирования крутящего момента приводных двигателей, основанные на измерениях датчиков различной физической природы и различных методов фильтрации. Представлена схема возможных вариантов объединения данных. В процессе полунатурного моделирования исследование движения робота проводилось одновременно с записью первичной информации со всех измерительных систем. В качестве заданного движения была выбрана стабилизация относительно плоскости горизонта из положения, когда угол наклона платформы составлял 40 градусов. Установлено, что двигатели робота могут развивать вращающие моменты в пределах от 0.05 до 0.22 Нм. Сочетание описанной математической модели, алгоритмов управления и особенностей отдельных структурных компонентов обеспечивает эффективность применения ОДР в образовательном процессе. Описаны несколько вариантов системы управления UTR и представлены результаты моделирования. Приведены результаты полунатурного моделирования.

Ключевые слова: *одноосный двухколёсный лабораторный робот, учебный процесс, алгоритмы, управление движением по траектории, навигация.*

1. INTRODUCTION

In the context of learning the academic disciplines in respect of the robotic engineering, it is necessary to reveal the properties of the certain systems of robots, which differ from each other in respect of the functional purposes of these systems (Stilman *et al.*, 2019; Nelson *et al.*, 2017). As concerns the ground uniaxial two-wheel robots (UTR) of the Segway type (Lantau and Bracke, 2018) with the upper pendulosity, the subject-matter under this investigation is as follows: system of stabilisation of the robot platform with respect to the horizon plane; system of stabilisation of the robot motion on the underlying surface; sensors of the initial information on the parameters of the robot motion, as well as on orientation of the robot; algorithms of formation of moments by the driving engines of wheels; autonomous navigation system and main components of the robot structure (Aloshin *et al.*, 2012; Maksimov and Chernomorskiy, 2016; Chernomorskiy *et al.*, 2018). In order to ensure utilisation of such a robot in the learning process, such a robot must also be sufficiently simple and inexpensive.

There exist training models of walking robots and crawling snake robots (Wang *et al.*, 2019), for which the problems that are connected with the investigation of their motion were analysed in detail. In addition, small-sized inexpensive packages were developed for do-it-yourself assembling robots on the basis of the UTR. They contain all necessary structural components, as well as relevant software (SW).

Please see below a description of this standard package (Description of the small-sized robot), assembled representation of which is presented in Figure 1.

Jjrobots B-robot EVO 2 is made as to the structure on the basis of Arduino computing board (Gingl *et al.*, 2019). This package includes documents, necessary open-source software, as well as models of its sections (these models are intended for printing with the help of the 3D printer). In addition, it is also possible to ensure the remote control with the help of special application software for smart phones. In the mode of the remote robot operation, it is possible to set up the speed of motion, as well as the turn rate of the robot, as well as to change coefficients in the laws

of control of the driving engines of wheels. Presence of the lever would help the robot to return to the upright position in the situation when the robot fell "face down". The control system consists of two separate subsystems: a system of stabilisation and speed control system. System of stabilisation is intended for solving the problem of stabilisation of the platform in the equilibrium position with respect to the horizon plane. This system is made as to the PID controller (proportional derivative action controller) (Wu *et al.*, 2014). Information on the angle of deflection of the platform from the horizon plane is transferred to the input of this controller. This information is obtained and collected in accordance with the results of measurements with the help of gyroscopes, accelerometers, and the magnetometer, which are installed in the inertial module MPU6050 (Tripathi *et al.*, 2019). In addition, this information is obtained due to the application of the open-source algorithm of orientation (Madwick algorithm) (Madgwick *et al.*, 2011). The speed control system is intended for solving the problem of keeping the necessary value of the speed, and this system is made as to the PI controller (proportional integral controller). Information on the linear speed of the robot motion is transferred to the input of this controller. This information is calculated on the basis of measurements with the help of encoders, which are installed on the axes of the wheel engines. In addition, it is also possible to change coefficients in the laws of control with the help of the mobile application software. However, there are no mounting devices for installation of additional measuring systems. Mode of integration of information from various measuring instruments is not envisaged as well. Structure of this robot is not sufficiently strong in order to ensure its multiple uses in the learning process. Cost of one package for assembly of Jrobots B-robot EVO 2 is approximately equal to \$140.

2. MATERIALS AND METHODS

The goal of the present article is to develop the structure and software of the small-sized UTR, which is intended for utilisation in various training courses in respect of robotic engineering. The device, which is to be developed, must make it possible to study the following specific features: the mathematical model of the UTR; the architecture of the robot; various sensors; algorithms of the system of stabilisation, a system of the motion control, and system of navigation. In the course of development of this robot, the following structural and general-circuit solutions

were used:

- the monolithic case was made of the plastic material. Therefore, it would be possible to reduce the likelihood of the robot breakdown. In addition, it would essentially simplify assembly of the product due to the minimum quantity of the details, which are used in the course of assembly;
- for the avoidance of possibility of occurrence of deformations (for example, lateral bending of the case), reinforcement ribs were envisaged in the structure of the monolithic case;
- Arduino Nano compact board (Nayyar and Puri, 2016) (dimensions of which are equal to 43 x 19 mm) was selected as the main computer unit (Appendix 1);
- the software, which we have developed, has a special modular structure, which makes it possible to edit the software code. This fact, in its turn, makes it possible to use the developed subprograms in order to establish various laws of control of the robot motion, as well as various methods of processing results of measurements, which were made by the inertial sensors.

The experimental prototype of the robot is presented in Figure 2.

Technical characteristics of the equipment, which is installed within the robot, are presented in Table 1 (Micromechanical Sensor Manufacturer). Estimates indicate that the cost of the developed robot in small batch production will not exceed \$80. Kinematic scheme of the UTR is similar to the kinematic scheme, which was analysed in the article (Maksimov, 2016), and this kinematic scheme is presented in Figure 3.

Conventional notations, which are accepted within this Figure, are as follows: $O_{\Pi}x_{\Pi}y_{\Pi}z_{\Pi}$ is the system of coordinates (SC) of the robot; axis x_{Π} is the longitudinal axis of the robot platform; axis y_{Π} is the axis, which is parallel to the axis of the wheel set, while axis z_{Π} passes through the centre of mass of the robot platform and through the centre of axis of the wheel set. $OX_cY_cZ_c$ is the SC, which is connected with the robot; axes X_c and Y_c coincide with projections of the axes x_{Π} and y_{Π} to the horizon plane, while axis Z_c coincides with the projection of axis z_{Π} to the vertical of the relevant place. $OXYZ$ is the reference system of coordinates, which coincides with the SC $OX_cY_cZ_c$ at the initial moment of

time. Centre of mass of the platform O_{Π} is situated above the centre of the wheel set axle O at the distance l_{Π} . Each of two wheels with masses m_k and radii R are situated at a distance l_k from the centre of the wheel set axle. Angular speeds of two wheels are designated as $\dot{\phi}_{\Pi}, \dot{\phi}_{\Pi}$, respectively (in this case, subscript "п" means "right", while subscript "л" means "left"), $v, \dot{\theta}$ are linear and angular speeds of the robot motion. Stabilisation of the platform with mass m_{Π} in the horizon plane around the axis of the wheel set axle according to the angle α is performed on the basis of the principle of the inertial stabilisation according to the moment of the inertial force $m_{\Pi} \dot{V} l_{\Pi}$. In this case, linear acceleration of the centre of the wheel set axle $\dot{V}$ is formed at the expense of the directed accelerated rotation of wheels, to the engines of which relevant actuating signals, which depend on the measurable angle α , are transferred.

3. RESULTS AND DISCUSSION:

3.1. Mathematical model of the uniaxial two-wheel robot

In accordance with articles (Maksimov, 2016; Aleshin *et al.*, 2017) equations of motion of the UTR are as follows in Equations 1 – 3. where $A, B, J_{cv}, J_{wm}, J_{ct}$ are reduced constant moments of inertia (Equations 4 – 8).

In order to ensure further investigations, it is possible to simplify Equations 1 – 3 due to the structural specific features of the UTR as follows: it is possible to neglect the terms of the second order of smallness. Then, these equations will take on the following form Equations 9 – 11.

Values of the structural parameters of the robot (Figure 2) are presented in Table 2.

Figures 4 and 5 present chart of the change in the linear speed of the UTR motion, as well as a chart of the change in the slope angle of the UTR platform in the course of simulation of full (solid line) and simplified (dashed line) equations of motion. Results of simulation show that the accepted assumptions in the course of simplifying the above equations are correct.

The specific feature of the robot structure, which we have developed, is the availability of the abundant number of sensors of various natures,

and this fact makes it possible to use various combinations of these sensors in the course of complexing computing facilities. In order to ensure the acquisition of information on the platform dynamics the following measuring instruments are used: acceleration meters, gyroscopes, and encoders. The latter are installed on the engines, and they measure the angular speed of rotation of wheels. In addition, the architecture of the robot can be changed as well in accordance with the problems, which were determined in the learning process. For example, the platform, which contains electronic devices, can be raised up over stands in order to increase the upper pendulosity of the platform.

The specific feature of the software is the formation of such system of control, which ensures the possibility of switching between various methods of formation measurements with the purpose of their adaptation to the current mode of operation of the robot. Particularly, if the robot is at a stop, then the acceleration due to gravity will be the main component in the measurements of accelerometers. This fact would make it possible to define the orientation of the robot with respect to the horizon plane through projections of the robot to the axes of the acceleration meter and to use this information in the law of control.

3.2. Control System

Control of the UTR motion is performed through formation of the controlling actions in respect of the linear speed $V_{y\text{np}}$ and angular speed $\dot{\theta}_{y\text{np}}$ of the robot motion with the help of the PID controllers Equations 12 – 13 (Thrun and Levinson, 2013) where α_x is the projection of the linear acceleration UTR to the axis x_{Π} (this projection is measured by the acceleration meter); $\dot{\alpha}$ is the angular speed of rotation of the platform with respect to the axis y_{Π} , which is measured by the gyroscope; V is the linear speed of motion of the platform, which is calculated on the basis of measurements with the help of the encoders, which are installed on the wheels; $V_{y\text{CT}}$ is the steady-state value of the linear speed. Rotational moments on the engines of the left wheel $\tau_{\text{лсв}}$ and the right wheel $\tau_{\text{рпав}}$ in the Expressions 9 – 11 are presented as follows Equations 14 – 15.

Algebraic Equations 14 – 15 determine moments on the engines of wheels, which must be achieved in order to ensure the prescribed mode

of the UTR motion over the selected trajectory on the condition of maintenance of the equilibrium position of the UTR platform with respect to the horizon plane.

3.3. Complexing computing facilities

The above-stated laws of control Equations 12 – 13 correspond to the base variants of these laws, which make it possible to demonstrate the possibility of stabilisation of the UTR platform both in the motion and in the stationary mode. Figure 6 presents the scheme of possible base variants of the various complexing measurements.

From the point of view of the training process, the possibility of changing the laws of control and ensuring various modes of the complexing measurements from various sensors is the problem of the greatest interest. Thus, for example, as concerns the above-described law of control (Equations 12 – 13), the complexing procedure involves selecting the method of formation of certain components of these laws. The first component corresponds to the value, which is proportional to the angle of deviation of the platform. It is possible to assume that this component is the acceleration projection onto the longitudinal axis of the platform x_{Π} (assembly unit 1 "Acceleration meter" in Figure 6) or that this component is the integral of the angular speed of rotation of the platform around the wheel set axle y_c (assembly units 4 and 6 in Figure 6). The second component describes the value, which is proportional to the speed of change in deviation of the platform from the horizon plane. The simplest base variant of formation of this component is as follows: measurement of the angular speed with the help of the gyroscope, which is installed on the axis y_{Π} . However, due to the great instability of the measurements, which are taken by the micromechanical sensors, it is recommended to use various methods of filtering measurements of the angular speed, for example, low-frequency and high-frequency filters, as well as median filter and/or Kalman filter (Li *et al.*, 2015) (assembly units 5 and 6 in Figure 6). The third component characterises the speed of the linear displacement of the UTR. This component can be obtained by calculation of the linear speed of motion in accordance with the results of measurements, which were made by encoders (assembly unit 7 in Figure 6), or by integration of indications of the acceleration meter, which is installed on the axis x_{Π} , in the following situations: if robot is at a stop

or if deviation of the UTR platform from the horizon plane is very insufficient (assembly units 1 and 2 in Figure 6). In addition, it is possible to ensure complexing the measurements, which were made by encoders, and the integral of the longitudinal acceleration (assembly units 1, 2, 3, 7 in Figure 6).

3.4. Simulation

Simulation of the equations of motion (Equations 9 – 11), as well as simulation of the system of control (Equations 12 – 13) was performed in the 'Matlab' programming environment under the following initial conditions: deviation of the UTR platform from the horizon plane $\alpha_0 = 0.2$ radian; linear speed of the UTR motion $v_0 = 0$ m/s; angular speed of the UTR motion $\dot{\theta}_0 = 0$ radians. Uncertainty of measurements of the on-board acceleration meter is determined in accordance with the normal law of distribution (Krithikadatta, 2014) with the zero expected value, as well as with the standard deviation equal to 0.9 m/s^2 . Uncertainty of measurements of the linear speed of motion is determined in accordance with the normal law of distribution with the zero expected value and with the standard deviation equal to 0.2 m/s^2 . Uncertainty of measurements of the on-board gyroscope is a random value in accordance with the uniform law of distribution, zero expected value, and the standard deviation equal to 0.03 radian. Figures 7 – 9 present results of simulation of the UTR motion for three variants of formation of the law of control, which are presented in Figure 6.

As it follows from Figure 7, in the case of utilisation of the law of control, which is implemented with the help of the assembly units 4, 5, and 6 (Figure 6), the entire system becomes unstable. This is caused by the great uncertainty due to determining the slope angle as the integral of the angular speed. In this case, we not only integrate errors in measurements of the gyroscope at each discretisation interval but also obtain the delayed action of the actuating signal in the Integrating device, in consequence of which this robot simply has no time to compensate deviation of the platform from the horizon plane.

In the case of utilisation of the law of control, which is implemented with the help of the assembly units 1, 2, 5, and 6 (Figure 6), results of simulation (Figure 8) show that the entire system is stable. Despite the fact that in this case the delayed action of the actuating signal is observed as well, the integral of the measured acceleration

does not exert any essential influence upon the stability of the entire system (this integral is used in the course of formation of the integral component in the law of control).

In addition, we have analysed the situation of the UTR motion on the condition that there is no any integral components in the law of control (Equation 12). Therefore, assembly units 1, 5, and 6 (Figure 6), respectively, are used in the course of formation of the proportional and differential components (Equation 16).

One can see from Figure 9 that despite the fact of absence of the integral component in the law of control, the entire system is stable. However, in this case, the robot cannot maintain the equilibrium position of the platform, while keeping its balance on the same place. The linear speed of motion will increase until the moment when the angle of deviation of the platform from the horizon plane will become equal to zero. Then the robot will continue its motion with constant speed.

3.5. Hardware simulation

In order to check the working ability of the UTR prototype, which we have developed (Figure 2), we have performed the hardware simulation of the operation of its system of control (Egorov, 2018).

Investigation of the robot motion has been performed simultaneously with recording the initial information from all measuring systems. In order to obtain the reference controlling actions for the prescribed motion of the UTR, the recorded measurements of sensors were transferred to the input of the mathematical model. Result of this in-line simulation (Figure 10) was as follows: we have obtained possibility to compare the controlling actions, which were directly formed by the on-board controller of the robot, with the reference actions (Figures 11, 12), which were obtained as the result of calculations with the help of the mathematical model.

We have selected stabilisation of the UTR with respect to the horizon plane from the position, when the angle of slope of the platform was equal to 40 degrees, as the prescribed motion of the UTR.

As it follows from a comparison of the graphs, which are presented in Figures 10 – 12, the controlling actions, which are formed by the on-board controller, have the same form as the form of the reference is controlling actions. However, actual values of the moments on the

engines of wheels are characterized by the essentially greater amplitude of oscillations as compared with the values, which were obtained with the help of the mathematical model. This is explained by the fact that engines have a certain minimum rotational moment, which must be achieved in order to overcome forces of friction in the axes of engines (Shabanov *et al.*, 2016), as well as in the points of contact of wheels with the underlying surface.

Therefore, engines of the robot can develop rotational moments within limits from 0.05 up to 0.22 Nm. The lower limit of moments is not taken into account in the mathematical model.

4. CONCLUSIONS:

The developed structure of the UTR, mathematical model of the UTR, and relevant software comply with the following requirements, which are established for the architecture of the state-of-the-art information and measurement controlling systems, namely: technological modularity, hierarchy principle, flexibility, reliability, and openness. It was demonstrated that the software, which we have developed, makes it possible to simulate the motion of the robot at various combinations of sensors and algorithms of the system of stabilisation, a system of the motion control, and system of navigation. In addition, it is possible to demonstrate the operation of the system of control taking into account the complexing procedure on the basis of various filtering algorithms. Results of the hardware simulation make it possible to demonstrate the difference between the mathematical model and the robot motion. The UTR, which we have analysed, makes it possible to clearly demonstrate the physical principles, which are used in the robotic engineering, and it can be used not only in the learning process of the higher education institutions but also in the course of solving problems of the educational robotic engineering in the higher forms of the secondary schools and in the institutions for supplementary education.

5. REFERENCES:

- [1] Aleshin, B.S., Kuris, E.D., Lelkov, K.S., Maksimov, V.N., Chernomorsky, A.I. *Proceedings of the RAS. Theory and Control Systems*, **2017**, 1, 150-160.
- [2] Aleshin, B.S., Chernomorskiy, A.I., Feshchenko, S.V. *Orientation, Navigation and Stabilization of Uniaxial Wheel*

- Modules, Moscow: Publishing House of the Moscow Aviation Institute, **2012**.
- [3] Chernomorskiy, A.I., Kuris, E.D. Melnikov, V.Ye. *Electronic Journal "Proceedings of the MAI"*, **2018**, 1, 3-14.
- [4] Description of small-sized robot, <https://www.jjrobots.com/much-more-than-a-self-balancing-robot/>, accessed April 2019.
- [5] Egorov, A.A. *Automation and IT in Power Engineering*, **2018**, 7(108), 18-20.
- [6] Gingl, Z., Mingesz, R., Makan, G., Mellar, J. *Physics Education*, **2019**, 54(2), 025010. doi: 10.1088/1361-6552/aafa41.
- [7] Krithikadatta, J. *Journal of Conservative Dentistry*, **2014**, 17(1), 96-97. doi:10.4103/0972-0707.124171
- [8] Lantau, J.-M., Bracke, M. Mathematical Modelling of Dynamical Systems and Implementation at School, *Tenth Congress of the European Society for Research in Mathematics Education*, Dublin, Ireland, **2018**, 121-136
- [9] Li, Q., Li, R., Ji, K., Dai, W. Kalman Filter and Its Application, *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, Tianjin, China, **2015**, 74-77. doi: 10.1109/ICINIS.2015.35
- [10] Madgwick, S.O.H., Harrison, A.J.L., Vaidyanathan, A. Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm, *Proceedings of the IEEE International Conference on Rehabilitation Robotics*, Zurich, Switzerland, **2011**, 1-7.
- [11] Maksimov, V.N. *Integrated Navigation and Stabilization System Platform Uniaxial Wheel Module for Monitoring Ground Infrastructure*, Moscow: Moscow Aviation Institute, **2016**.
- [12] Maksimov, V.N., Chernomorskiy, A.I. *Gyroscopy and Navigation*, **2016**, 92(1), 116-132.
- [13] Micromechanical Sensor Manufacturer, <https://www.st.com>, accessed March 2019.
- [14] Nayyar, A., Puri, V. *British Journal of Mathematics & Computer Science*, **2016**, 15(5), 1-12. doi: 10.9734/BJMCS/2016/24854.
- [15] Nelson, G., Saunders, A., Playter, R. *The Petman and Atlas Robots at Boston Dynamics*, Dordrecht: Springer, **2017**.
- [16] Shabanov, A.U., Zaitsev, A.B., Metelev, A.A., Pystovalov, Y.P. *St. Petersburg Polytechnic University Journal of Engineering Sciences and Technology*, **2016**, 3(249), 15-21.
- [17] Stilman, M., Wang, J., Teeyapan, K., Marceau, R. Optimized Control Strategies for Wheeled Humanoids and Mobile Manipulators, *9th IEEE-RAS International Conference on Humanoid Robots*, Paris, France, **2009**, 568-573.
- [18] The code for uniaxial robot platform for "Robotic Engineering" course, https://github.com/pec-orange/Balance_one, accessed April 2019.
- [19] Thrun, S., Levinson, J.S., *Systems, Methods and Devices for Adaptive Steering Control of Automotive Vehicles*, US Patent 8,392,064 **2013**.
- [20] Tripathi, V., Bansal, A., Gupta, R. Development of Self-stabilizing Platform Using MPU-6050 as IMU, *Select Proceedings of ICSC 2018, Advances in Signal Processing and Communication*, Springer, Singapore, **2019**, 373-382, https://link.springer.com/chapter/10.1007/978-981-13-2553-3_36, accessed April 2019.
- [21] Wang, L., Yang, Yu., Correa, G., Karydis, K., Fearing, R.S. *OpenRoACH: A Durable Open-Source Hexapedal Platform with Onboard Robot Operating System (ROS)*, **2019**, <https://arxiv.org/abs/1903.00131>, accessed April 2019.
- [22] Wu, H., Su, W., Liu, Z. PID controllers: Design and Tuning Methods, *2014 9th IEEE Conference on Industrial Electronics and Applications*, Hangzhou, China, **2014**, 808-813, doi: 10.1109/ICIEA.2014.6931273

$$\ddot{\alpha} = \frac{J_{wn} B \sin 2\alpha}{2(A(\cos \alpha)^2 - J_{cv})} \dot{\theta}^2 + \frac{rl_{\Pi}^2 m_{\Pi}^2 \sin 2\alpha}{2(A(\cos \alpha)^2 - J_{cv})} \dot{\alpha}^2 -$$

$$- \frac{(l_{\Pi} m_{\Pi} \cos \alpha - J_{wn})(\tau_{\text{лев}} + \tau_{\text{прав}})}{A(\cos \alpha)^2 - J_{cv}} - \frac{J_{wn} l_{\Pi} g m_{\Pi} \sin \alpha}{A(\cos \alpha)^2 - J_{cv}} \quad (\text{Eq. 1})$$

$$\dot{V} = \frac{(J_y + m_{\Pi} l_{\Pi}^2) r d m_{\Pi} \sin \alpha}{A(\cos \alpha)^2 - J_{cv}} \dot{\alpha}^2 - \frac{rl_{\Pi} m_{\Pi} B \sin 2\alpha \cos \alpha}{2(A(\cos \alpha)^2 - J_{cv})} \dot{\theta}^2 -$$

$$- \frac{(J_y + m_{\Pi} l_{\Pi}^2 - r d m_{\Pi} \cos \alpha)(\tau_{\text{лев}} + \tau_{\text{прав}})}{A(\cos \alpha)^2 - J_{cv}} - \frac{g r l_{\Pi}^2 m_{\Pi}^2 \sin 2\alpha}{2((\cos \alpha)^2 - J_{cv})} \quad (\text{Eq. 2})$$

$$\ddot{\theta} = - \frac{\sin 2\alpha (l_{\Pi}^2 m_{\Pi} + J_x - J_z)}{\left(2J_d + J_z + J_{ct} (\sin \alpha)^2 + \frac{2l_{\text{к}}^2}{r^2} (J_{\text{к}} + r^2 m_{\text{к}}) \right)} \dot{\theta} \dot{\alpha} +$$

$$+ \frac{\frac{l_{\text{к}}}{r} (\tau_{\text{лев}} - \tau_{\text{прав}})}{2J_d + J_z + J_{ct} (\sin \alpha)^2 + \frac{2l_{\text{к}}^2}{r^2} (J_{\text{к}} + r^2 m_{\text{к}})} \quad (\text{Eq. 3})$$

$$B = J_z - J_x - m_{\Pi} l_{\Pi}^2; \quad (\text{Eq. 4})$$

$$A = r l_{\Pi}^2 m_{\Pi}^2; \quad (\text{Eq. 5})$$

$$J_{wn} = \frac{2}{r} J_{\text{к}} + 2r m_{\text{к}} + r m_{\Pi}; \quad (\text{Eq. 6})$$

$$J_{cv} = J_{wn} (J_y + l_{\Pi}^2 m_{\Pi}); \quad (\text{Eq. 7})$$

$$J_{ct} = J_x - J_z + l_{\Pi}^2 m_{\Pi}. \quad (\text{Eq. 8})$$

$$\ddot{\alpha} = - \frac{(l_{\Pi} m_{\Pi} \cos \alpha - J_{wn})(\tau_{\text{лев}} + \tau_{\text{прав}})}{r l_{\Pi}^2 m_{\Pi}^2 (\cos \alpha)^2 - J_{wn} J_y} - \frac{J_{wn} l_{\Pi} g m_{\Pi} \sin \alpha}{r l_{\Pi}^2 m_{\Pi}^2 (\cos \alpha)^2 - J_{wn} J_y}; \quad (\text{Eq. 9})$$

$$\dot{V} = - \frac{(J_y - r l_{\Pi} m_{\Pi} \cos \alpha)(\tau_{\text{лев}} + \tau_{\text{прав}})}{r l_{\Pi}^2 m_{\Pi}^2 (\cos \alpha)^2 - J_{wn} J_y} - \frac{g r l_{\Pi}^2 m_{\Pi}^2 \sin 2\alpha}{2(r l_{\Pi}^2 m_{\Pi}^2 (\cos \alpha)^2 - J_{wn} J_y)}; \quad (\text{Eq. 10})$$

$$\ddot{\theta} = \frac{\frac{l_K}{r}(\tau_{\text{лев}} - \tau_{\text{прав}})}{J_z + (J_x - J_z)(\sin \alpha)^2 + \frac{2l_K^2}{r^2}(J_K + r^2 m_K)} \quad (\text{Eq. 11})$$

$$V_{\text{упр}} = k_a a_x + k_{da} \dot{\alpha} + (V - V_{\text{уст}}) k_b ; \quad (\text{Eq. 12})$$

$$\dot{\theta}_{\text{упр}} = (\dot{\theta} - \dot{\theta}_{\text{уст}}) k_t , \quad (\text{Eq. 13})$$

$$\tau_{\text{лев}} = \frac{V_{\text{упр}} - \dot{\theta}_{\text{упр}}}{2} \quad (\text{Eq. 14})$$

$$\tau_{\text{прав}} = \frac{V_{\text{упр}} + \dot{\theta}_{\text{упр}}}{2} \quad (\text{Eq. 15})$$

$$V_{\text{упр}} = k_a a_x + k_{da} \dot{\alpha} , \quad (\text{Eq. 16})$$

Figure 1. Jirobots B-robot EVO 2

Figure 2. The experimental prototype UTR

Figure 3. Kinematic scheme of the robot

Figure 4. Chart of the linear speed of the UTR motion in the course of simulation of full (solid line) and simplified (dashed line) equations of motion

Figure 5. Chart of change in the slope angle of the platform UTR in the course of simulation of full (solid line) and simplified (dashed line) equations of motion

Figure 6. Scheme of possible base variants of the complexing various measurements

Figure 7. Simulation of the UTR motion in the case of utilisation of the law of control, which is implemented with the help of the assembly units 4, 5, and 6 in Figure 6

Figure 8. Simulation of the UTR motion in the case of utilisation of the law of control, which is implemented with the help of the assembly units 1, 2, 5, and 6 (Figure 6)

Figure 9. Simulation of the UTR motion in the case of utilisation of the law of control in accordance with Equation 15 and Equation 18

Figure 10. Chart of the longitudinal acceleration ϵ of the UTR prototype

Figure 11. Charts of the real (blue line) and reference (red line) rotational moments of engines in the course of stabilisation of the UTR with respect to the horizon plane

Figure 12. Enlarged fragment of Figure 11

Table 1. Technical characteristics of the UTR main equipment

Engines Gekko MR12-100	
Speed of rotation under no-load condition	320 rpm
Maximum torque	0.22 Nm
Gyroscope L3G4200D	
Range of the measurable deviations	±500 degree/s
Frequency of measurements	100 Hz
Sensitivity	0.017 degree/s
Acceleration meter LIS331DLH	
Range of the measurable accelerations	±4g
Frequency of measurements	100 Hz
Sensitivity	0.002g

Table 2. Mass and dimensional parameters of the UTR and calculated values of moments of inertia

Mass of the platform m_{Π}	0.22 kg
Mass of wheels m_k	0.02 kg
Displacement of the centre of mass of the platform l_{Π}	0.017 m
Length of the semiaxle of the wheel set axle l_k	0.085 m
Radius of wheels r	0.018 m
Moment of inertia of the platform J_x	$5.7 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$
Moment of inertia of the platform J_y	$6.4 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$
Moment of inertia of the platform J_z	$1.6 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$
Moment of inertia of wheels J_k	$3.2 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$
Moment of inertia of wheels J_d	$1.6 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$

APPENDIX 1

Arduino Code (The code for uniaxial robot platform for "Robotic Engineering" course)

```
// I2C library
#include <Wire.h>
// Timers library
#include <time.h>
// IMU sensor library
#include <TroykaIMU.h>
// SD card library
#include <SD.h>

// Connections:
// Encoder on LEFT wheel:
const byte encoder0pinA_LEFT = 3; // A pin = interrupt pin 1
const byte encoder0pinB_LEFT = 4; // B pin
// Encoder on Right wheel:
const byte encoder0pinA_RIGHT = 2; // A pin = interrupt pin 0
const byte encoder0pinB_RIGHT = 5; // B pin

// Motor connections:
const int rightSpeed = 6; // PWM for right wheel rotation speed (pin 1 on L293D chip)
const int leftSpeed = 9; // PWM for left wheel rotation speed (pin 9 on L293D chip)
const int logic_0 = 7; // Wheels rotation direction. High – forward (pin 15 on L293D chip)
const int logic_1 = 10; // Wheels rotation direction. High – backward (pin 10 on L293D chip)

// SD card module:
const int CS_pin = 8;

// Accelerometer class object
Accelerometer accel;
// Gyroscope class object
Gyroscope gyro;

// LEFT encoder variables:
byte encoder0PinALast_LEFT; // the last value on A pin
boolean Direction_LEFT; // the rotation direction
int duration_LEFT = true; // the number of the pulses
void wheelSpeed_LEFT(); // encoder function

// RIGHT encoder variables:
byte encoder0PinALast_RIGHT; // the last value on A pin
int duration_RIGHT; // the number of the pulses
boolean Direction_RIGHT = true; // the rotation direction
void wheelSpeed_RIGHT(); // encoder function

// Motion control coefficients
float ka = 2.6;
float kda = 0.22;
float kb = 1.9;
float kt = -0.002;

// Logger time
long start_time = 0;
// Static adjustments for engines imperfection
const int leftMotor_adj = -28; // static adjustment for left motor
const int rightMotor_adj = 0; // static adjustment for right motor
```

```

// main cycle operation time
const int cycle_time = 75; // ms
// maximum allowed rotation speed
const int max_w = 220;

void setup()
{
  // set pins mode:
  pinMode(rightSpeed, OUTPUT);
  pinMode(leftSpeed, OUTPUT);
  pinMode(logic_0, OUTPUT);
  pinMode(logic_1, OUTPUT);
  pinMode(CS_pin, OUTPUT);
  pinMode(encoder0pinB_LEFT, INPUT);
  pinMode(encoder0pinB_RIGHT, INPUT);

  // start serial port:
  Serial.begin(57600); // Initialize the serial port

  // Initialize encoders
  attachInterrupt(0, wheelSpeed_RIGHT, CHANGE); // pin 2 on arduino
  attachInterrupt(1, wheelSpeed_LEFT, CHANGE); // pin 3 on arduino

  // Init gyroscope
  gyro.begin();
  // Set gyroscope sensivity mode
  gyro.setRange(RANGE_250DPS);
  // Init accelerometer
  accel.begin();
  // Set accelerometer sensivity mode
  accel.setRange(RANGE_2G);

  // create a start time mark
  start_time = millis();

  // Check if SD card is connected
  if (!SD.begin(CS_pin))
  {
    Serial.println("Card Failure");
    return;
  }
  Serial.println("Card Ready");
  // Check is Data.txt already exists
  if (SD.exists("Data.txt"))
    SD.remove("Data.txt");
  File logFile = SD.open("Data.txt", FILE_WRITE);
  // check if file was created succesefully
  if (logFile)
  {
    String header = "Tang, accX, Time, Velocity, Power_left, Power_right, Angle";
    logFile.println(header);
    logFile.close();
    Serial.println(header);
  }
  else
    Serial.println("Couldn't open log file");
}

```

```

}

void loop()
{
  float Tang = gyro.readDegPerSecX() * 3.14 / 180; // this is in radians
  float ACSEL = accel.readAY();
  float Kyrs = gyro.readDegPerSecZ();

  Serial.print("Duration left: ");
  Serial.print(duration_LEFT);
  Serial.print(", Duration right: ");
  Serial.println(duration_RIGHT);

  // convert velocity from odometer pulses to meters per second
  float frequency = 1000 / cycle_time;
  float Velocity = ( duration_LEFT + duration_RIGHT ) * 0.021 * 30 * frequency * 3.14 / 180 / 2;
  // Velocity control signal
  float control_Vy = ka * ACSEL + kda * Tang + kb * (Velocity);
  // Angular rate control signal
  float control_tetta = kt * (Kyrs - 0);

  Serial.print("ACSEL: ");
  Serial.print(ACSEL);
  Serial.print(", Tang: ");
  Serial.print(Tang);
  Serial.print(", Velocity: ");
  Serial.println(Velocity);

  // Torque values for both engines, in N*m
  float Tau_l = (control_Vy - control_tetta) / 2;
  float Tau_r = (control_Vy + control_tetta) / 2;
  // PWM signal for both engines (in [0 255] range)
  float w_Tau_l = 10.5 * abs(Tau_l) + 110;
  float w_Tau_r = 10.5 * abs(Tau_r) + 110;

  // Set movement direction
  if (Tau_l > 0)
  {
    digitalWrite(logic_0, HIGH); // Move forward
    digitalWrite(logic_1, LOW);
  }
  else
  {
    digitalWrite(logic_0, LOW);
    digitalWrite(logic_1, HIGH); // Move backward
  }

  // limit w_Tau to max values and make sure they're decimal
  w_Tau_l = ceil(w_Tau_l);
  w_Tau_r = ceil(w_Tau_r);
  if (abs(w_Tau_l) > max_w)
    w_Tau_l = max_w;
  if (abs(w_Tau_r) > max_w)
    w_Tau_r = max_w;

  Serial.print("Tau_l: ");

```

```

Serial.print(Tau_l);
Serial.print(", Tau_r: ");
Serial.print(Tau_r);
Serial.print(", w_Tau_l: ");
Serial.print(w_Tau_l);
Serial.print(", w_Tau_r: ");
Serial.println(w_Tau_r);

// drop used encoder measurements
duration_LEFT = 0;
duration_RIGHT = 0;

analogWrite(leftSpeed, abs(w_Tau_l) + leftMotor_adj); // just to make sure they are positive
analogWrite(rightSpeed, abs(w_Tau_r) + rightMotor_adj);

// get current time mark
long cur_time = millis() - start_time; // timestamp
// Write logs to the SD card
String dataString = String(Tang) + ", " + String(ACSEL) + ", " + String(cur_time) + ", " + String(Velocity)
+ ", " + String(w_Tau_l) + ", " + String(w_Tau_r) + ", " + String(Kyrs);
File logFile = SD.open("Data.txt", FILE_WRITE);
if (logFile)
{
    logFile.println(dataString);
    logFile.close();
    Serial.println(dataString);
}

// main cycle delay
delay(cycle_time);
}

void wheelSpeed_LEFT()
{
    int Lstate_LEFT = digitalRead(encoder0pinA_LEFT);
    if ((encoder0PinALast_LEFT == LOW) && Lstate_LEFT == HIGH)
    {
        int val = digitalRead(encoder0pinB_LEFT);
        if (val == LOW && Direction_LEFT)
        {
            Direction_LEFT = false; //Reverse
        }
        else if (val == HIGH && !Direction_LEFT)
        {
            Direction_LEFT = true; //Forward
        }
    }
    else if ((encoder0PinALast_LEFT == HIGH) && Lstate_LEFT == LOW)
    {
        int val = digitalRead(encoder0pinB_LEFT);
        if (val == HIGH && Direction_LEFT)
        {
            Direction_LEFT = false; //Reverse
        }
        else if (val == LOW && !Direction_LEFT)
        {

```

```

    Direction_LEFT = true; //Forward
  }
}
encoder0PinALast_LEFT = Lstate_LEFT;
if (!Direction_LEFT) duration_LEFT++;
else duration_LEFT--;

}

void wheelSpeed_RIGHT()
{
  int Lstate_RIGHT = digitalRead(encoder0pinA_RIGHT);
  if ((encoder0PinALast_RIGHT == LOW) && Lstate_RIGHT == HIGH)
  {
    int val = digitalRead(encoder0pinB_RIGHT);
    if (val == LOW && !Direction_RIGHT)
    {
      Direction_RIGHT = true; //Reverse
    }
    else if (val == HIGH && Direction_RIGHT)
    {
      Direction_RIGHT = false; //Forward
    }
  }
  else if ((encoder0PinALast_RIGHT == HIGH) && Lstate_RIGHT == LOW)
  {
    int val = digitalRead(encoder0pinB_RIGHT);
    if (val == HIGH && !Direction_RIGHT)
    {
      Direction_RIGHT = true; //Reverse
    }
    else if (val == LOW && Direction_RIGHT)
    {
      Direction_RIGHT = false; //Forward
    }
  }
  encoder0PinALast_RIGHT = Lstate_RIGHT;
  if (!Direction_RIGHT) duration_RIGHT++;
  else duration_RIGHT--;

}

```