

Article

Not peer-reviewed version

A Genetic Algorithm Incorporating Knowledge and Natural Selection Theory in Adaptive Control of a Wheeled Robot

[Paweł Penar](#) and [Piotr Gierlak](#) *

Posted Date: 6 May 2025

doi: 10.20944/preprints202504.2589.v1

Keywords: adaptive control; evolutionary algorithms; genetic algorithms; wheeled robots



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Genetic Algorithm Incorporating Knowledge and Natural Selection Theory in Adaptive Control of a Wheeled Robot

Paweł Penar  and Piotr Gierlak * 

Department of Applied Mechanics and Robotics, Rzeszów University of Technology

* Correspondence: pgierlak@prz.edu.pl

Abstract: The control of a mobile wheeled robot under varying operating conditions can be based on classical adaptive control. Then a genetic algorithm, which is one implementation of evolutionary optimisation algorithms, can be used in the control structure. In this paper, the classical genetic algorithm is replaced by a genetic algorithm based on the theory of natural selection (GABONST), which is known from the literature. An additional feature of the proposed algorithm is the inclusion of information about the state of the controlled object, i.e. the wheeled robot. This approach provides an input of knowledge to the genetic algorithm. This results in a hybrid type approach that combines a data-driven approach with a physics-based approach. This favours effective control under varying operating conditions. This fact was confirmed by performing numerical tests of the adaptive control of a mobile wheeled robot under varying operating conditions.

Keywords: adaptive control; evolutionary algorithms; genetic algorithms; wheeled robots

1. Introduction

Adaptive control methods are still in the mainstream of research in the control of non-linear objects [1,2]. This is due to the fact that control should take into account variable operating conditions. One approach that takes into account variable operating conditions and compensates for the non-linearity of the control object is feedback linearisation [3]. It is based on a model of the dynamics of the controlled object. Therefore, the quality of control obtained in this way strongly depends on the quality of the model, which is affected by the accuracy of identification of its parameters [4–6]. It is reduced by the disturbances acting on the object, which can be divided into those that result from modelling inaccuracies and those that are the result of varying operating conditions. In the context of mobile wheeled robots, this can be a change in the mass carried or a change in the resistance to motion of wheel-surface pairs. It therefore becomes crucial to introduce a mechanism into the structure of the control algorithm that allows real-time adjustment of the model parameters. This is done by gradient methods, based on the control error or its function [7–9]. This approach integrates adaptive control with optimisation problems [10,11]. Therefore, modern control algorithms make extensive use of artificial intelligence methods, which are based on the search for the minimum function of the model parameters. This opens up the possibility of using models known from machine learning methods, including hybrid algorithms that integrate data-driven approaches with knowledge of the controlled object [12].

In 2023, the IEEE Control Systems Society published a report [13]. In 2023, the IEEE Control Systems Society published a report [1] that outlines how control systems will evolve by 2030. Systems will respond to increasing levels of automation across sectors. The report identifies five key methodologies emerging in control systems. One of these is data-driven modelling and control, which can be understood as the integration of machine learning and control systems. A particular expression of this integration is the combination of data-driven models with physics-based models in control algorithms, e.g. in adaptive control. The aforementioned report and the authors of many papers, e.g.

[14,15] propose the use of hybrid modelling methods in control which are a combination of data-based and physics-based models. The paper [14] provides an overview of hybrid models used in smart manufacturing. There, a physical model was compared with a data-based model and the benefits of hybrid models were pointed out. An analogous review was included in the paper [16], where 81 articles on the application of models integrating knowledge (physics of the controlled object) and data were selected. As before, reference was made to manufacturing issues. The article [16] provides a comprehensive overview of predictive models used in different industrial sectors. It classifies models based on the level of knowledge used to construct them: full knowledge models, zero knowledge models and partial knowledge models. The paper [15] discusses the development and application of hybrid machine learning methods supported by physics models in the context of cyber-physical systems. The authors emphasise that traditional physics-based and data-only modelling approaches have their limitations. Their combination offers new possibilities for mapping complex physical objects.

It was mentioned earlier that adaptive control algorithms include a real-time parameter estimation mechanism. This process can be understood as an optimisation problem, which consists of selecting the values of the parameters of the mathematical model in such a way as to minimise the control error or its functions. Hence, genetic algorithms, which the literature refers to as nature-inspired approximation optimisation algorithms [17] can be used in the parameter estimation problem. Reviews on genetic algorithms. [18–20] and in particular the article [21] state that genetic algorithms belong to the group of metaheuristic algorithms. They are strategies for searching the solution space, typically of an optimisation problem, to obtain an approximate solution.

In this paper, a successful attempt is made to apply a genetic algorithm that is based on the theory of natural selection (GABONST) to the parameter estimation of a mathematical model of a wheeled robot. This algorithm was proposed in the article [22] and is an extension of the classical genetic algorithm by introducing Darwin's idea of natural selection. Moreover, the proposed algorithm has been extended with knowledge-based mechanisms that address, among other things, the value of the tracking error of the wheeled robot. Another type of knowledge used is that the model parameters are interval constant. In this way, a hybrid algorithm was obtained that combines the features of a genetic algorithm and process knowledge.

In summary, there are three areas of contribution of this paper:

- Synthesis of an adaptive control law that uses a hybrid genetic algorithm to estimate the parameters of a mathematical model of a wheeled robot.
- The use of the GABONST algorithm as a hybrid model based on process knowledge.
- Numerical verification of the proposed adaptive control algorithm for a mobile wheeled robot under varying operating conditions.

The rest of the article, in Section 2, presents a description of the kinematics and dynamics of a mobile wheeled robot. Section 3 presents the synthesis of optimal adaptive control of the robot. The control synthesis was carried out using a mathematical model of the control object. Section 4 presents the genetic algorithm used in the control to optimize the model parameters. Selection, crossover and mutation procedures are presented in detail, as well as how to incorporate physical knowledge of the robot into the genetic algorithm procedures. Section 5 presents and discusses the results of a numerical test, confirming the correctness of the developed theoretical solutions. Conclusions on the results of the work are gathered in Section 6.

Translated with DeepL.com (free version).

2. Kinematics and Dynamics of a Wheeled Robot

The mathematical description of robots, in particular wheeled robots, consists of the equations of kinematics and dynamics. The former describe the relationships between the angular velocities of the wheels and the linear velocity of the robot. Their derivation can be based on Denavit-Hartenberg notation, which remains the standard in robotics. The dynamics equations are used to describe and

analyse the motion of the robot under the influence of forces and moments acting on it. Their formalism follows from the Lagrange equations and, in the case of mobile robots, the Lagrange equations with multipliers [23–25].

2.1. Kinematics

Knowledge of the kinematics equations will allow the angles of rotation and angular velocities of the drive wheels of the mobile robot to be determined. These will be used as the desired trajectory in the tracking task.

Figure 1a shows a scheme of a 2-wheeled mobile robot, consisting of: two driving wheels, denoted by 1 and 2, a self-aligning wheel mounted at point D, and a frame [25]. Its CAD model is shown in Figure 1b. The drive wheels with radius r rotate about their own axis, which does not change position relative to the frame. They are driven by separate drives consisting of electric motors and gears.

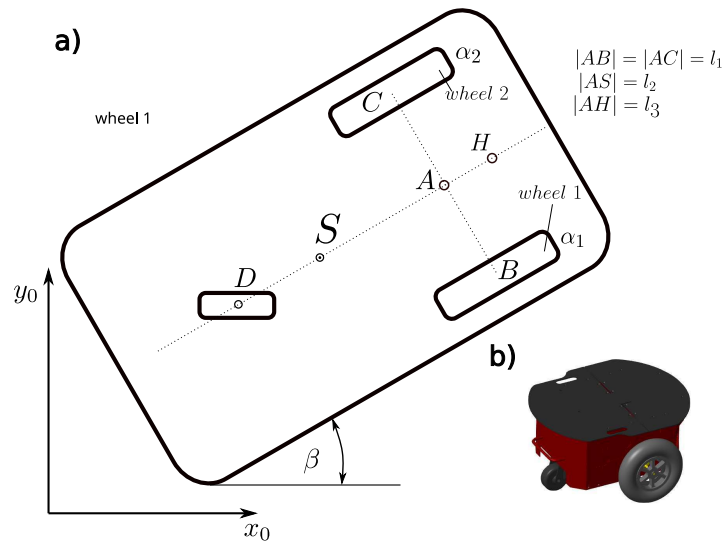


Figure 1. Scheme (a) and CAD model (b) of a mobile wheeled robot

The description of the robot's position and orientation requires three time-varying coordinates, i.e.: $x_H(t), y_H(t)$ - the position of the selected characteristic point of the robot (in this case H) in the reference system x_0y_0 , $\beta(t)$ - the orientation angle of the frame. These coordinates are related by two equations [25]:

$$\begin{cases} \dot{x}_H - \dot{x}_A - l_3 \dot{\beta} \sin \beta = 0 \\ \dot{y}_H - \dot{y}_A - l_3 \dot{\beta} \cos \beta = 0 \end{cases} \quad (1)$$

where v_A is the velocity of point A , l_3 is the distance between point A and point H and $\dot{x}_A \equiv \dot{x}_A(t) \equiv \frac{dx_A}{dt}$, $\dot{y}_A \equiv \dot{y}_A(t) \equiv \frac{dy_A}{dt}$, $\dot{\beta} \equiv \dot{\beta}(t) \equiv \frac{d\beta}{dt}$, $\dot{x}_H \equiv \dot{x}_H(t) \equiv \frac{dx_H}{dt}$, $\dot{y}_H \equiv \dot{y}_H(t) \equiv \frac{dy_H}{dt}$.

The projection of the velocity of point A on the axes of the reference system x_0y_0 is $\dot{x}_A = v_A \cos \beta$ and $\dot{y}_A = v_A \sin \beta$ where v_A is the value of the velocity of point A . Taking this into account, the Equation (1), can be written in the form of

$$\begin{cases} \dot{x}_H - v_A \cos \beta + l_3 \dot{\beta} \sin \beta = 0 \\ \dot{y}_H - v_A \sin \beta - l_3 \dot{\beta} \cos \beta = 0 \end{cases} \quad (2)$$

It is not possible to determine the three coordinates from the two equations, i.e. from (2). Therefore, in order to determine them, a motion path is assumed, which is given by the function

$$f(x_H(t), y_H(t)) = 0 \quad (3)$$

which is differentiable. From the fact that the velocity of a point H is tangent to the path, the equation follows: $\text{grad} \vec{f} \times \vec{v}_H = 0$, which can be written in another form, i.e.:

$$\frac{\partial f}{\partial x_H} \dot{x}_H + \frac{\partial f}{\partial y_H} \dot{y}_H = 0 \quad (4)$$

Now the coordinates x_H, y_H and β can be determined from the following system of equations

$$\begin{cases} \dot{x}_H - v_A \cos \beta + l_3 \dot{\beta} \sin \beta = 0 \\ \dot{y}_H - v_A \sin \beta - l_3 \dot{\beta} \cos \beta = 0 \\ \frac{\partial f}{\partial x_H} \dot{x}_H + \frac{\partial f}{\partial y_H} \dot{y}_H = 0 \end{cases} \quad (5)$$

For a complete description of the robot's movement, equations are needed to describe the angular velocities of the drive wheels. These can be written in vector-matrix form

$$\begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & l_1 \\ 1 & -l_1 \end{bmatrix} \begin{bmatrix} v_A \\ \dot{\beta} \end{bmatrix} \quad (6)$$

where $l_1 = |AB| = |AC|$ and $\dot{\alpha}_1 \equiv \dot{\alpha}_1(t) \equiv \frac{d\alpha_1}{dt}$, $\dot{\alpha}_2 \equiv \dot{\alpha}_2(t) \equiv \frac{d\alpha_2}{dt}$ are the angular velocities of wheels 1 and 2.

By combining the systems of Equations (5) and (6) and solving them, it is possible to determine the five coordinates, i.e. $x_H, y_H, \beta, \alpha_1$ and α_2 assuming the velocity of point A , i.e. v_A and the path of the robot. Hence, it follows that the mobile wheeled robot is a system with two degrees of freedom.

2.2. Dynamics

An advantageous form of expression for the motion dynamics of a mobile wheeled robot due to control and identification issues is the vector-matrix form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) = \mathbf{u} \quad (7)$$

where $\mathbf{q} = [\alpha_1, \alpha_2]^T$ is the vector of angles of rotation of the mobile robot's drive wheels, $\mathbf{u} = [M_1, M_2]^T$ is the vector of driving moments, \mathbf{M} is the inertia matrix of the robot, $\mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{q}}$ is the vector of moments resulting from the centrifugal and Coriolis forces and $\mathbf{F}(\dot{\mathbf{q}})$ is the motion resistance vector.

The use of Lagrange equations to describe nonholonomic systems does not directly allow a description of the dynamics in the form of Equation (7) due to the multipliers. Therefore, another mathematical formalism is used in the literature to describe nonholonomic systems, such as the Maggie equation. [25,26].

Taking the parameter vector $\mathbf{a} = [a_1, \dots, a_6]$, matrices \mathbf{M}, \mathbf{C} and the vector \mathbf{F} from the Equation (7) can be written as

$$\mathbf{M} = \begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix}, \mathbf{C}(\dot{\mathbf{q}}) = \begin{bmatrix} 0 & 2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) \\ -2a_4(\dot{\alpha}_2 - \dot{\alpha}_1) & 0 \end{bmatrix}, \mathbf{F}(\dot{\mathbf{q}}) = \begin{bmatrix} a_5 \text{sign}(\dot{\alpha}_1) \\ a_6 \text{sign}(\dot{\alpha}_2) \end{bmatrix} \quad (8)$$

whereby the sgn function allows for the effect of the direction of movement on the resistance to movement to be taken into account. Due to the discontinuity of the sgn function, its approximation with the \tanh function was adopted in the numerical tests. Hence, the vector \mathbf{F} took the form:

$$\mathbf{F}(\dot{\mathbf{q}}) = \begin{bmatrix} a_5 \tanh(c_F \dot{\alpha}_1) \\ a_6 \tanh(c_F \dot{\alpha}_2) \end{bmatrix} \quad (9)$$

where c_F is a design constant. The dynamics of the wheeled robot in the form of Equation (7) constitutes its mathematical model used in the numerical tests. Furthermore, it is the basis for the synthesis of the adaptive control law, which is described in the next section.

3. Synthesis of Adaptive Control in a Tracking Task

An adaptive control algorithm is designed in such a way that it can modify its properties under varying operating conditions. In the case described here, the adaptive algorithm is used to control a mobile robot in a tracking task. The task of following a desired trajectory for mobile robots can be understood as determining such a solution for the initial conditions $[\mathbf{q}(0), \dot{\mathbf{q}}(0)] \in \Omega = \{(\mathbf{q}, \dot{\mathbf{q}}) : \mathbf{q}_d(0) = \mathbf{q}(0), \dot{\mathbf{q}}_d(0) = \dot{\mathbf{q}}(0)\}$, that converges to a manifold defined as: $\Omega_d = \{(\mathbf{q}, \dot{\mathbf{q}}) : \mathbf{q} = \mathbf{q}_d, \dot{\mathbf{q}} = \dot{\mathbf{q}}_d\}$, where \mathbf{q}_d is the vector of desired coordinates, $\dot{\mathbf{q}}_d$ is the vector of desired angular velocity resulting from the solution of the inverse kinematics problem [25].

The equation describing the dynamics of the mobile robot has the form (7). The tracking error in relation to angle of rotation and angular velocity was defined in the following form

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}, \quad \dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \quad (10)$$

In addition, a filtered tracking error was defined as

$$\mathbf{s} = \dot{\mathbf{e}} + \Lambda \mathbf{e} \quad (11)$$

where Λ is a diagonal design matrix with positive elements λ_i . By differentiating the generalised tracking error (11), the expression was obtained

$$\dot{\mathbf{s}} = \ddot{\mathbf{e}} + \Lambda \dot{\mathbf{e}} = \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} + \Lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (12)$$

Multiplying both sides of Equation (12) by the inertia matrix \mathbf{M} we obtain

$$\mathbf{M}\dot{\mathbf{s}} = \mathbf{M}\ddot{\mathbf{q}}_d - \mathbf{M}\ddot{\mathbf{q}} + \mathbf{M}\Lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (13)$$

By substituting Equation (7) into the above equation, the following was obtained

$$\mathbf{M}\dot{\mathbf{s}} = \mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{u} + \mathbf{M}\Lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (14)$$

and after transformation, the result is

$$\mathbf{M}\dot{\mathbf{s}} = \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda \dot{\mathbf{e}}) + \mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{u} \quad (15)$$

Multiplying both sides of relation (11) by matrix $-\mathbf{C}(\dot{\mathbf{q}})$ yields

$$-\mathbf{C}\mathbf{s} = -\mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{e}} - \mathbf{C}(\dot{\mathbf{q}})\Lambda \mathbf{e} \quad (16)$$

Hence it is easy to determine the expression

$$\mathbf{C}(\dot{\mathbf{q}})\dot{\mathbf{q}} = -\mathbf{C}(\dot{\mathbf{q}})\mathbf{s} + \mathbf{C}(\dot{\mathbf{q}})(\dot{\mathbf{q}}_d - \Lambda \mathbf{e}) \quad (17)$$

Taking this into account, relation (15) can be written as a function of the filtered tracking error \mathbf{s}

$$\mathbf{M}\dot{\mathbf{s}} = -\mathbf{C}(\dot{\mathbf{q}})\mathbf{s} + \mathbf{f}(\mathbf{x}) - \mathbf{u} \quad (18)$$

with non-linear function

$$\mathbf{f}(\mathbf{x}) = \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda \dot{\mathbf{e}}) + \mathbf{C}(\dot{\mathbf{q}})(\dot{\mathbf{q}}_d - \Lambda \mathbf{e}) + \mathbf{F}(\dot{\mathbf{q}}) \quad (19)$$

whereby the vector $\mathbf{x} = [\mathbf{e}, \dot{\mathbf{e}}, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d]$ contains the signals on which $\mathbf{f}(\mathbf{x})$ depends. By defining the auxiliary signals

$$\mathbf{v} = \dot{\mathbf{q}}_d - \Lambda \mathbf{e}, \quad \dot{\mathbf{v}} = \ddot{\mathbf{q}}_d + \Lambda \dot{\mathbf{e}} \quad (20)$$

the form of the function $\mathbf{f}(\mathbf{x})$ can be simplified to the form

$$\mathbf{f}(\mathbf{x}) = \mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\dot{\mathbf{q}})\mathbf{v} + \mathbf{F}(\dot{\mathbf{q}}) \quad (21)$$

Note that the function $\mathbf{f}(\mathbf{x})$ can be written in linear form with respect to the model parameters \mathbf{a} , i.e.

$$\mathbf{f}(\mathbf{x}) = \mathbf{Y}(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\mathbf{a} \quad (22)$$

The matrix $\mathbf{Y}(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})$ contains the functions resulting from the description of the robot's dynamics and has the form:

$$\mathbf{Y}(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}}) = \begin{bmatrix} \dot{v}_1 + \dot{v}_2 & \dot{v}_1 - \dot{v}_2 & \dot{v}_1 & 2(\dot{\alpha}_2 - \dot{\alpha}_1)v_2 & F_1(\dot{\alpha}_1) & 0 \\ \dot{v}_1 + \dot{v}_2 & -\dot{v}_1 + \dot{v}_2 & \dot{v}_2 & -2(\dot{\alpha}_2 - \dot{\alpha}_1)v_1 & 0 & F_2(\dot{\alpha}_2) \end{bmatrix} \quad (23)$$

where v_1, v_2 are the elements of vector \mathbf{v} , \dot{v}_1, \dot{v}_2 are the elements of vector $\dot{\mathbf{v}}$, $F_1(\dot{\alpha}_1), F_2(\dot{\alpha}_2)$ are the elements of vector $\mathbf{F}(\dot{\mathbf{q}})$.

Determining the value of the function $\mathbf{f}(\mathbf{x})$ requires knowledge of the parameter values. This is difficult to implement, especially in the case of varying operating conditions. Hence, adaptive control uses an estimate of the function $\mathbf{f}(\mathbf{x})$ i.e. $\hat{\mathbf{f}}(\mathbf{x})$ which is given as:

$$\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{Y}(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\hat{\mathbf{a}} \quad (24)$$

where $\hat{\mathbf{a}}$ is an estimate of the model parameter vector. This approach requires an extension of the control algorithm to include the parameter adaptation law, which is usually implemented as:

$$\dot{\hat{\mathbf{a}}} = \mathbf{P}\mathbf{Y}^T(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\mathbf{s} \quad (25)$$

It should be noted that this approach requires the assumption that the \mathbf{a} parameters are constant over the time intervals.

Usually, the law of parameter adaptation is applied in continuous time, but in the context of using adaptation techniques operating in discrete time, the law of parameter adaptation given in the form (25) can be written in discrete form [27]:

$$\frac{\hat{\mathbf{a}}_k - \hat{\mathbf{a}}_{k-1}}{h} = \mathbf{P}[\mathbf{Y}^T(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\mathbf{s}]^k \quad (26)$$

where h is a discrete step. In the context of the application of optimisation methods, the Equation (26) is more favourably written in the form of

$$\hat{\mathbf{a}}_k = \hat{\mathbf{a}}_{k-1} + \mathbf{P}h[\mathbf{Y}^T(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\mathbf{s}]^k \quad (27)$$

On this basis, the function to be minimised was defined in quadratic form

$$J = \argmin_{\hat{\mathbf{a}}} \mathbf{E}^T \mathbf{Q} \mathbf{E} \quad (28)$$

where $\mathbf{Q} = \mathbf{Q}^T > 0$ is a design matrix and $\mathbf{E} = \hat{\mathbf{a}}_{k-1} + \mathbf{P}h[\mathbf{Y}^T(\dot{\mathbf{q}}, \mathbf{v}, \dot{\mathbf{v}})\mathbf{s}]^k - \hat{\mathbf{a}}_k$. If optimisation methods, like genetic algorithms, are designed to search for a maximum, the function J can be transformed to the form

$$J^* = \argmax_{\hat{\mathbf{a}}} \frac{c}{J + c} \quad (29)$$

where $c > 0$ is a design constant.

We want to ensure the stability of the closed-loop system without changing the designed compensating control signal $\hat{\mathbf{f}}(\mathbf{x})$ given by Equation (24). This means that the control law \mathbf{u} must be designed in such a way that the filtered tracking error \mathbf{s} is bounded i.e.

$$|s_i| \leq \Theta_i, \quad i = 1, 2 \quad (30)$$

where $\Theta_i > 0$ is a constant value. For this purpose, the control signal was enriched with a supervising control \mathbf{u}_s , generated by an additional supervisory term, which is different from zero when the tracking error reaches the limits of the following set:

$$\{s_i : |s_i| \leq \Theta_i, i = 1, 2\} \quad (31)$$

In addition, a component responsible for stabilising tracking errors in the form of proportional-differential control expressed as \mathbf{Ks} is included. Finally, to ensure the stability of the closed system, the control law will take the form

$$\mathbf{u} = \hat{\mathbf{f}}(\mathbf{x}) + \mathbf{Ks} + \mathbf{I}_s \mathbf{u}_s \quad (32)$$

where $\mathbf{I}_s = \text{diag}[I_{s1}, I_{s2}]$ is a matrix with elements I_{si} , which take the value 1 when $s_i \geq \Theta_i$ and 0 otherwise. By substituting the Equation (32) into the Equation (18) the result was obtained:

$$\mathbf{M}\dot{\mathbf{s}} = -\mathbf{C}(\dot{\mathbf{q}})\mathbf{s} + \mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x}) - \mathbf{Ks} - \mathbf{I}_s \mathbf{u}_s \quad (33)$$

By defining a positive definite function $L = \frac{1}{2}\mathbf{s}^T \mathbf{M}\mathbf{s}$, differentiating it and considering the worst-case when tracking errors are large and supervisory control is active, i.e. $\mathbf{I}_s = \text{diag}[1, 1]$, the result is:

$$\dot{L} = \mathbf{s}^T \mathbf{M}\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^T \dot{\mathbf{M}}\mathbf{s} = -\mathbf{s}^T \mathbf{C}(\dot{\mathbf{q}})\mathbf{s} + \mathbf{s}^T \mathbf{f}(\mathbf{x}) - \mathbf{s}^T \hat{\mathbf{f}}(\mathbf{x}) - \mathbf{s}^T \mathbf{Ks} - \mathbf{s}^T \mathbf{u}_s + \frac{1}{2}\mathbf{s}^T \dot{\mathbf{M}}\mathbf{s} \quad (34)$$

After transformations, the following form was obtained:

$$\dot{L} = -\mathbf{s}^T \mathbf{Ks} + \frac{1}{2}\mathbf{s}^T (\dot{\mathbf{M}} - 2\mathbf{C})\mathbf{s} + \mathbf{s}^T (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})) - \mathbf{s}^T \mathbf{u}_s \quad (35)$$

Since $\dot{\mathbf{M}} - 2\mathbf{C}$ is a skew-symmetric matrix, the derivative of the scalar function (35) takes the form

$$\dot{L} \leq -\mathbf{s}^T \mathbf{Ks} + \sum_{i=1}^2 |s_i| \left[|f_i(\mathbf{x})| - |\hat{f}_i(\mathbf{x})| \right] - \sum_{i=1}^2 s_i u_{si} \quad (36)$$

If we select the supervisory control signal in the following form

$$\mathbf{u}_{si} = \left[F_i + |\hat{f}_i(\mathbf{x})| + \eta_i \right] \text{sgn}(s_i) \quad (37)$$

and assuming $|f_i(\mathbf{x})| \leq F_i, i = 1, 2, F_i > 0, \eta_i > 0$, we obtain:

$$\dot{L} < -\mathbf{s}^T \mathbf{Ks} - \sum_{i=1}^2 \eta_i |s_i| \quad (38)$$

The derivative of the Lyapunov function is negatively definite, so the closed system (18) is stable, for tracking errors s_i such that $|s_i| \geq \Theta_i$. When $|s_i| < \Theta_i$ then the supervisory control is inactive and the derivative of the Lyapunov function has the form $\dot{L} < -\mathbf{s}^T \mathbf{Ks}$. This proves that the filtered tracking error is bounded. The elements of the vector $\hat{\mathbf{a}}$ are generated by the genetic algorithm, which is formulated in such a way that it searches for solutions in a limited region of the parameter space. Therefore, the estimate of the parameter vector $\hat{\mathbf{a}}$ is by assumption bounded. Thus, no proof of this fact is required in the stability analysis.

There is a step function I_{s_i} in the control law (32), which activates the supervisory control signal u_s at the boundary $s_i = \Theta_i$. To eliminate the high switching frequency of the actuator, an approximation of $\text{sgn}(s_i)$ in the region defined by Θ_i is introduced, according to the relation

$$\text{sgn}(s_i) \approx \begin{cases} \frac{s_i}{\Theta_i}, & |s_i| < \Theta_i \\ \text{sgn}(s_i), & |s_i| \geq \Theta_i \end{cases}, \quad i = 1, 2 \quad (39)$$

A general diagram of the control system is shown graphically in Figure 2.

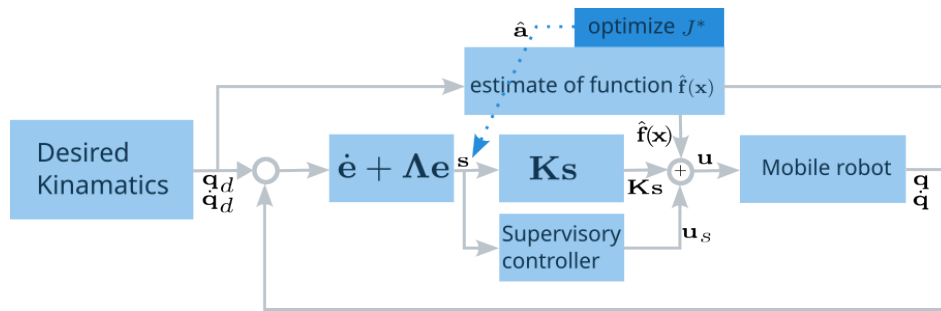


Figure 2. Scheme of the adaptive control system in the tracking task of a mobile wheeled robot

Finally, the adaptive control of the trailing motion of a wheeled mobile robot is in the form of the Equation (32) and is realised based on the estimation of the non-linear function $\hat{f}(x)$ and the parameter adaptation law (26), which is implemented as an optimisation problem of a discrete function given in the form (29). In this case, the optimisation algorithm is implemented based on genetic algorithms, which are described in the next section.

4. Classical Genetic Algorithm

A genetic algorithm is a non-linear, discrete and stochastic process, rather than a mathematically controlled algorithm, which was introduced by John Holland of the University of Michigan in 1975 [28]. Since then, this approach has found a permanent place in machine learning methods. Its task is understood as the search for the minimum or maximum of a function in a specified solution space [28,29]. The practice has been introduced that genetic algorithms search for the maximum of the objective function. Furthermore, the paper [28] states that, depending on the particular problem, it may be sufficient to determine a local maximum or a solution that is close to the local or global maximum. Hence the term that the genetic algorithm is an approximation algorithm, as it returns a solution for which the value of the objective function is as high as possible. The search for the maximum of the objective function, which is called the fitness function, is implemented based on the cyclic execution of the selection, crossover and mutation mechanism, which is preceded by the initialisation of the population. From this perspective, the genetic algorithm can be regarded as iterative.

The initialisation of the algorithm, i.e. the generation of the first population in the specified range, appears to have a significant impact on the success of the algorithm. This was confirmed in the paper [19], where the selection of the population was identified as the *main challenge* determining the quality of the solutions. It is not only its proper selection that is at the heart of initialisation. The choice of the type of chromosome representation is also made at this stage. The papers [20,21] note that the common coding scheme is binary coding, but octal, hexadecimal or decimal coding are also used. The latter was used in this work.

4.1. Selection

Selection reproduces the phenomenon of natural selection, the emergence of a group of the best individuals from a given population, i.e. those for which the values of the fitness function are greatest. The selection operation determines which of the genes in the current generation can constitute the parental group of the next generation. Such an operation is intended to ensure that the optimal solution

is approached. Several selection methods are described in the paper [30], including roulette wheel, linear ranked selection or tournament selection.

4.2. Crossover

Individuals that pass the selection process can generate new offspring. Through crossover operators, new offspring are created on the basis of the two parents [31]. Through crossover, using biological comparison, new genetic material is introduced, thereby maintaining genetic diversity [32]. In the context of an optimisation task, the crossover operator prevents the genetic algorithm from converging too quickly and allows a larger region of the solution space to be searched. Many crossover techniques can be found in the literature that differ in the way genes are exchanged between parents. An overview of these methods can be found in the paper [33].

In the work described in this article, the Simulated Binary Crossover (SBX) method is used for crossover. Its operation is based on binary individuals (encoded in decimal), which can be written as vectors $X^{(t)} = (x_1^{(t)}, x_2^{(t)}, \dots, x_n^{(t)})$ and $Y^{(t)} = (y_1^{(t)}, y_2^{(t)}, \dots, y_n^{(t)})$ where t encodes the iteration number of the genetic algorithm. In the SBX method, offspring generation starts with the determination of the coefficient β , który kontroluje odległość między rodzicami a potomstwem. which controls the distance between parents and offspring. Its value is calculated according to the relation:

$$\beta_{SBX} = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & u \leq 0.5, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}, & \text{if } u > 0.5. \end{cases} \quad (40)$$

where $u \in [0, 1]$ is the random crossover coefficient, and $\eta > 0$ is the parameter controlling its intensity. Based on the value of the β_{SBX} coefficient, the values of the offspring genes are determined:

$$\begin{aligned} x_i^{(t+1)} &= \frac{1}{2} (x_i^{(t)} + y_i^{(t)}) + \frac{1}{2} \beta_{SBX} (x_i^{(t)} - y_i^{(t)}), \\ y_i^{(t+1)} &= \frac{1}{2} (x_i^{(t)} + y_i^{(t)}) - \frac{1}{2} \beta_{SBX} (y_i^{(t)} - x_i^{(t)}). \end{aligned} \quad (41)$$

Based on the relationship (41) and taking into account the crossover interval (determines which elements of the vector representing the individual will be changed) and the mutation probability, the SBX algorithm can be presented in pseudocode form:

Algorithm 1 Simulated Binary Crossover (SBX)

```

1: function SBX( $X, Y, \lambda_{lb}, \lambda_{ub}, p_c \eta$ )    ▷ where  $X, Y$  is a pair of parents,  $\eta$  is the intensity parameter,
    $\lambda_{lb}, \lambda_{ub}$  is the crossover interval,  $p_c$  is the crossover probability
2:   if  $rand < p_c$  then
3:     for  $i = 1$  to  $n$  do
4:       if  $i \in [\lambda_{lb}, \lambda_{ub}]$  then
5:         choose randomly  $u \in [0, 1]$ 
6:         if  $u \leq 0.5$  then
7:            $\beta_{SBX} = (2u)^{\frac{1}{\eta+1}}$ 
8:         else
9:            $\beta_{SBX} = \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}$ 
10:        end if
11:         $x_i^{(t+1)} = \frac{1}{2}(x_i^{(t)} + y_i^{(t)}) + \frac{1}{2}\beta_{SBX}(x_i^{(t)} - y_i^{(t)})$ 
12:         $y_i^{(t+1)} = \frac{1}{2}(x_i^{(t)} + y_i^{(t)}) - \frac{1}{2}\beta_{SBX}(y_i^{(t)} - x_i^{(t)})$ 
13:      else
14:         $x_i^{(t+1)} = x_i^{(t)}$ 
15:         $y_i^{(t+1)} = y_i^{(t)}$ 
16:      end if
17:    end for
18:  else
19:     $X^{(t+1)} = X^{(t)}$ 
20:     $Y^{(t+1)} = Y^{(t)}$ 
21:  end if
22: end function

```

Figure 3 shows the values of the β_{SBX} coefficient for different values of the η parameter.

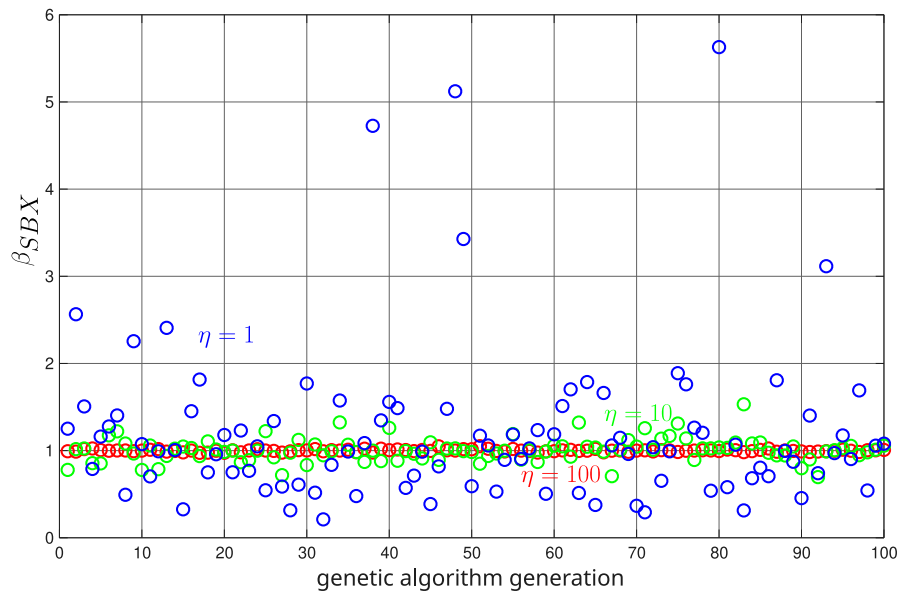


Figure 3. Values of the β_{SBX} coefficient for different values of the η parameter

The β_{SBX} values for $\eta = 100$ are close to 1 which means that the offspring genes will be more similar to those of the parents. In the case of $\eta = 1$, the values of β_{SBX} are more diverse, allowing the generation of more diverse offspring.

4.3. Mutation

The author of dissertation [34] notes, that one of the limitations of genetic algorithms is premature convergence, which results in a narrowing of the evolutionary process to a local optimum. The author of paper [17] sees the cause of this phenomenon in the lack of genetic diversity in the population, which is due to the fact that the crossover operator does not introduce new genes into the population. So this is a problem that needs to be solved. To do this, mutation operators are used, which by changing the values stored in the vectors representing the individuals, new areas of the considered search space can be explored. [34]. A number of mutation techniques can be found in the literature, which differ in the way they work. A list of these can be found in the paper [35]. A large number of citations relating to mutations of individuals encoded by real numbers can be found in the paper [20].

This work uses the Gaussian mutation method, which is one of the most commonly used mutation operators in genetic algorithms with coding by real numbers. It consists in adding a random number generated from a normal distribution to the value of the gene, which allows modifying the values of the elements of the vector $X = (x_1, x_2, \dots, x_n)$. These are modified according to the equation:

$$x_i^* = x_i + \mathcal{N}(0, \sigma_t^2) \quad (42)$$

where x_i^* is the value of the gene i after mutation and $\mathcal{N}(0, \sigma^2)$ denotes a random number from a normal distribution with mean $\mu = 0$ and variance σ_t^2 which can vary depending on the iteration of the genetic algorithm.

The pseudocode of the Gaussian mutation method, which takes into account the mutation probability p_m and the parameter σ_t , can be presented in the form:

Algorithm 2 Gaussian Mutation

```

1: function GAUSSIANMUTATION( $X, p_m, \sigma_t$ )           ▷ where  $X$  is a chromosome,  $p_m$  is the mutation
   probability,  $\sigma$  is the standard deviation
2:   for  $i = 1$  to  $n$  do
3:     choose randomly  $u \in [0, 1]$ 
4:     if  $u < p_m$  then
5:        $x_i^* = x_i + \mathcal{N}(0, \sigma_t^2)$ 
6:     end if
7:   end for
8: end function

```

4.4. Algorithm GABONST

The paper published in 2020 [22] proposes modifications to the classical genetic algorithm by adding mechanisms to incorporate Darwin's postulates of natural selection. The authors postulate that the genetic algorithm should incorporate a mechanism of natural selection that makes the best individuals likely to survive. These chances for weaker individuals are less, but possible in two cases: 1) crossover with an individual with a high ability to survive in the environment, which can lead to the transmission of favourable genetic features; 2) mutation, which can lead to favourable genetic features. If the individual resulting from these two cases does not meet the requirements of the environment, it may become extinct over time.

This biological description of the natural selection method was converted by the authors into the GABONST algorithm, available in the paper [22]. A modified version of it, which was used in this work, is presented below:

1. Generate an initial population $\mathbf{P}^{(0)} = \{\mathbf{p}_1^{(0)}, \mathbf{p}_2^{(0)}, \dots, \mathbf{p}_n^{(0)}\}$ consisting of n individuals that are stored as a vector. Set $gen = 0$.
2. Calculate the value of the fitness function $f_{GA}(\mathbf{p}_i^{gen})$ for each individual in the population $\mathbf{P}^{(gen)}$.
3. Return the smallest value of the fitness function for p_{best} percent of the best individuals and store it in the variable f_{best} .

4. For $i = 1, 2, \dots, n$ compare the value of the fitness function $f_{GA}(\mathbf{p}_i^{gen})$ with the value of f_{best} :
 - (a) If $f_{GA}(\mathbf{p}_i^{gen}) \geq f_{best}$: mutate an individual using the Gaussian method (with a variance σ_1 and a mutation probability p_{m_1}); record in the population $\mathbf{P}^{(gen+1)}$;
 - (b) If $f_{GA}(\mathbf{p}_i^{gen}) < f_{best}$:
 - i. Select a random individual from the set of best individuals k_{best} and write to the variable **best**.
 - ii. Perform a crossover of an individual \mathbf{p}_i^{gen} with individual **best** using the SBX method (with parameters η_2, p_{c_2} for all elements of the individual's coding vector) and save the single offspring to the variable \mathbf{p}_{cross} .
 - iii. If $f_{GA}(\mathbf{p}_{cross}) \geq f_{best}$ then save the individual \mathbf{p}_{cross} in the population $\mathbf{P}^{(gen+1)}$
 - iv. If $f_{GA}(\mathbf{p}_{cross}) < f_{best}$:
 - A. Perform a mutation of an individual \mathbf{p}_i^{gen} using the Gaussian method (with a variance σ_2 and a mutation probability p_{m_2}) and write to the variable \mathbf{p}_{mut} .
 - B. If $f_{GA}(\mathbf{p}_{mut}) \geq f_{best}$ then save the individual \mathbf{p}_{mut} in the population $\mathbf{P}^{(gen+1)}$.
 - C. If $f_{GA}(\mathbf{p}_{mut}) < f_{best}$ then generate a new individual and save it in the population $\mathbf{P}^{(gen+1)}$.
5. Increase gen by 1 and return to step 2.

The above genetic algorithm taking into account the mechanism of natural selection, in the context of this work, is used to adapt the parameters of the estimation of the function $\hat{\mathbf{f}}(\mathbf{x})$ in the control law (32). Therefore, there is no specific stopping criterion. In this application, the GABONST algorithm is executed N_{GA} times within one discrete step. After its completion, the individual whose fitness function is the largest is determined. This is the individual that encodes the value of the parameter estimate $\hat{\mathbf{a}}$.

4.5. Knowledge in the Genetic Algorithm

Knowledge of the fitness function optimised by the genetic algorithm can be used to control the optimisation process. In the context of the use of the GABONST algorithm in adaptive control, this could be the use of information about the filtered tracking error \mathbf{s} , the assumption of equal movement resistance at each driving wheel or the assumption of the variability of the model parameters, which are interval constant. In view of the above statement, several assumptions can be made regarding the assumed parameters of the GABONST algorithm:

1. The estimates of the values of the parameters related to resistance to movement in the initial population should be the same and within the assumed range.
2. The structure of the GABONST algorithm reflects the assumption of interval constancy of parameters.
3. The percentage of individuals, i.e. p_{best} determining how they are processed in the GABONST algorithm, should be dependent on the filtered tracking error \mathbf{s} .
4. The mutation intensity for the case when $f_{GA}(\mathbf{p}_i^{gen}) \geq f_{best}$ should increase as the filtered tracking error \mathbf{s} increases.
5. The influence of an individual *best* on the outcome of a crossover with an individual \mathbf{p}_i^{gen} should be reduced as the filtered tracking error \mathbf{s} increases.
6. In the case when $f_{GA}(\mathbf{p}_i^{gen}) < f_{best}$, the intensity of mutation that can lead to the inclusion of an individual in a new population should increase as the filtered tracking error \mathbf{s} increases.

The above points, which are written down as suggestions, arise from knowledge of the controlled object. The first point relates to the assumption of an interval of variation in movement resistance, which can be derived from knowledge of the surfaces on which the robot moves. The assumption of equality of movement resistance is related to the assumption that both wheels are on the same surface when the robot starts.

The second and third points refer to the assumption that the parameters of the robot model are constant intervals. This, in turn, means that for a given interval there is no need to adjust the parameters according to the idea of natural selection, but only a small mutation is necessary to fine-tune them. This approach will also reduce the noise associated with the operation of the genetic algorithm. Only if there is a change in the parameter values, which happens at the boundary of the interval, should a larger number of individuals undergo crossover and mutation, which in the GABONST algorithm is done in steps related to natural selection. Whether the robot's parameters are within the new interval can be determined by the value of the filtered tracking error \mathbf{s} , as discussed in point 3.

Points 4-6 deal with mutation and crossover intensity. In the case when $f_{GA}(\mathbf{p}_i^{gen}) \geq f_{best}$, an individual \mathbf{p}_i^{gen} can be assumed not to survive in a given environment. For the application under consideration, the intensity of mutation and crossover should increase as the filtered tracking error \mathbf{s} increases, on which the parameters of the SBX method (crossover) and the Gaussian method (mutation) can be made dependent. This can be realised based on the error norm of \mathbf{s} .

5. Numerical Test

This section describes the numerical test that was performed to verify the correct operation of the proposed adaptive control, in which the knowledge-based GABONST algorithm was used to adapt the parameters of the mathematical model of the wheeled robot. The test assumes that the desired trajectory of the robot point H is a fragment of the Lamé curve of the form

$$f(x_H(t), y_H(t)) = \left(\frac{x_H}{a_L}\right)^{2L} + \left(\frac{y_H}{b_L}\right)^{2L} - 1 \quad (43)$$

where $a_L, b_L > 0$, $L \in \mathbb{Z}^+$ are the design constants. From the Equations (5) and (6) were determined:

$$\begin{cases} \dot{x}_H = \frac{v_A x_H \left(\frac{y_H}{b_L}\right)^{2L}}{x_H \cos(\beta) \left(\frac{y_H}{b_L}\right)^{2L} - y_H \sin(\beta) \left(\frac{x_H}{a_L}\right)^{2L}} \\ \dot{y}_H = -\frac{v_A y_H \left(\frac{x_H}{a_L}\right)^{2L}}{x_H \cos(\beta) \left(\frac{y_H}{b_L}\right)^{2L} - y_H \sin(\beta) \left(\frac{x_H}{a_L}\right)^{2L}} \\ \dot{\beta} = -\frac{v_A y_H \cos(\beta) \left(\frac{x_H}{a_L}\right)^{2L} + v_A x_H \sin(\beta) \left(\frac{y_H}{b_L}\right)^{2L}}{l_3 \left(x_H \cos(\beta) \left(\frac{y_H}{b_L}\right)^{2L} - y_H \sin(\beta) \left(\frac{x_H}{a_L}\right)^{2L}\right)} \\ \dot{\alpha}_1 = \frac{v_A}{r} - \frac{v_A \left(l_1 y_H \cos(\beta) \left(\frac{x_H}{a_L}\right)^{2L} + l_1 x_H \sin(\beta) \left(\frac{y_H}{b_L}\right)^{2L}\right)}{l_3 r \left(x_H \cos(\beta) \left(\frac{y_H}{b_L}\right)^{2L} - y_H \sin(\beta) \left(\frac{x_H}{a_L}\right)^{2L}\right)} \\ \dot{\alpha}_2 = \frac{v_A}{r} + \frac{v_A \left(l_1 y_H \cos(\beta) \left(\frac{x_H}{a_L}\right)^{2L} + l_1 x_H \sin(\beta) \left(\frac{y_H}{b_L}\right)^{2L}\right)}{l_3 r \left(x_H \cos(\beta) \left(\frac{y_H}{b_L}\right)^{2L} - y_H \sin(\beta) \left(\frac{x_H}{a_L}\right)^{2L}\right)} \end{cases} \quad (44)$$

Using the Matlab/Simulink package, a coordinate vector $[x_H, y_H, \beta, \alpha_1, \alpha_2]$ for a wheeled robot moving along a section of the Lamé curve was determined based on the system of Equations (44). The length of the path is determined by the numerical test time ($t_k = 25[s]$) and the profile of velocity v_A which assumed that the acceleration and deceleration phases of the robot occurred in the 3rd and 18th second, respectively, at the maximum steady-state velocity $v_A^* = 0.4[m/s]$. The other parameters associated with the assumed kinematics equations are: $l_1 = 0.163[m]$, $l_3 = 0.2[m]$, $r = 0.0825[m]$, $a_L = 2[m]$, $b_L = 1[m]$ and $L = 5$.

Figure 4a shows the speed course of the drive wheels, where the acceleration and braking phases are visible. Figure 4b shows the trajectory of the wheeled robot in the reference system x_0y_0 . This is a fragment of the Lamé curve, which is described by the equation (43).

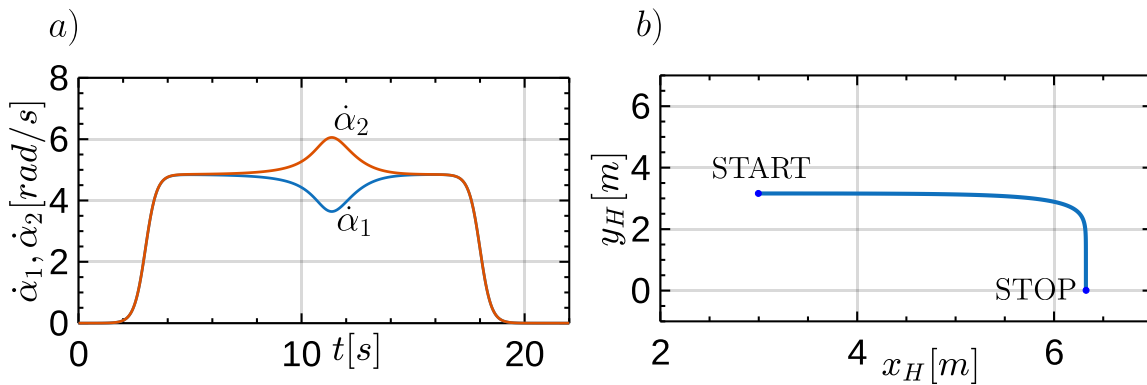


Figure 4. Przebieg prędkości kół napędzających robota (a) oraz tor ruchu robota kołowego w układzie odniesienia x_0y_0 (b)

The presented trajectory of the robot and the obtained kinematic parameters of the driving wheels will be the basis for determining the filtered tracking error given by Equation (11) in the tracking control task, for a wheeled robot described by the dynamic equation of motion (7). The parameters of the robot were assumed as follows: $a_1 = 0.0691$, $a_2 = 0.0511$, $a_3 = 0.0291$, $a_4 = 0.0014$, $a_5 = a_6 = 4.3001$. The control law was assumed according to the Equation (32) and for its implementation it was assumed that: $\mathbf{\Lambda} = 2\mathbf{I}_{2 \times 2}$, $\mathbf{K} = 2\mathbf{I}_{2 \times 2}$, $\Theta_1 = \Theta_2 = 1.1$, $F_1 = F_2 = 6$, $\eta_1 = \eta_2 = 0.3$.

In the numerical test considered, the model parameters that are related to the resistance to motion, i.e. a_5 and a_6 , were adapted, while the others were assumed constant. For their adaptation, the GABONST algorithm was used, which in this work was enriched with object knowledge. As an initial population, 16 individuals were randomly generated, which are represented by the vectors $\mathbf{p}_i = [p_{i1}, p_{i2}]$, $i = 1, 2, \dots, 16$, where p_{i1}, p_{i2} is the value of the parameters related to movement resistance. The values of p_{i1}, p_{i2} are randomly chosen from the range $[3.6001, 5.0001]$. Furthermore, due to the assumption that the wheels of the robot are on the same surface, it was assumed for the initial population that $p_{i1}^{(0)} = p_{i2}^{(0)}$.

The objective function that the genetic algorithm maximised is given by the Equation (29) with $c = 0.7$, $\mathbf{Q} = 12\mathbf{I}_{2 \times 2}$, $\mathbf{P} = 0.01\mathbf{I}_{2 \times 2}$. The genetic algorithm was implemented according to the procedure that is presented in Section 4.4. In one discrete step of the control algorithm, which was $h = 0.01[s]$, three loops of the genetic algorithm were implemented. The value of k_{best} was 8. The other parameter values of the GABONST algorithm used in the numerical test were dependent on the knowledge of the object, as indicated in section 4.5. Their detailed values are summarised in Table 1.

Table 1. Parameters of the GABONST algorithm in the numerical test.

Algorithm step according to GABONST procedure	Number of assumption in section 4.5	Parameters	Determining parameters method based on knowledge
(3): condition for the execution of natural selection	2,3	p_{best}	$p_{best} = 0.8 - 0.1 \mathbf{s} $
(a): mutation when $f_{GA}(\mathbf{p}_i^{gen}) \geq f_{best}$	4	σ_1, p_{m1}	$\begin{cases} \sigma_1 = 0.05, p_{m1} = 40\% & \mathbf{s} > 0.1 \\ \sigma_1 = 0.01, p_{m1} = 20\% & \mathbf{s} \leq 0.1 \end{cases}$
(i): crossover in natural selection	5	η_2, p_{c2}	$\begin{cases} \eta_2 = 10, p_{c2} = 100\% & \mathbf{s} > 0.1 \\ \eta_2 = 140, p_{c2} = 100\% & \mathbf{s} \leq 0.1 \end{cases}$
(A): mutation in natural selection	6	σ_2, p_{m2}	$\begin{cases} \sigma_2 = 0.65, p_{m2} = 60\% & \mathbf{s} > 0.1 \\ \sigma_2 = 0.1, p_{c2} = 20\% & \mathbf{s} \leq 0.1 \end{cases}$

In order to force the GABONST algorithm to adapt the model parameters, it was assumed in the numerical test that the parameters related to the movement resistance would have different values in

the time intervals. Therefore, it was assumed in the robot model that the parameters a_5 and a_6 would vary according to the relationship:

$$a_i = \begin{cases} 4.3001 & t \in [0, 5) \\ 5.5901 & t \in [5, 15] \\ 4.9451 & t \in [15, 25] \end{cases}, \quad i = 5, 6 \quad (45)$$

where t is the simulation time.

The control signals are presented in Figure 5. Figure 5a) shows the waveform of the overall control \mathbf{u} consisting of the stabilising control PD, i.e. $\mathbf{u}_{Ks} = \mathbf{K}\mathbf{s}$, the adaptive control $\hat{\mathbf{f}}(\mathbf{x})$ and the supervisory control \mathbf{u}_s . For the assumed parameter $\Theta_i = 1.1$ the supervisory control term \mathbf{u}_s did not activate because the condition $|s_i| < \Theta_i$ was satisfied. The system remained stable during the movement.

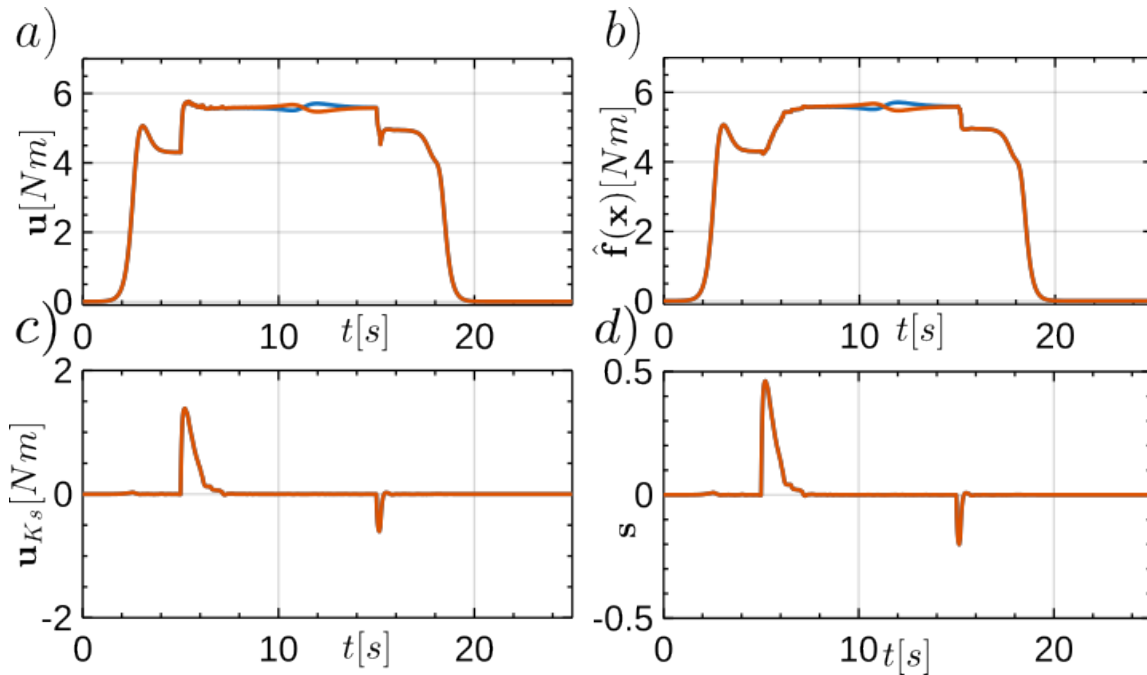


Figure 5. Control signals: (a) overall control, (b) PD control, (c) compensating controls, (d) filtered tracking error.

During the initial phase of the movement, the control $\hat{\mathbf{f}}(\mathbf{x})$ (rys. 5b), is immediately activated, as the initial values of the parameter evaluations are consistent with the values assumed in the robot dynamics model and the initial population of the GABONST algorithm was drawn near the nominal values of the parameters a_5 and a_6 . Only when the parameter values in the model are increased by 30 % (which happens at time $t = 5[s]$) is the PD control (Figure 5c) activated. This is related to the higher values of the tracking errors \mathbf{e} and $\dot{\mathbf{e}}$, and therefore also the filtered tracking error \mathbf{s} (rys. 5d). This is due to the maladaptation of the values of the parameter estimates, which are represented by the individual with the maximum value of the fitness function after three loops of the GABONST algorithm in a given iteration step. As the adaptation progresses, the compensating control becomes more and more adequate to the changed dynamics of the object, while the PD control signal decreases. The quality of adaptation is also confirmed by the course of the filtered tracking error \mathbf{s} , which remains close to zero after the increase due to the change in the value of the parameters a_5 and a_6 .

The above description is also valid for the situation occurring at time $t = 15[s]$ when the values of the parameters a_5 and a_6 change and reach 115 % of the nominal values. Then, analogously as before, the role of PD control temporarily increases and then decreases (in amplitude) as the adaptive control $\hat{\mathbf{f}}(\mathbf{x})$ adapts to the new values of the changed parameters.

The Figure 6a) shows the course of adaptation of the parameter estimates related to motion resistance. As mentioned earlier, the initial population was chosen so that no significant adaptation is

required during the initial phase of the numerical test. This changes when for $t = 5[s]$ there is a change in the value of the parameters a_5 and a_6 . Then, as shown in Figure 6a), the values of the parameter estimates \hat{a}_5 and \hat{a}_6 adapt to the new values. This is related to the operation of the genetic algorithm, which seeks to maximise the fitness function (Figure 6b). An analogous situation occurs when the parameters are changed for $t = 15[s]$

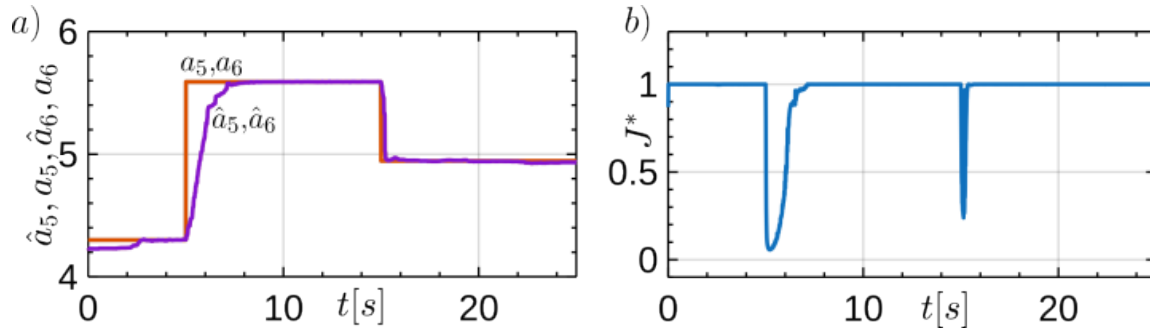


Figure 6. Course of adaptation of parameter estimates (a) and the fitness function J^* (b).

The above comments on the generalised tracking error s are confirmed by the waveform of the error e , which values remain small (Figure 7a). Similar conclusions arise from the waveforms of the desired and obtained angular velocities of the robot wheels (Figure 7b). These differ significantly only when there are disturbances, i.e. for $t = 5[s]$ and $t = 15[s]$.

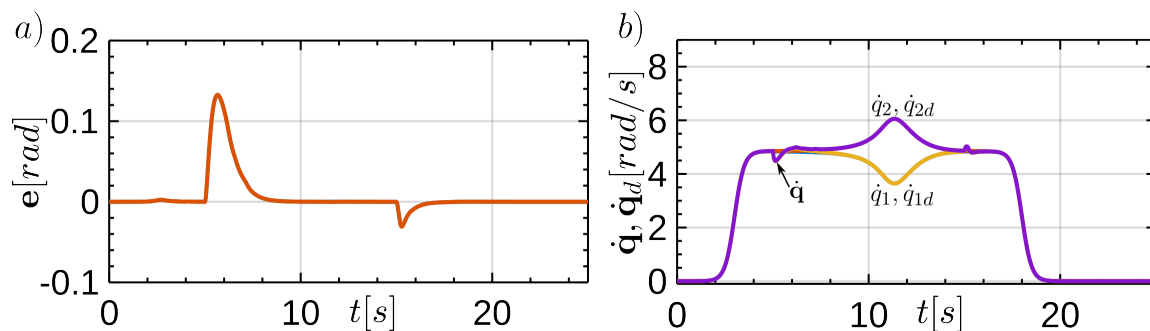


Figure 7. Tracking error e (error waveforms for both wheels have similar values) (a) and desired and obtained angular velocities (b).

6. Conclusions

In this paper, the synthesis of tracking control of a mobile wheeled robot was carried out. A genetic algorithm that takes into account the idea of natural selection was used to adapt the estimates of parameters required for the implementation of compensating control. The method provides an opportunity to take into account knowledge of the object's operating conditions, which usually change interval-wise, for example, due to a change in the roadway surface, as studied in the article, or a change in the load being carried. In such a situation, an increase in the intensity of crossover and mutation is required, which is allowed by the algorithm used. This results in a rapid adjustment of the model parameter estimates to the new conditions. On the other hand, when the conditions of the robot do not change significantly, that is, the parameters of the robot do not undergo significant changes, the intensity of crossover and mutation in the algorithm is reduced so that there is no oscillation of parameter estimates.

The results of the simulation indicated the correctness of the assumptions made, in particular that the control system remains stable and the tracking errors are limited. Based on the values of tracking errors, the quality of the tracking task execution should be considered good. In addition, the adapted parameter estimates tend towards the values of the model parameters, and the adaptation algorithm shows sensitivity to parameter changes during the robot's operation. Further work planned will be the study of the impact of selection, crossover and mutation algorithms on the quality of control with a

particular focus on the impact of object knowledge on the parameters of these operations. The next stage of the research will be the verification of solutions using the real robot.

Author Contributions: Conceptualization, P.P.; methodology, P.P., P.G.; software, P.P.; validation, P.G.; formal analysis, P.P.; writing—original draft preparation, P.P., P.G.; writing—review and editing, P.P., P.G.; supervision, P.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The data are contained within the article

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Annaswamy, A.M.; Fradkov, A.L. A Historical Perspective of Adaptive Control and Learning. *Annual Reviews in Control* **2021**, *52*, 18–41. <https://doi.org/10.1016/j.arcontrol.2021.10.014>.
2. Verrelli, C.M.; Tomei, P. Adaptive Learning Control for Nonlinear Systems: A Single Learning Estimation Scheme Is Enough. *Automatica* **2023**, *149*, 110833. <https://doi.org/10.1016/j.automatica.2022.110833>.
3. Lee, H.G. *Linearization of Nonlinear Control Systems*; Springer Nature Singapore: Singapore, 2022. <https://doi.org/10.1007/978-981-19-3643-2>.
4. Homma, T.; Yamaura, H. Development of an Identification Method for the Minimal Set of Inertial Parameters of a Multibody System. *Multibody System Dynamics* **2025**, *63*, 435–452. <https://doi.org/10.1007/s11044-024-10026-0>.
5. Benchouche, W.; Mellah, R.; Bennouna, M.S. The Impact of the Dynamic Model in Feedback Linearization Trajectory Tracking of a Mobile Robot. *Periodica Polytechnica Electrical Engineering and Computer Science* **2021**, *65*, 329–343. <https://doi.org/10.3311/PPee.17127>.
6. Hassan, N.; Saleem, A. Neural Network-Based Adaptive Controller for Trajectory Tracking of Wheeled Mobile Robots. *IEEE Access* **2022**, *10*, 13582–13597. <https://doi.org/10.1109/ACCESS.2022.3146970>.
7. Esfandiari, K.; Abdollahi, F.; Talebi, H.A. *Neural Network-Based Adaptive Control of Uncertain Nonlinear Systems*; Springer International Publishing: Cham, 2022. <https://doi.org/10.1007/978-3-030-73136-6>.
8. Ge, S.; Wang, C. Adaptive Neural Control of Uncertain MIMO Nonlinear Systems. *IEEE Transactions on Neural Networks* **2004**, *15*, 674–692. <https://doi.org/10.1109/TNN.2004.826130>.
9. Gierlak, P. Adaptive Position/Force Control of a Robotic Manipulator in Contact with a Flexible and Uncertain Environment. *Robotics* **2021**, *10*, 32. <https://doi.org/10.3390/robotics10010032>.
10. Huang, Z.; Bai, W.; Li, T.; Long, Y.; Chen, C.P.; Liang, H.; Yang, H. Adaptive Reinforcement Learning Optimal Tracking Control for Strict-Feedback Nonlinear Systems with Prescribed Performance. *Information Sciences* **2023**, *621*, 407–423. <https://doi.org/10.1016/j.ins.2022.11.109>.
11. Penar, P.; Hendzel, Z. Experimental Verification of the Differential Games and H_∞ Theory in Tracking Control of a Wheeled Mobile Robot. *Journal of Intelligent & Robotic Systems* **2022**, *104*, 61. <https://doi.org/10.1007/s10846-022-01584-6>.
12. Liu, Z.; Kambali, P.N.; Nataraj, C. Hybrid Adaptive Modeling Using Neural Networks Trained with Nonlinear Dynamics Based Features, 2025. <https://doi.org/10.48550/ARXIV.2501.11835>.
13. Annaswamy, A.M.; Johansson, K.H.; Pappas, G. Control for Societal-Scale Challenges: Road Map 2030. *IEEE Control Systems* **2024**, *44*, 30–32. <https://doi.org/10.1109/MCS.2024.3382376>.
14. Wang, J.; Li, Y.; Gao, R.X.; Zhang, F. Hybrid Physics-Based and Data-Driven Models for Smart Manufacturing: Modelling, Simulation, and Explainability. *Journal of Manufacturing Systems* **2022**, *63*, 381–391. <https://doi.org/10.1016/j.jmsy.2022.04.004>.
15. Rai, R.; Sahu, C.K. Driven by Data or Derived Through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques With Cyber-Physical System (CPS) Focus. *IEEE Access* **2020**, *8*, 71050–71073. <https://doi.org/10.1109/ACCESS.2020.2987324>.
16. Kasilingam, S.; Yang, R.; Singh, S.K.; Farahani, M.A.; Rai, R.; Wuest, T. Physics-Based and Data-Driven Hybrid Modeling in Manufacturing: A Review. *Production & Manufacturing Research* **2024**, *12*, 2305358. <https://doi.org/10.1080/21693277.2024.2305358>.

17. Korani, W.; Mouhoub, M. Review on Nature-Inspired Algorithms. *Operations Research Forum* **2021**, *2*, 36. <https://doi.org/10.1007/s43069-021-00068-x>.
18. Alam, T.; Qamar, S.; Dixit, A. Genetic Algorithm: Reviews, Implementations, and Applications.
19. Anwaar, A.; Ashraf, A.; Bangyal, W.H.K.; Iqbal, M. Genetic Algorithms: Brief Review on Genetic Algorithms for Global Optimization Problems. In Proceedings of the 2022 Human-Centered Cognitive Systems (HCCS), Shanghai, China, 2022; pp. 1–6. <https://doi.org/10.1109/HCCS55241.2022.10090327>.
20. Katoch, S.; Chauhan, S.S.; Kumar, V. A Review on Genetic Algorithm: Past, Present, and Future. *Multimedia Tools and Applications* **2021**, *80*, 8091–8126. <https://doi.org/10.1007/s11042-020-10139-6>.
21. Faizatulhaida Md Isa.; Wan Nor Munirah Ariffin.; Muhammad Shahar Jusoh.; Erni Puspanantasari Putri. A Review of Genetic Algorithm: Operations and Applications. *Journal of Advanced Research in Applied Sciences and Engineering Technology* **2024**, *40*, 1–34. <https://doi.org/10.37934/araset.40.1.134>.
22. Albadr, M.A.; Tiun, S.; Ayob, M.; AL-Dhief, F. Genetic Algorithm Based on Natural Selection Theory for Optimization Problems. *Symmetry* **2020**, *12*, 1758. <https://doi.org/10.3390/sym12111758>.
23. Siciliano, B.; Khatib, O., Eds. *Springer Handbook of Robotics*; Springer: Berlin, 2008.
24. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*, 3. print ed.; Wiley: New York, 1989.
25. Szuster, M.; Hendzel, Z. *Intelligent Optimal Adaptive Control for Mechatronic Systems*; Vol. 120, *Studies in Systems, Decision and Control*, Springer International Publishing: Cham, 2018. <https://doi.org/10.1007/978-3-319-68826-8>.
26. Giergiel, J.; Żylski, W. Description of Motion of a Mobile Robot by Maggie's Equations. *Journal of Theoretical and Applied Mechanics* **2005**, *43*.
27. Penar, P.; Hendzel, Z. Adaptive Controller Using Genetic Algorithm for Autonomous Wheeled Mobile Robot. In *Automation 2024: Advances in Automation, Robotics and Measurement Techniques*; Szewczyk, R.; Zieliński, C.; Kaliczyńska, M.; Bučinskas, V., Eds.; Springer Nature Switzerland: Cham, 2024; Vol. 1219, pp. 223–232. https://doi.org/10.1007/978-3-031-78266-4_20.
28. Alhijawi, B.; Awajan, A. Genetic Algorithms: Theory, Genetic Operators, Solutions, and Applications. *Evolutionary Intelligence* **2024**, *17*, 1245–1256. <https://doi.org/10.1007/s12065-023-00822-6>.
29. Bodenhofer, U. Genetic Algorithms: Theory and Applications.
30. Jebari, K.; Madiafi, M. Selection Methods for Genetic Algorithms.
31. Drachal, K.; Pawłowski, M. A Review of the Applications of Genetic Algorithms to Forecasting Prices of Commodities. *Economies* **2021**, *9*, 6. <https://doi.org/10.3390/economies9010006>.
32. Lambora, A.; Gupta, K.; Chopra, K. Genetic Algorithm- A Literature Review. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019; pp. 380–384. <https://doi.org/10.1109/COMITCon.2019.8862255>.
33. Gwiazda, T.D. *Crossover for Single-Objective Numerical Optimization Problems*; TOMASZGWIAZDA E-BOOKS: Łomianki, 2006.
34. Optymalizacja Procesu Dystrybucji Zasobów w Złożonych Sieciach Logistycznych Przy Użyciu Algorytmów Genetycznych. PhD thesis.
35. Gwiazda, T.D. *Genetic Algorithms Reference: Mutation Operator for Numerical Optimization Problems. Volume II* / Tomasz Dominik Gwiazda; Tomasz Gwiazda: Poland, 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.