



Neural network control of a wheeled mobile robot based on optimal trajectories

Pavol Bozek¹ , Yury L Karavaev^{2,3} , Andrey A Ardentov⁴ and Kirill S Yefremov³

Abstract

This article is concerned with developing an intelligent system for the control of a wheeled robot. An algorithm for training an artificial neural network for path planning is proposed. The trajectory ensures steering optimal motion from the current position of the mobile robot to a prescribed position taking its orientation into account. The proposed control system consists of two artificial neural networks. One of them serves to specify the position and the size of the obstacle, and the other forms a continuous trajectory to reach it, taking into account the information received, the coordinates, and the orientation at the point of destination. The neural network is trained on the basis of samples obtained by modeling the equations of motion of the wheeled robot which ensure its motion along trajectories in the form of Euler's elastica.

Keywords

Wheeled mobile robots, robust, adaptive and optimal control, collision avoidance and multi-vehicle systems, path planning and navigation, robot learning

Date received: 24 December 2019; accepted: 25 February 2020

Topic: Mobile Robots and Multi-Robot Systems

Topic Editor: Yangquan Chen

Associate Editor: Martin Grossman

Introduction

At present, the optimal path planning for mobile robots is based, as a rule, on the analysis of the kinematics or dynamics of their motion. As optimality criteria, one can use various parameters, for example, maximal speed, minimal path length, minimal or maximal acceleration, the requirement to ensure the specified orientation, and their combinations.^{1–4} However, due to the specificity of applied problems, the results of theoretical analysis do not always apply in practice. For example, the environment in which the robot functions can change as it encounters an obstacle, another robot, or a human in its path. This requires the control system to respond instantaneously, that is, to adjust its trajectory, but, at the same time, to keep performing the task assigned to it earlier. Artificial neural networks (ANNs) are coming increasingly into use to solve such problems, namely, both the problems of navigating robots

and those of recognizing and manipulating individual classes of objects.^{5–9}

The use of neural networks for the control of robots has a long history. Originally, they were only part of the control system and performed a particular function, for example, adjustment of control coefficients, the choice of the

¹Slovak University of Technology in Bratislava, Slovakia

²Center for Technologies in Robotics and Mechatronics Components, Innopolis University, Innopolis, Russia

³Kalashnikov Izhevsk State Technical University, Izhevsk, Russia

⁴Ailamazyan Program Systems Institute of RAS, Pereslavl-Zalessky, Russia

Corresponding author:

Yury L Karavaev, Center for Technologies in Robotics and Mechatronics Components, Innopolis University, ul. Universitetskaya 1, Innopolis, 420500, Russia.

Email: karavaev_yury@istu.ru



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

direction of motion, speed correction, identification of information from sensors, and so on. For these operations, one can use the simplest architectures of ANNs, which have proved to be efficient.

In the study by Gigliotta and Nolfi,¹⁰ using a simple recurrent network, robots learn to structure the space around them by choosing the direction of motion. At the input, the neural network receives information from illumination level sensors, and at the output, it receives commands for controlling the wheels' motors. In the study by Velagic et al.,¹¹ a recurrent neural network with feedback is applied to control a mobile robot with two driving wheels and a ball castor. The dynamical neural network contains one hidden layer consisting of 10 neurons with a tan-sigmoid activation function and one neuron at the output with a linear activation function. The error of the robot's position relative to the black strip is fed to the input of the network, and at the output, the network linear velocities of the left and right wheels are formed. Training is performed by the method of optimized back propagation of error. The speed of training changes adaptively. Numerical experiments in MATLAB have been conducted. In the study by Fierro and Lewis,¹² a control system integrating the kinematic model of a robot with a neural network torque controller has been developed to control a mobile robot with a similar design. The neural network is a perceptron with 10 neurons in a hidden layer and a sigmoid activation function. The system parameters that are not defined precisely (the velocities of the wheels from the robot's model, mismatch between the position and the orientation of the robot, and its kinetic energy) are fed to the input of the neural network. The parameters defining the dynamics and the torques of the robot (the inertia matrix, the matrix of centripetal and Coriolis accelerations, and the surface friction matrix) are fed to the output of the network. The neural network is trained in the process of operation and has a control unit with limiting parameters for protection against overshoot. In the study by Boukens and Boukabou,¹³ use is made of a control circuit which is similar to that presented by Fierro and Lewis¹² and is supplemented with units to monitor control boundaries and keep track of the weights of the neural network. Computational experiments are presented for three types of mobile robots: a mobile wheeled robot with two driving wheels and one ball castor, a mobile wheeled robot with two driving wheels and one ball castor with a manipulator, and a balancing mobile wheeled robot with two driving wheels. Giusti et al.⁵ have proposed an algorithm for selecting a maneuver for a quadcopter (turns to the left, to the right, and forward motion) under conditions of paths through the woods using information from a video camera on the basis of a deep high-accuracy neural network. The training of the neural network was performed using video files during hikes. Natural experiments were conducted using air drones under forest conditions. In the study by Levine et al.,⁷ the control of a manipulator robot with known kinematics was proposed on the basis of a

high-accuracy deep neural network. In this article, the neural network is taught to grip objects by means of a manipulator, using information from one camera. To train the neural network, 6–14 identical manipulator robots were used for 2 months at different times.

With advances in computer engineering, for control systems based on ANNs is not important correspond to the dynamics of controlled systems. Modern artificial intelligent technologies, such as deep learning and reinforcement learning, make it possible to completely exclude the investigation of the dynamics or kinematics of motion without taking into consideration the specific features of interaction of individual elements of the system with each other. However, in this case, it takes plenty of time to train the neural network, and a technically complicated system for self-training of the control system is required.

The merits of the application of neural networks as robot control devices include high loyalty to the dynamics and kinematics of the model and quick task execution with a trained network. The key disadvantages are ambiguity in the choice of network architecture, long duration and a large number of teaching samples, and problems of transferability of training results from one robot to another.

This article is concerned with developing a neural network control system for a mobile wheeled robot. In our case, the proposed neural network control system forms an optimal trajectory of motion of a mobile wheeled robot taking into account both its position and its orientation at the end point of motion. In addition, if the robot encounters an obstacle in its path, the control system ensures that a new optimal trajectory is formed to maneuver around the obstacle.

Formulation of the problem

Consider the problem of the motion of a mobile wheeled robot which has an independent drive for each wheel, and a ball castor. In the general case, the motion of the mobile robot from the initial position specified in two-dimensional space by coordinates $q_0 = (x_0, y_0)$ and orientation θ_0 to a point with given coordinates $q_1 = (x_1, y_1)$ and orientation θ_1 can be implemented in two ways with $\theta_0 \neq \theta_1$. In the first case, the motion of the robot can be divided into two main maneuvers: spot turn operation (at the initial point and/or the end point) and motion along a straight line (trajectory 1 in Figure 1(a)). In this case, even during motion with the maximal velocity, additional time for turns is required. It should also be noted that any other modification of a mobile wheeled robot, for example, a mobile robot with an automobile chassis layout, or a multilink wheeled robot (having trailers with passive wheels) and other robots, will not allow the spot turn operation to be performed. In the second case, the robot can move along some curve connecting the initial and final positions, and the velocity of motion along this curve can be continuous,

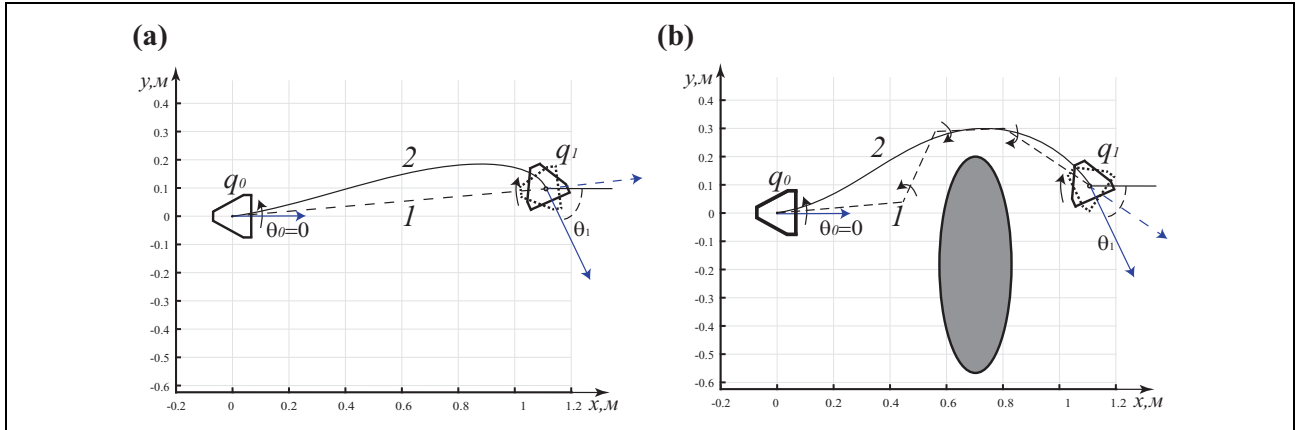


Figure 1. Various trajectories of the robot. The numbers denote trajectories: (1) Euler's elastica with minimal curvature and (2) rectilinear trajectory with additional turns of the mobile robot at nodal and terminal points: (a) without an obstacle and (b) with an obstacle in the path of the mobile robot.

regardless of the orientation at the end point, that is, the robot can move without stops.

If the mobile robot encounters an obstacle in its path (see Figure 1(b)), its trajectory will become still more complex: a greater number of stops and additional turns will be required. When planning the path using continuous trajectories in the velocity, one can easily correct the trajectory of motion while ensuring the specified position and orientation at the end point (in this case, the trajectory is shown as a dashed line in Figure 1(b)). This solution is unique if the mobile robot has several links—passive trailers.

As continuous trajectories of motion, one can use approximations in the form of sinusoids,^{3,4} polynomials,^{14–16} splines,^{17,18} and trajectories in the form of Euler's elastica.^{19,20} However, all the abovementioned curves used as trajectories require rather lengthy calculations of their parameters. In our opinion, Euler's elastica, which also ensures minimization of the control given by the difference of the angular velocity of rotation of the wheels,²¹ holds the greatest promise. Trajectories in the form of Euler's elastica can also be applied for multilink mobile robots.

The path planning algorithm which relates the initial and end points with the specified orientation of the mobile robot at the end point using Euler's elastica is discussed in the study by Ardentov et al.²⁰ It is noted that the process of calculating the parameters of the elastica involves highly intensive computations, which can result in high costs for the control system of the mobile robot.

The purpose of this work is to search for an alternative solution to the complicated problem of calculating the parameters of Euler's elastica. We suggest using ANNs as such a solution. Namely, using the data received from the sensors on the current position and velocity of the mobile robot and the point of destination specified by the coordinates $q_1 = (x_1, y_1)$ and the orientation θ_1 , the ANN forms a

trajectory of motion in the form of Euler's elastica, from which control actions are calculated later. Moreover, if the mobile robot encounters an obstacle in its path, the control system must adjust the trajectory of motion depending on the size and position of the obstacle.

In this article, we propose a structure of the control system of the mobile robot. This structure is based on two ANNs, which are trained separately. One of them determines the data from the sensors, the position, and the size of the obstacle; and the other forms the parameters of the optimal trajectory of motion using these data and information on the point of destination and on the current position of the mobile robot. We also consider special features of the practical realization of the mobile robot, its control system, as well as algorithms for training neural networks.

The control of the mobile robot based on an ANN

Let us consider a modification of the control system where ANNs are applied for a prototype of the mobile wheeled robot with two driving wheels and one ball castor. A kinematic scheme and a three-dimensional model of the robot are shown in Figure 2. The driving wheels have a diameter of $r_0 = 5$ cm and are spaced $l_0 = 24$ cm from each other. In addition, their axes of rotation coincide, and the ball castor is equidistant from them.

The electric motors and the incremental encoders are located on the same axis as the wheels. This design makes it possible to define and estimate the angular velocity of one driving wheel independently of each other.

To control the mobile wheeled robot along optimal trajectories connecting its initial and final positions, taking into account the obstacles encountered in its path, a control system based on ANNs has been developed. The general structural scheme of the control system of the mobile wheeled robot is presented in Figure 3.

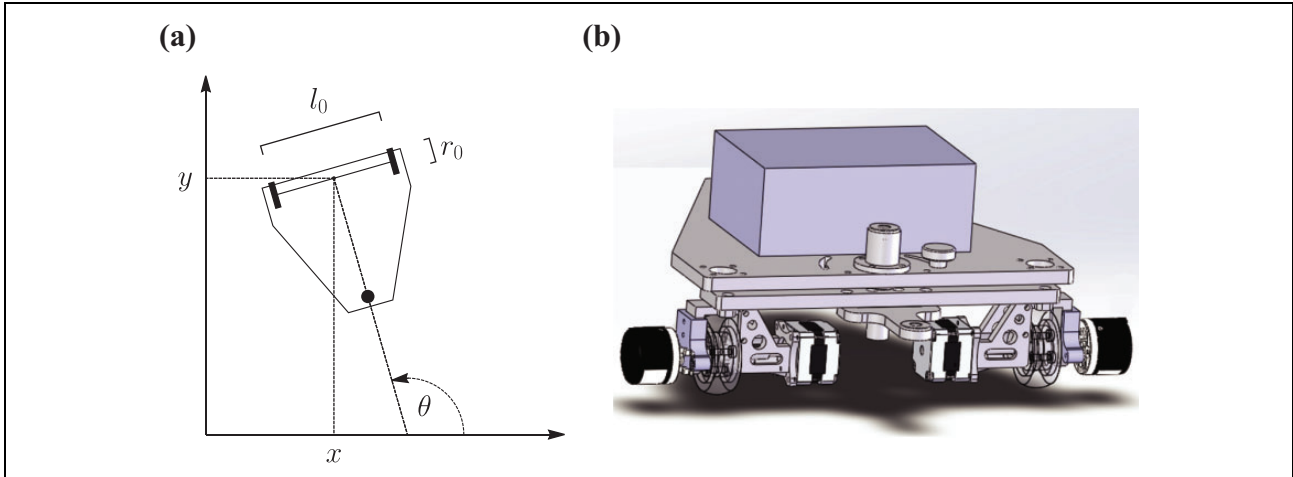


Figure 2. A prototype of the mobile wheeled robot: (a) a kinematic scheme and (b) a three-dimensional model.

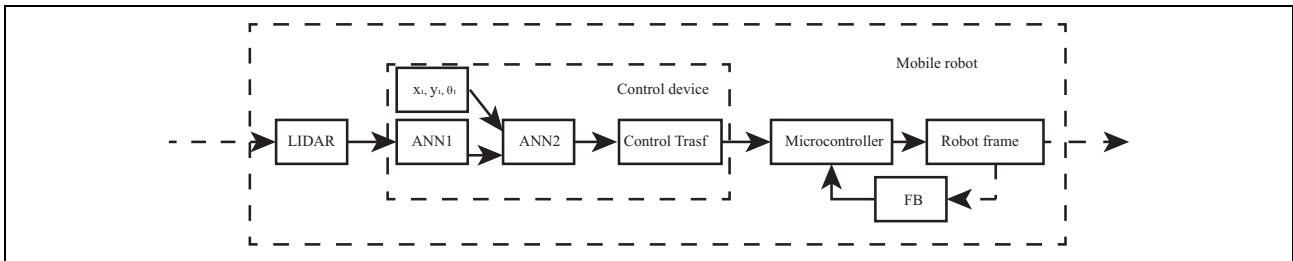


Figure 3. A generalized structural scheme of the control system.

The control system includes two ANNs and a unit for formation of commands for the onboard microcontroller of the robot (Control Transf). The system has been divided into two neural networks for ease of training and for their separate use in the future. Technically, the control system unit can be realized on a stationary personal computer or in the cloud to accelerate the process of training the network using a large amount of simulated data. In this case, the transfer of data on the environment and control actions to the microcontroller of the onboard control system is performed via a wireless communication channel.

The control system of the mobile robot receives data on the environment from the laser scanner (Lidar). These data are information on the distance to the surrounding objects or on their absence in the path of the robot. The structure of ANN1 is shown in Figure 4. The data coming from the Lidar sensor are a vector consisting of 360 values, which correspond to the distances to the surrounding objects around the robot with a discreteness of 1° . The data refresh rate depends on the operation speed of the laser scanner. The experiments were conducted for a laboratory prototype with the laser scanner RPLidar A1, which has a refresh rate of 5 Hz.

The ANN1 has three layers. The number of neurons in the first layer is 360, which corresponds to the dimension of the input vector containing information

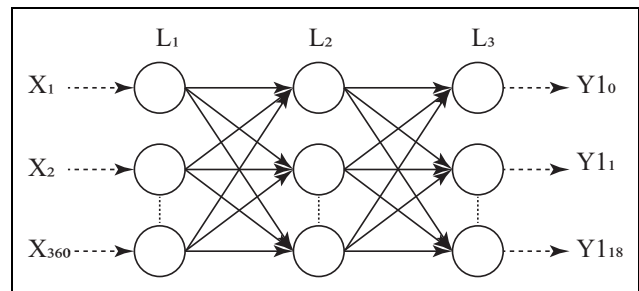


Figure 4. Structure of ANN1. ANN: artificial neural network.

on distances to the surrounding objects. The second is an intermediate layer which also contains 360 neurons, and the third is an output layer containing 18 neurons. The output neurons encode the coordinates of the obstacle $\{x_b, y_b\}$ and its radius R_b , respectively. Figure 5 shows a graphical representation of the data coming to the input of ANN1, where ϕ is the angle at which the obstacle is located relative to the mobile robot. This picture corresponds to the situation where only one object is located at distance d_0 from the robot. In the section in which the measurements are done, this object has the form of a circle with radius R_b . The maximal detection zone b of RPLidar A1 is 6 m, and the distances at other points are equal to infinity or to distances to the walls of the room.

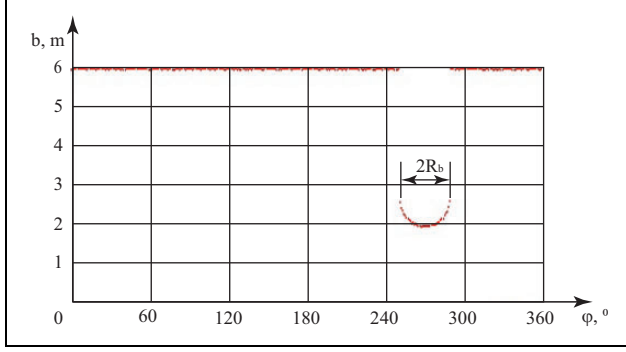


Figure 5. Graphical representation of the data obtained from the Lidar sensor.

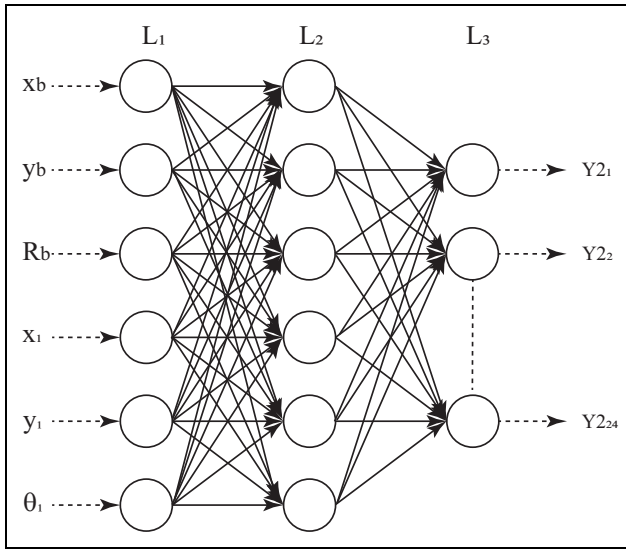


Figure 6. Structure of ANN2. ANN: artificial neural network.

For objects having more complicated form, ANN1 also formed three parameters: the radius of the circumscribed circle and the coordinates of its center. These values suffice to form a trajectory required for the robot to take a detour around the obstacle.

The second ANN2 (see Figure 6) also has three layers. The first layer of ANN2 has 6 neurons, the second also 6, and the output layer has 24 neurons, which encode the parameters of the elastica, k_0, l, p_e, L . The values of the coordinates of the point of destination $\{x_1, y_1\}$ and the orientation θ_1 at this point, as well as the coordinates $\{x_b, y_b\}$ and the radius of the circle, R_b , which circumscribe the obstacle in the path of the mobile robot, are fed to the input of ANN2. A detailed analysis of the values of these parameters is made in the study by Ardentov et al.²⁰ In what follows, we present the main calculations necessary for understanding the process of training and using the parameters obtained to control the robot.

Formation of optimal trajectories

The kinematics of the motion of the mobile wheeled robot with a differential drive and a ball castor is described by the following system of differential equations:

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t) \\ \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = w(t) \end{cases} \quad (1)$$

where t is the time parameter, $w(t)$ is the angular velocity of the robot, and $v(t)$ is its linear velocity.

The solution to the system of equation (1) can be found in the class of optimal controls of continuous functions, called Euler's elastica,²¹ which are described by the following equations

$$\begin{cases} x'(s) = \cos \theta(s) \\ y'(s) = \sin \theta(s) \\ \theta'(s) = u_E(s) \end{cases} \quad (2)$$

where s is the length of the trajectory, symbol “'” denotes the derivative with respect to s , and $u_E(s)$ defines the curvature of the trajectory. Under the conditions of our problem the, curvature of the trajectory is given by the angular velocity of the mobile robot, $w \equiv u_E$.

Euler's elastica ensures local minimization of the squared curvature (control)²¹

$$\int u_E^2(s) ds \rightarrow \min \quad (3)$$

It is convenient to realize the relation between the angular velocity and the curvature of the trajectory by introducing the parameter defining smooth motion along the trajectory

$$w(t) = v(t)u_E(s(t)) \quad (4)$$

Then, it is convenient to divide the trajectory of motion into three segments: acceleration from rest, motion with maximal velocity, and smooth stop

$$v(t) = \begin{cases} v_b + (v_m - v_b) \sin \frac{\pi t}{2T_b}, & t \in [0, T_b] \\ v_m, & t \in [T_b, T_b + T_m] \\ v_e + (v_m - v_e) \cos \frac{\pi(t - T_b - T_m)}{2T_e}, & t \in [T - T_e, T] \end{cases} \quad (5)$$

where v_b , v_m , and v_e are the velocities of motion on the corresponding segments, and T_b , T_m , and T_e are the time intervals during which acceleration, motion with maximal velocity, and stop occur, respectively. For the length of the segments, the equation $L_b + L_m + L_e = L$ is satisfied, where L is the length of one period of the trajectory.

A graphical representation of changes in the velocity during motion along Euler's elastica is shown in Figure 7 for $t \in [0, T]$ for continuous acceleration.

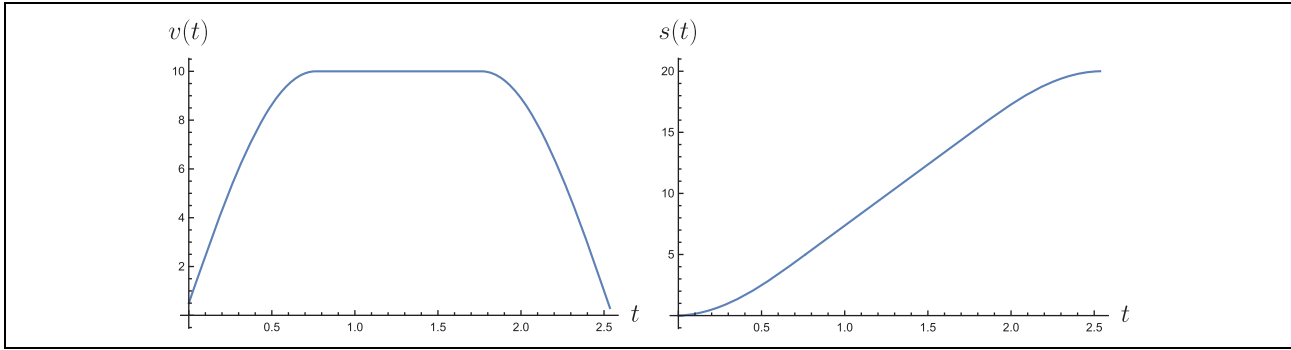


Figure 7. An example of the functions $v(t)$ and $s(t)$ for $L = 20$, $v_b = 5/10$, $v_e = 3/10$, $v_m = 10$, $L_b = L/4$, $L_m = L/2$, $L_e = L/4$ for continuous acceleration.

$$s(t) = \begin{cases} v_b t - \frac{2(v_m - v_b)T_b}{\pi} \left(\cos \frac{\pi t}{2T_b} - 1 \right), & t \in [0, T_b] \\ L_b + v_m(t - T_b), & t \in [T_b, T_b + T_m] \\ (L_b + L_m) + v_e(t - T_b - T_m) + \frac{2(v_m - v_e)T_e}{\pi} \sin \frac{\pi(t - T_b - T_m)}{2T_e}, & t \in [T - T_e, T] \end{cases} \quad (6)$$

The lengths of the corresponding time intervals are

$$T_b = \frac{\pi L_b}{(\pi - 2)v_b + 2v_m}; T_m = \frac{L_m}{v_m}; T_e = \frac{\pi L_e}{(\pi - 2)v_e + 2v_m} \quad (7)$$

Using the reparameterization (4)–(6), we obtain

$$\begin{cases} \dot{x}(s) = v(s) \cos \theta(s) \\ \dot{y}(s) = v(s) \sin \theta(s) \\ \dot{\theta}(s) = w(s) \end{cases} \quad (8)$$

The curvature of the elastica is defined in terms of elliptic Jacobi functions (cn, dn) and a complete elliptic integral of the first kind K . Depending on the solution of equation (8), one distinguishes two types of elastica corresponding to different types of trajectories and controls. The first type includes inflexional elastica, which are described by the following dependence

$$w(s) = \frac{8kK}{l} \operatorname{cn} \left(4K \left(p_E + \frac{s}{l} \right); k \right) \quad (9)$$

where $k \in (0, 1)$ is the main parameter defining the form of the elastica, $p_E \in [0, 1]$ is the parameter defining the initial point of the elastica (by virtue of the periodicity of the elastica, the values of $p_E + n$, $n \in \mathbb{Z}$, are equivalent).

The second type includes noninflexional elastica, which are calculated in accordance with the expression

$$w(s) = \pm \frac{4K}{l} \operatorname{dn} \left(2K \left(p_E + \frac{s}{l} \right); k \right) \quad (10)$$

and differ from the inflexional elastica in that they have no inflexion points. Examples of inflexional and noninflexional elastica are given in Figure 8.

The above analysis allows the conclusion that the key parameters defining the trajectory of motion in the form of Euler's elastica are the parameters k_0, l, p_e, L , which can be calculated in solving the boundary-value problem for two points on the plane

$$x(0) = x_0, y(0) = y_0, \theta(0) = \theta_0 \quad (11)$$

$$x(L) = x_e, y(L) = y_e, \theta(L) = \theta_e \quad (12)$$

with a fixed value of L for the system (8) with a global minimization of the functional

$$\int_0^L (\alpha u_E^2(s) + \beta) ds \rightarrow \min \quad (13)$$

where $\alpha > 0$ and $\beta > 0$ are the constants defining the weights for a compromise between the length of the trajectory and the energy spent on steering control actions. For the investigations, we set $\alpha = \beta = 1$.

Formation of controls for implementation of the motion of the mobile robot along optimal trajectories

We describe an algorithm for forming control actions to implement motion from an arbitrary initial position (x_0, y_0, θ_0) to the final position (x_1, y_1, θ_1) along the elastica.

1. Denote the distance on the plane between the initial and final positions by

$$d_{01} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (14)$$

from which we calculate the parameters for an optimal elastica of the length

$$L = k_e d_{01} \quad (15)$$

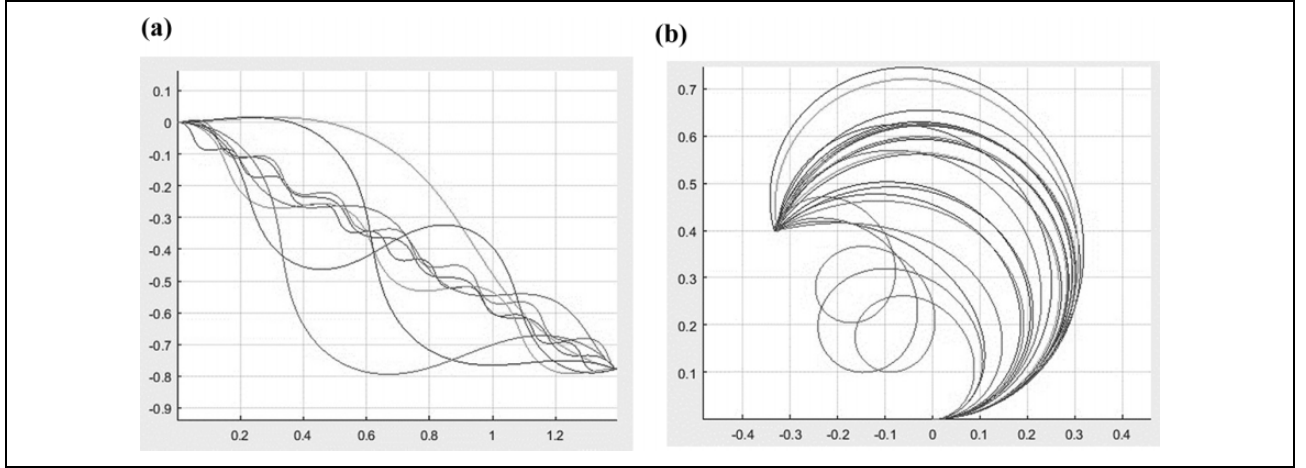


Figure 8. Examples of Euler's elasticas: (a) inflexional elasticas and (b) noninflexional elasticas.

namely, the values of l, k, p_E , as well as the type of the elastica (inflexional and noninflexional), in accordance with the algorithm presented in the study by Ardentov and Sachkov.²² Here, k_e is the coefficient defining the maximal curvature of the elastica for which the calculations are performed. For modeling purposes, this coefficient was taken to be $k_e = 2$, although later, depending on the size of the object (obstacle) for which a detour trajectory is to be formed, it will be this object that will be able to define the value of this coefficient.

2. Construct elasticas of equal or smaller length for the resulting value of the maximal length of the elastica, L ; calculate the values of the integral (13) from the parameter values of these elasticas. The parameters ensuring minimization of the integral are used later to calculate the angular velocities.
3. Calculate the linear and angular velocities $v(t)$ and $w(t)$ for the given law of acceleration and deceleration in equations (4) and (5).
4. Calculate, from the values obtained for $v(t)$ and $w(t)$, the angular velocities of the wheels, w_r and w_l , in accordance with the following kinematic expressions

$$w_l(t) = \frac{v(t) - l_0 w(t)/2}{r_l}, \quad w_r(t) = \frac{v(t) + l_0 w(t)/2}{r_r} \quad (16)$$

where $w_l(t)$ and $w_r(t)$ are the angular velocities and r_l and r_r are the radius of the left and right wheels of the mobile robot, respectively.

Using relation (4), one can rewrite the integral (13) in terms of the linear and angular velocities of the robot

$$\int_0^T \frac{\alpha w^2(t) + \beta v^2(t)}{v(t)} dt \rightarrow \min \quad (17)$$

We note that, when $\alpha = l_0^2/4, \beta = 1$, the integral that is minimized can be rewritten in terms of the angular velocities of the wheels w_r, w_l as follows

$$\int_0^T \frac{r_l^2 w_l^2(t) + r_r^2 w_r^2(t)}{r_l w_l(t) + r_r w_r(t)} dt \rightarrow \min \quad (18)$$

5. The resulting values of the angular velocities are transmitted to the control system, which regulates them by means of feedback sensors.

This algorithm was used to model and form teaching sets for training the ANN2.

Training of a neural network control system

The implementation of the ANNs, their training, and testing were carried out using the framework “TensorFlow.”²³ For the training of the networks, we used the method of gradient descent with back propagation of error. To prepare teaching samples for ANN1, we developed a program that simulates the environment, that is, randomly generates objects—obstacles having different sizes and different locations. To reduce the training time, we considered only cases where the obstacles were in the following intervals: the position of the obstacle $x_0, y_0 \in [0.1, 1.0]$ and the radius of the circle in the form of which the obstacle was presented $R_b \in [0.1, 0.5]$.

To train ANN2, the teaching samples were formed for different coordinates of the point of destination x, y and for different orientations θ of the mobile robot at the end point. For modeling an obstacle that the mobile robot encounters in its path, the curvature of the trajectory was increased with the current and final positions and orientations of the mobile robot kept fixed. The radius of the curvature of the trajectory, ρ , was defined by the radius of the obstacle which the mobile robot was able to take a detour around along the trajectory

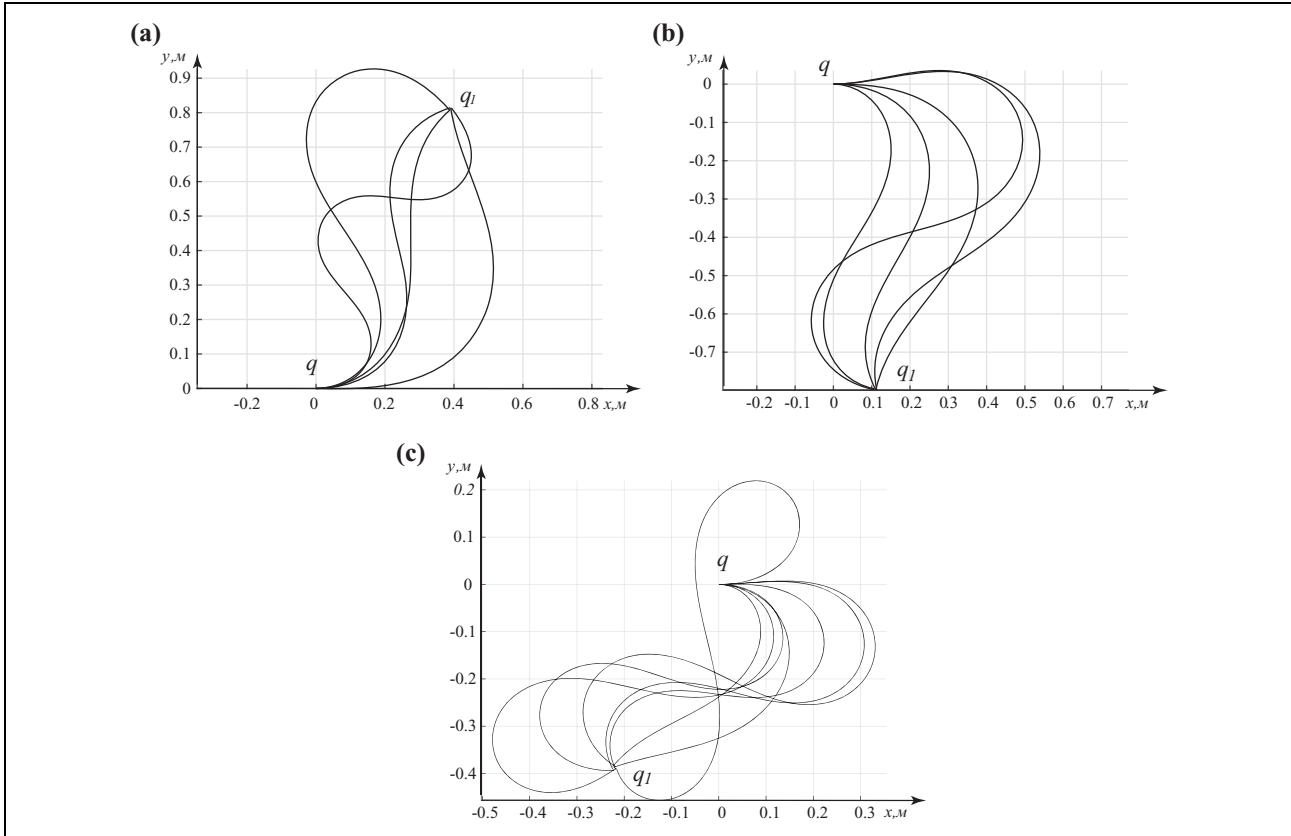


Figure 9. Examples of inflexional Euler elastica. Coordinates of the end points: (a) $q_i[0.39; 0.81]$. (b) $q_i[0.11; -0.8]$. (c) $q_i[-0.22; -0.39]$.

of this curvature, $R_o = \rho - \delta$, where δ is the distance necessary for the mobile robot to move safely along the trajectory near the object. In the experiments, this distance was taken to be $\delta = 0.05$ m. The teaching sets were formed for values of the orientation of the mobile robot at the end point, $\theta = n\pi/4$, where $n = 0, 1, 2, \dots, 8$. A total of 1000 samples were prepared to train ANN2. Examples of trajectories used for training are given in Figure 9.

Experiments

After the training of ANN1 and ANN2, two natural experiments were conducted in which the mobile robot moved to the point with coordinates (1.3; -0.5 m) and the orientation angle 0.13 rad. One of the experiments was carried out with an obstacle in the path of the mobile robot (see Figure 10) and without an obstacle (see Figure 11).

Experimental trajectories were fixed by cameras of a motion capture system Vicon. A cylindrical object with diameter 9 cm and coordinates (0.41; -0.28) was used as an obstacle. The resulting trajectories showed that at the end points, the difference of the trajectories obtained

in the natural experiments from the theoretical trajectories does not exceed 1 cm with a trajectory length of 1.5 m. This error is due to random errors such as the accuracy of the position of the mobile robot at the start, surface irregularities, and inaccuracies in determining the parameters of the mobile robot. On the whole, the algorithm has shown its efficiency.

Conclusions

The results of the natural experiments confirm the efficiency of the proposed control algorithm for the mobile robot to move to the point with a given position and a given orientation. The training of the mobile wheeled robot to move along predetermined trajectories in the form of Euler's elastica will ensure an optimal control and reduce the training time as compared to the training based only on experimental data. For the future, it is planned to increase the base of teaching samples, teach ANN to use a larger number of trajectories, and check the performance capability of its control. It is also planned to extend the existing functional of ANN to include the possibility of taking detours around dynamical obstacles and to perform reinforcement learning.

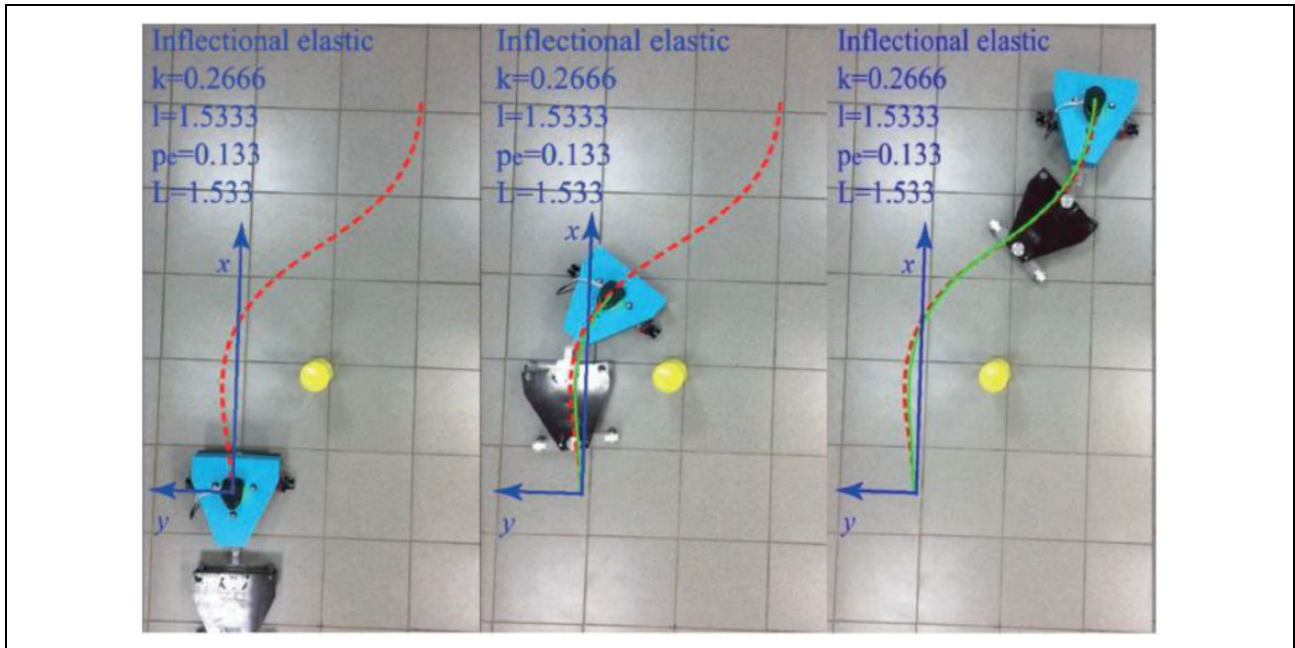


Figure 10. Experimental and theoretical trajectories of the mobile wheeled robot with an obstacle in its path. The dashed line indicates the theoretical trajectory, and the solid line corresponds to the experimental trajectory obtained during control using the neural network.

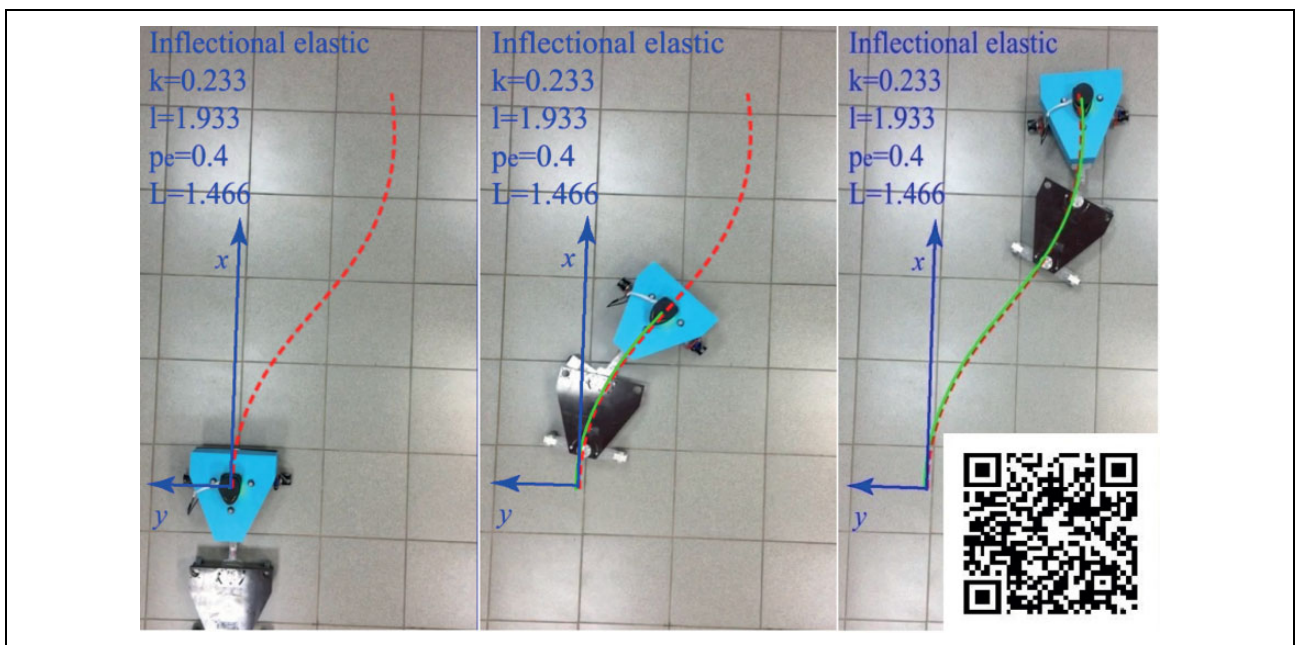


Figure 11. Experimental and theoretical trajectories of the mobile wheeled robot without an obstacle in its path. The dashed line indicates the theoretical trajectory, and the solid line corresponds to the experimental trajectory obtained during control using the neural network.

Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.


Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The work of Pavol Bozek and Yuriy L Karavaev (sections 1, 3, 5, and

7) was funded by research project VEGA 1/0019/20 and by the RFBR grant 18-38-00454 mol_a, and is sponsored by the project 015STU-4/2018. The reported study of Andrey A Ardentov (Formation of optimal trajectories) was funded by RFBR, project number 19-31-51023 and was carried out at the Sirius University of Science and Technology (Sochi, Russia) and Ailamazyan Program Systems Institute of RAS (Pereslavl-Zalessky, Russia), and the work of Kirill S Yefremov (sections 2 and 6) was carried out within the framework of the project of the State Assignment at the Kalashnikov Izhevsk State Technical University under grant no 1.2405.2017/4.6.

ORCID iD

Pavol Bozek  <https://orcid.org/0000-0002-3891-3847>

Yury L Karavaev  <https://orcid.org/0000-0002-6679-1293>

References

1. Stentz A. Optimal and efficient path planning for partially-known environments. In *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, May 1994, Vol. 4, pp. 3310–3317. IEEE.
2. Bobrow JE. Optimal robot plant planning using the minimum-time criterion. *IEEE T Robotic Autom* 1988; 4(4): 443–450.
3. Murray RM and Sastry SS. Nonholonomic motion planning: steering using sinusoids. *IEEE T Automat Contr* 1993; 38(5): 700–716.
4. Laumond JP, Sekhavat S, and Lamiroux F. Guidelines in nonholonomic motion planning for mobile robots. In: Laumond JP (ed) *Robot motion planning and control*, 1998, pp. 1–53. Berlin: Springer-Verlag.
5. Giusti A, Guzzi J, Ciresan DC, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Rob Autom Lett* 2015; 1(2): 661–667.
6. Panigrahi PK, Ghosh S, and Parhi DR. Navigation of autonomous mobile robot using different activation functions of wavelet neural network. *Arch Contr Sci* 2015; 25(1): 21–34.
7. Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int J Robot Res* 2018; 37(4-5): 421–436.
8. Mahler J, Liang J, Niyaz S, et al. Dex-net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*. 2017.
9. Zidek K, Maxim V, Pitel J, et al. Embedded vision equipment of industrial robot for inline detection of product errors by clustering-classification algorithms. *Int J Adv Robot Syst* 2016; 13(5): 1729881416664901
10. Gigliotta O and Nolfi S. Formation of spatial representations in evolving autonomous robots. In: *2007 IEEE symposium on artificial life, 2007. ALIFE '07*. IEEE Symposium, pp. 171–178.
11. Velagic J, Osmic N, and Lacevic B. Neural network controller for mobile robot motion control. *World Academy Sci Eng Technol* 2008; 47: 193–198.
12. Fierro R and Lewis FL. Control of a nonholonomic mobile robot using neural networks. *IEEE T Neural Networ* 1998; 9(4): 589–600.
13. Boukens M and Boukabou A. Design of an intelligent optimal neural network-based tracking controller for nonholonomic mobile robot systems. *Neurocomputing* 2017; 226: 46–57.
14. Bushnell L, Tilbury D, and Sastry S. Steering three-input nonholonomic systems: the fire-truck example. *Int J Robot Res* 1995; 14(4): 366–381.
15. Boissonnat JD and Lazard S. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacle. *Int J Comput Geom Ap* 2003; 13(03): 189–229.
16. Tilbury D, Murray R, and Sastry S. Trajectory generation for the n-trailer problem using Goursat normal form. *IEEE T Automat Contr* 1995; 40(5): 802–819.
17. Lau B, Sprunk C, and Burgard W. Kinodynamic motion planning for mobile robots using splines. In: *2009 IEEE/RSJ international conference IROS*, October 2009, pp. 2427–2433.
18. Ghilardelli F, Lini G, and Piazzi A. Path generation using η^4 -splines for a truck and trailer vehicle. *IEEE T Autom Sci Eng* 2014; 11(1): 187–203.
19. Ardentov AA. Multiple solutions in Euler's elastic problem. *Automat Rem Contr* 2018; 79(7): 1191–1206.
20. Ardentov AA, Karavaev YL, and Yefremov KS. Euler elasticas for optimal control of the motion of mobile wheeled robots: the problem of experimental realization. *Regul Chaotic Dyn* 2019; 24(3): 312–328.
21. Sachkov YL. Maxwell strata in the Euler elastic problem. *J Dyn Control Syst* 2008; 14(2): 169–234.
22. Ardentov AA and Sachkov YL. Solution to Euler's elastic problem. *Automat Rem Contr* 2009; 70(4): 633–643.
23. Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning. In: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. Savannah, Georgia, USA, 2016.