



# Investigation of Internal Model for Unmanned Vehicle Control in Case of Its Aggressive Motion along a Spatial Trajectory <sup>†</sup>

Igor Prokopiev <sup>1</sup>, Elena Sofronova <sup>1,\*</sup>  and Viktoria Moiseenko <sup>2</sup> 

<sup>1</sup> Federal Research Center “Computer Science and Control”, Russian Academy of Sciences, 44 bld. 2, Vavilova Str., 119333 Moscow, Russia

<sup>2</sup> Engineering Academy, RUDN University, 6 Miklukho-Maklaya St., 117198 Moscow, Russia

\* Correspondence: [sofronova\\_ea@mail.ru](mailto:sofronova_ea@mail.ru)

<sup>†</sup> Presented at the 15th International Conference “Intelligent Systems” (INTELS’22), Moscow, Russia, 14–16 December 2022.

**Abstract:** The work is devoted to the study of methods that are used to control the movement of an object along a given trajectory. A control method involving an accurate internal model is proposed. This internal model was built on the basis of the object’s mathematical model and real object, performed by artificial neural networks. For a limited period of time the model is able to determine the object state without surveillance system usage. The dynamic model of an unmanned vehicle was obtained by method developed at the Robotics Center of the FRC CSC RAS. This method acquires experimental data and performs model identification by means of a neural network. The trajectory is a set of spatial points generated by the developed real unmanned vehicle simulator. The control was carried out on the basis of PID-controller and model predictive control method. The comparison of control methods for a real and virtual unmanned vehicles was conducted in the simulator developed. The results of field experiments, during which control by internal model was applied, are presented.

**Keywords:** identification; unmanned vehicle; path tracking



**Citation:** Prokopiev, I.; Sofronova, E.; Moiseenko, V. Investigation of Internal Model for Unmanned Vehicle Control in Case of Its Aggressive Motion along a Spatial Trajectory. *Eng. Proc.* **2023**, *33*, 38. <https://doi.org/10.3390/engproc2023033038>

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 20 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

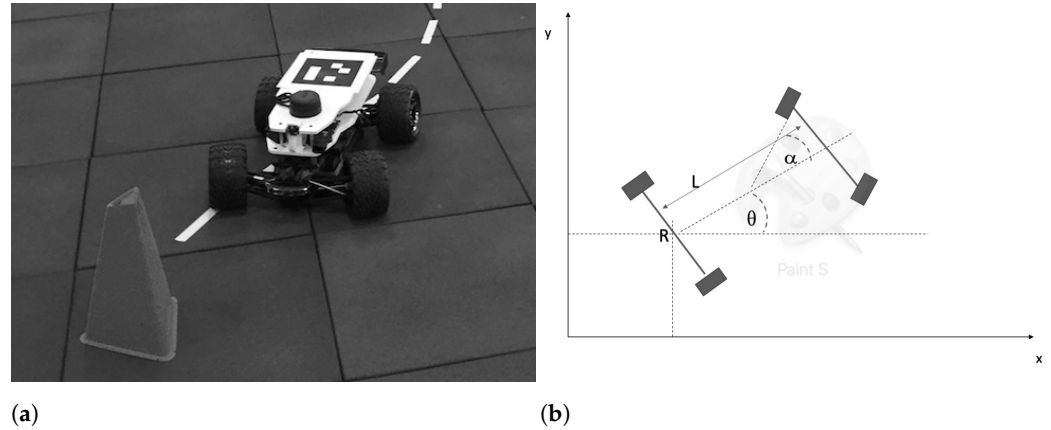
Unmanned vehicles control from the initial state to the final one should meet the desired quality requirements. At the same time, it is necessary to synthesize such control laws that ensure the achievement of the control goal in the wide class of the uncertainty of vehicle’s dynamics. To synthesize the required control laws, the unmanned vehicles control system simulation should be performed. A mathematical model of a real system can be obtained with the help of well-known identification methods [1,2]. Herein the following system identification methods are used, ARMAX (autoregressive moving average models with exogenous inputs) and NARMAX (non-linear ARMAX). These methods allow to obtain linear and non-linear polynomial functions that model the relationship between the perception of the robot sensor and its motor response. These methods are extended to neural network structures [3] and are successfully applied to non-linear unmanned vehicles.

One of the approaches in the field of unmanned mobile systems, such as unmanned vehicles operating in automatic mode, is aimed at the development of non-linear control laws for real time trajectories tracking [4–6].

Unmanned vehicles’ real-time trajectory control is usually divided into three sub-tasks, trajectory generation, position determination, and trajectory tracking.

This paper presents the results of model identification of a real unmanned vehicle. Kinematic equations of unmanned vehicle describe mass center position ( $x, y$ ) and direction of movement ( $\theta$ ). These parameters depend on linear velocity ( $v$ ) and rotation angle of front wheels ( $\alpha$ ). Unmanned vehicle and its kinematic movement control scheme are presented in Figure 1.

Artificial neural networks belong to modern model identification methods used in robotics. Combined with differential equations, ANNs provide satisfying results in real non-linear dynamics systems identification.



**Figure 1.** Unmanned vehicle (a) and its kinematic movement control scheme (b).

The article presents a formal statement of the model identification problem. A universal strategy for its solution that uses ANNs is proposed. The obtained model, or so-called the internal one, is applied in a special simulator developed to automate path generation and the process of obtaining the state of the unmanned vehicle that is conducted in selected control methods in case of some parametrical changes.

## 2. Unmanned Vehicle Model Identification

The unmanned vehicle (UV) is built on the basis of an automobile chassis on a scale of 1:10, has a weight of 1.5 kg and can reach speed up to 10 m/s. The appearance of the UV is shown in Figure 1a. The front wheels of the UV implement steering, the rear ones monitor the state of the robot. Figure 1b shows the kinematic movement control scheme of the unmanned vehicle, where  $x, y$  refer to the coordinates;  $\alpha$  is the angle of front wheels rotation;  $\theta$  is the UV orientation relative to the  $x$  axis;  $L$  is the distance between the front and rear axes of the robot; and  $R$  is the base point located in the center of the robot's rear axis.

Experimental data on the vehicle movement was obtained according to a special technique developed in the Robotics Center of the FRC CSC RAS.

The vehicle model consists of hybrid finite-difference equations of the following form

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \\ v(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + \Delta t v(k-1) \cos(\theta(k-1)) \\ y(k-1) + \Delta t v(k-1) \sin(\theta(k-1)) \\ f^\theta(v(k-1), \theta(k-1), u(k), \omega(k), \Delta t) \\ f^v(v(k-1), \theta(k-1), u(k), \omega(k), \Delta t) \end{bmatrix}, k = 1, \dots, K, \quad (1)$$

where  $\mathbf{x}(k)$  is a state vector,

$$\mathbf{x}(k) = [x(k) \ y(k) \ \theta(k) \ v(k)]^T = [x(k) \ y(k) \ \mathbf{u}(k)]^T, \quad (2)$$

$v(k)$  is the velocity of the object,  $\theta(k)$  is the UV orientation relative to the  $x$  axis,  $\omega(k)$  is the component of control for orientation  $\theta(k)$ ,  $u(k)$  is the component of control for linear velocity  $v(k)$ ,  $\Delta t$  is the time span,  $f^\theta(v(k-1), \theta(k-1), u(k), \omega(k), \Delta t)$  and  $f^v(v(k-1), \theta(k-1), u(k), \omega(k), \Delta t)$  are the outputs of a neural network,  $K$  is the number of control steps.

Object movement is determined by orientation  $\theta(k)$  and velocity  $v(k)$  being predicted by neural network. The inputs of the neural network include orientation and speed control signals  $\omega(k)$  and  $u(k)$  along with the previous state of orientation  $\theta(k-1)$  and velocity  $v(k-1)$ . At the output, we obtain the current values of orientation  $\theta(k)$  and velocity  $v(k)$ .

The neural network has the following structure

$$\mathbf{z}^{(l)} = \boldsymbol{\varphi}^{(l)} \left( \mathbf{W}^{(l)} \mathbf{z}^{(l-1)} + \mathbf{z}_0^{(l)} \right), l = 1, \dots, L, \quad (3)$$

where  $\mathbf{z}^{(l)}$  is the output vector of layer  $l$ ,  $\boldsymbol{\varphi}^{(l)}(\mathbf{s})$  is a vector activation function of layer  $l$ ,  $\mathbf{W}^{(l)}$  is a weight matrix of layer  $l$ ,  $\mathbf{W}^{(l)} = [w_{ij}^{(l)}]$ ,  $i = 1, \dots, n_l, j = 1, \dots, n_{l-1}$ ,  $\dim \mathbf{W}^{(l)} = n_l \times n_{l-1}$ ,  $\mathbf{z}_0^{(l)}$  is a bias of  $l$ ,  $\mathbf{z}_0^{(l)} = [v(k-1) \quad \theta(k-1) \quad u(k) \quad \omega(k) \quad \Delta t]^T$ ,  $L$  is the number of layers,  $\mathbf{z}^{(L)} = [\theta(k) \quad v(k)]^T$ ,  $n_0 = 5$ ,  $n_L = 2$ .

Activation function  $\boldsymbol{\varphi}^{(l)}(\mathbf{s})$  is applied to each component of the state vector

$$\boldsymbol{\varphi}^{(l)}(\mathbf{s}) = [\varphi_1^{(l)}(s_1) \quad \dots \quad \varphi_{n_l}^{(l)}(s_{n_l})]^T.$$

The sought-for parameters of the neural network include the components of the weight matrix and bias of each layer. The total number of the parameters sought is  $\sum_{l=1}^L n_l(n_{l-1} + 1)$ .

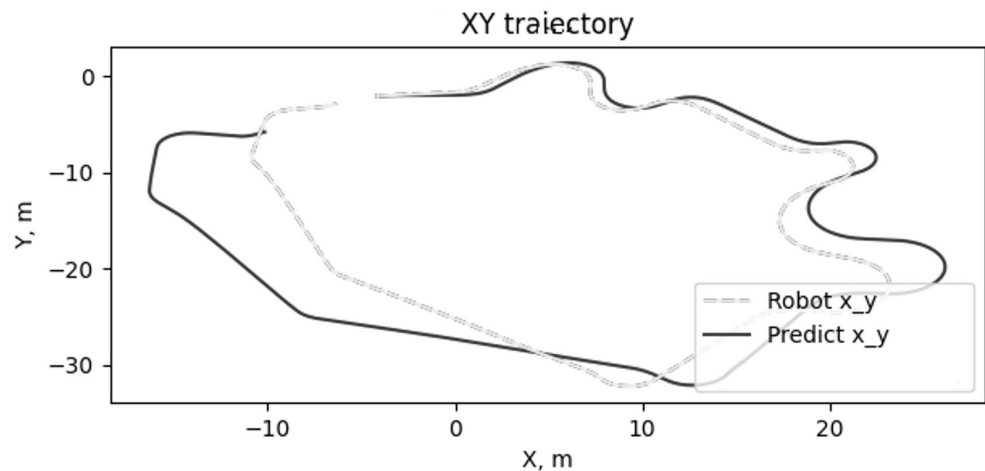
The quality criterion for neural network parameter optimization is the root-mean-square error of state prediction (2)

$$\sum_{k=1}^K \|\mathbf{z}^L(k) - \mathbf{u}(k)\|_2 \rightarrow \min. \quad (4)$$

For the experiments the following neural network was used:  $L = 4$ ,  $n_0 = 5$ ,  $n_1 = n_2 = n_3 = 64$ ,  $n_4 = 2$ . All layers but the last used ReLu as activation function. To find the vector of optimal neural network parameters the stochastic gradient descent method ADAM was used [7].

To collect the required amount of data in the entire robot operating control range, VISLAM system was used. This system is based on RealSense t265 camera, that records data on control and state vectors at frequency of 20 Hz.

Figure 2 shows a trajectory obtained from a training sample (the light dotted line) and a trajectory obtained by internal model (1) with a neural network (the solid line).



**Figure 2.** The trajectories of the vehicle movement on the plane  $\{x, y\}$ .

Figure 3 depicts the change of state vector components  $v(t)$  and  $\theta(t)$  over time. The light line shows the movement of a real vehicle, the solid one corresponds to the movement according to the model (1), and the dotted line corresponds to control components.

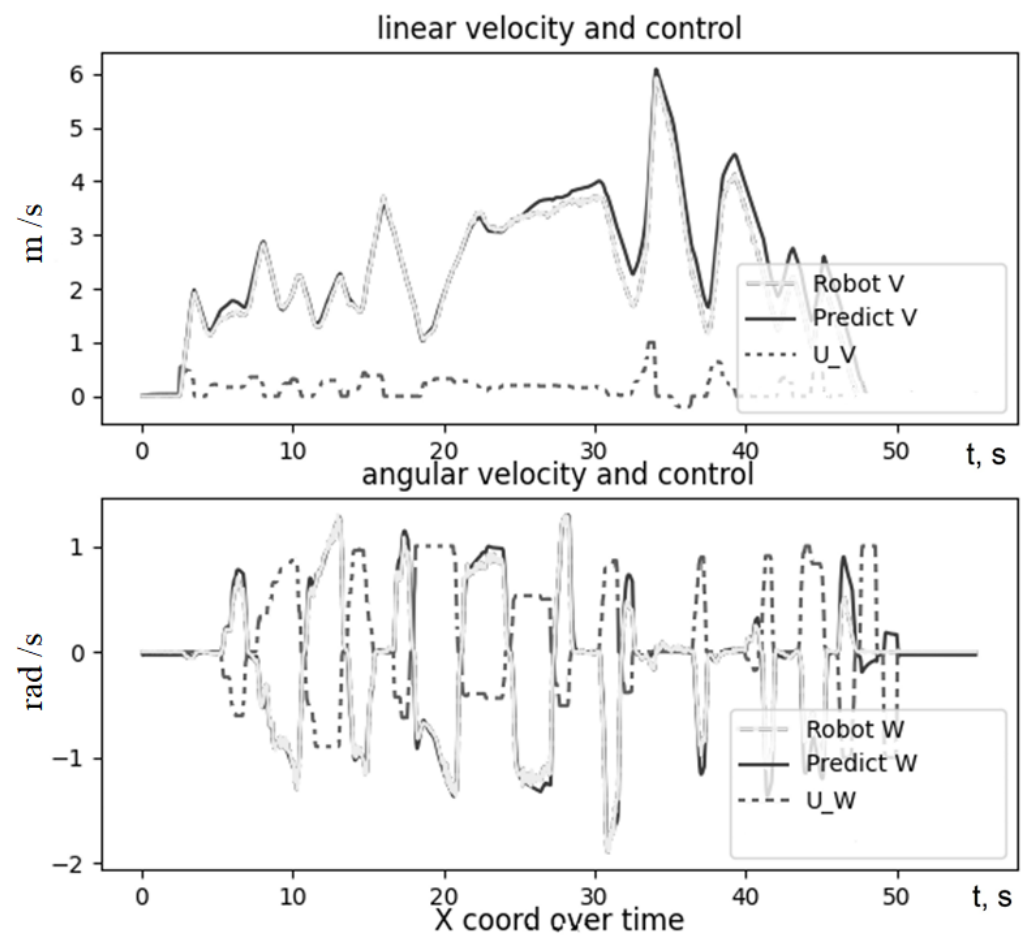


Figure 3. The change of state and control components over time.

### 3. Trajectory Tracking Scheme

To implement motion control along the trajectory, it is necessary to solve the control synthesis problem [8] and represent control as function that depends on the coordinates of the state space. In the majority of cases, a trajectory tracking error is calculated and a control signal, that works out this error, is set.

Depending on the task, the desired trajectory is set as a sequence of points on a plane or in three-dimensional space, and instead of calculating the distance to the trajectory, the distance to the points of the desired trajectory is determined. The block diagram of the unmanned vehicle control system, in which either PID-controller or model predictive control is used, is shown in Figure 4.

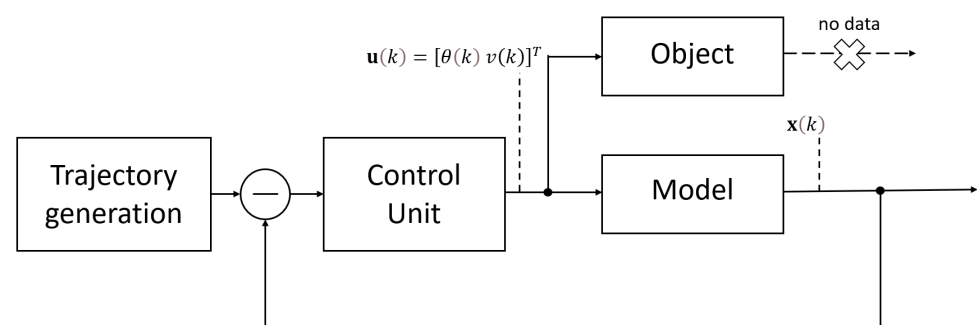


Figure 4. Block diagram of an internal model control system.

## 4. Controllers

### 4.1. Closed-Loop Control

In the steering control loop of the vehicle a P-controller was used. The control signal had the following form

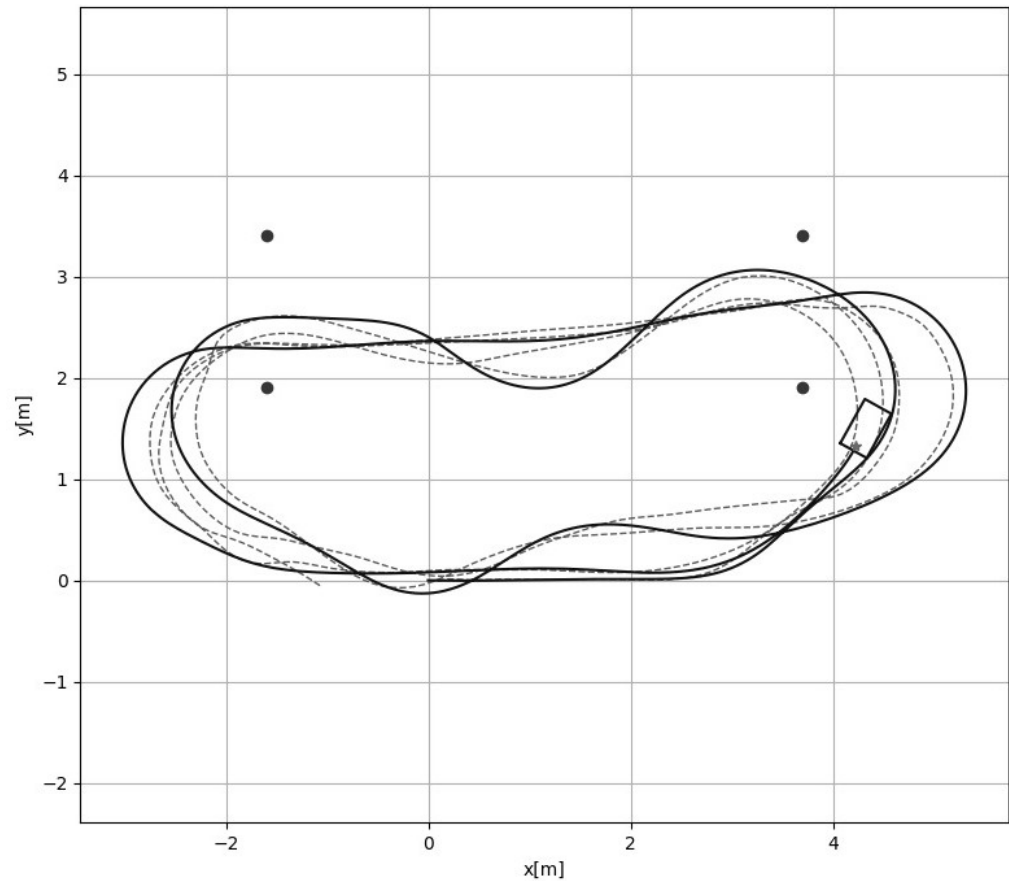
$$\alpha = \arctan \frac{L\omega}{v}, \quad (5)$$

where  $L$  is the distance between the axes of the robot,  $v$  and  $\omega$  is the linear and angular velocities of the robot,

$$\omega = \frac{vk(s)\cos(\theta_e)}{1 - k(s)e} - k_\theta |v| \theta_e - \left( k_e v \frac{\sin(\theta_e)}{\theta_e} \right) e, \quad (6)$$

where  $k(s)$  is a path curvature,  $\theta_e$  is an error in motion direction,  $e$  is a position error, and  $k_\theta$  and  $k_e$  are controller coefficients.

Figure 5 shows the results of tracking the desired trajectory using P-controller and an internal model. The dashed line shows the desired trajectory, and the solid line shows the trajectory that has been tracked. The dots mark the constraints in the form of gates.



**Figure 5.** The movement along the trajectory with the help of internal model and P-controller.

### 4.2. Model Prediction Control

Predictive control is an optimization strategy that has a variety of advantages, such as accounting constraints imposed on system's state and control signals, and ability to work with non-linear objects. The optimal trajectory is the solution to the optimization problem. Predictive control includes object simulation under various control actions that ensure current problem solution, for example, bypassing the obstacles or passing through a gate. Among the resulting set of trajectories, one trajectory is selected according to some

criterion. As a criterion, the squared error of achieving the goal and the squared control signal with some weighting factors are often used

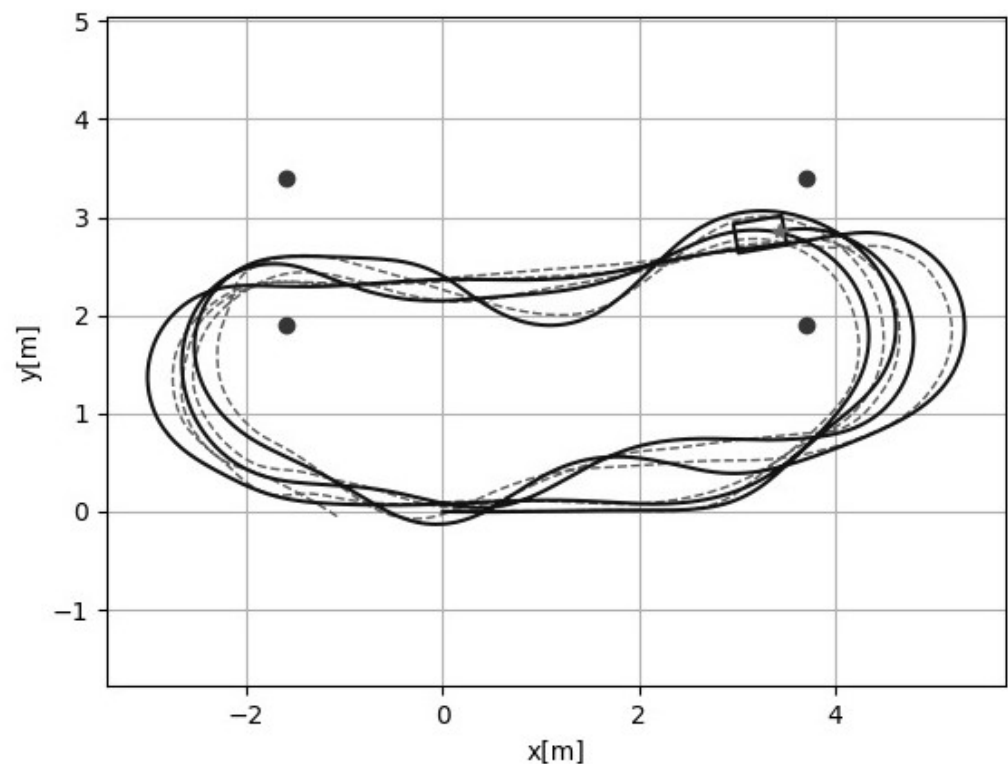
$$J = \sum_{i=1}^L w_x (r_i - x_i)^2 + \sum_{i=1}^L w_u \Delta u_i^2, \quad (7)$$

where  $w_x$  and  $w_u$  are weights,  $r_i$  is a desired trajectory,  $x_i$  is a manipulated variable,  $u_i$  is a control signal,  $L$  is a control horizon.

The goal of predictive control lies in sequential search for optimal control strategy for a certain time interval that is called control horizon. Each time the first step of the strategy found is the only one applied. This step completed, the prediction horizon is being shifted and new optimization starts.

Online optimization taking place at every step has become possible due to the availability of high-performance computing facilities embedded on the vehicle's board.

For the control object model (1) and the prediction horizon value equal to 8, the desired and obtained trajectories are shown in Figure 6.



**Figure 6.** The movement along the trajectory with the help of internal model and model prediction control.

## 5. Results

To study the approaches proposed, an environment for unmanned vehicles simulation based on ROS1 and ROS2 operating systems, has been developed.

Experimental comparison of control methods that apply internal model for real-time control was conducted, with the problem of unmanned vehicle trajectory tracking served as experimental domain. The control was carried out on the basis of P-controller and model predictive control (MPC). The results of field experiments are shown in Table 1. The difference between the length of the desired trajectory  $S_1$  and the length of the path  $S_2$  covered under different velocity  $v$  was used as a comparison criterion. The desired trajectory length was 67.4 m.

**Table 1.** Results of field experiments.

Experiment Number	Speed $v$ , m/s	Deviation When Using P-Controller $S_1$ , m	Deviation When Using MPC $S_2$ , m
1	1.41	4.4	3.5
2	1.12	2.1	1.9
3	0.88	1.6	1.7

According to the experiments the following conclusions may be performed:

1. Control based on P-controller is distinguished by the possibility of using relatively simple controllers;
2. With both control methods being used at high vehicle speeds, the deviation from the trajectory increases significantly;
3. High speeds can be achieved if controller parameters are fine-tuned or in case of applying an intelligent P-controller that adapts to the speeds and curvature of the trajectory;
4. High-performance computers are needed for MPC.

**Author Contributions:** Conceptualization, I.P. and E.S.; methodology, I.P.; software, I.P.; validation, I.P. and V.M.; formal analysis, E.S.; investigation, I.P.; data curation, I.P.; writing—original draft preparation, I.P., E.S. and V.M.; writing—review and editing, E.S.; visualization, V.M.; supervision, I.P. 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 experimental data for both methods is available at <https://cloud.mail.ru/public/raik/h6vjgsMkR> (accessed on 19 June 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ljung, L. *System Identification*; Prentice Hall PTR: Hoboken, NJ, USA, 1999.
2. Pupkov, K.A.; Egupov, N.D.; Trofimov, A.I. *Statisticheskie Metody Analiza, Sinteza i Identifikacii Nelinejnyh Sistem Avtomaticheskogo Upravleniya*; BMSTU: Moscow, Russia, 1998. (In Russian)
3. Liu, G.P. *Nonlinear Identification and Control: A Neural Network Approach*; Springer: Berlin/Heidelberg, Germany, 2012.
4. Paden, B.; Cáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [CrossRef]
5. Mehta, B.R.; Reddy, Y.J. Advanced process control systems. In *Industrial Process Automation Systems*; Mehta, B.R., Reddy, Y.J., Eds.; Butterworth-Heinemann: Oxford, UK, 2015; pp. 547–557. [CrossRef]
6. Huang, B.; Kadali, R. *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*; Springer: Berlin/Heidelberg, Germany, 2008; 241p.
7. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980v8.
8. Diveev, A.; Sofronova, E. The Synthesis of Optimal Control System by the Network Operator Method. In *IFAC Proceedings Volumes*; Elsevier: Amsterdam, The Netherlands, 2009; Volume 42, pp. 232–237.

**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.