

# Improving Industrial Robot Positioning Accuracy to the Microscale Using Machine Learning Method

Vytautas Bucinskas, Andrius Dzedzickis \*, Marius Sumanas, Ernestas Sutinyš, Sigitas Petkevicius, Jurate Butkiene, Darius Virzonis and Inga Morkvenaite-Vilkonciene

Department of Mechatronics, Robotics and Digital Manufacturing, Vilnius Gediminas Technical University, 10223 Vilnius, Lithuania

\* Correspondence: andrius.dzedzickis@vilniustech.lt

**Abstract:** Positioning accuracy in robotics is a key issue for the manufacturing process. One of the possible ways to achieve high accuracy is the implementation of machine learning (ML), which allows robots to learn from their own practical experience and find the best way to perform the prescribed operation. Usually, accuracy improvement methods cover the generation of a positioning error map for the whole robot workspace, providing corresponding correction models. However, most practical cases require extremely high positioning accuracy only at a few essential points on the trajectory. This paper provides a methodology for the online deep Q-learning-based approach intended to increase positioning accuracy at key points by analyzing experimentally predetermined robot properties and their impact on overall accuracy. Using the KUKA-YouBot robot as a test system, we perform accuracy measurement experiments in the following three axes: (i) after a long operational break, (ii) using different loads, and (iii) at different speeds. To use this data for ML, the relationships between the robot's operating time from switching on, load, and positioning accuracy are defined. In addition, the gripper vibrations are evaluated when the robot arm moves at various speeds in vertical and horizontal planes. It is found that the robot's degrees of freedom (DOFs) clearances are significantly influenced by operational heat, which affects its static and dynamic accuracy. Implementation of the proposed ML-based compensation method resulted in a positioning error decrease at the trajectory key points by more than 30%.

**Keywords:** online machine learning; deep-q learning; positioning accuracy; industrial robot; vibrations

**Citation:** Bucinskas, V.; Dzedzickis, A.; Sumanas, M.; Sutinyš, E.; Petkevicius, S.; Butkiene, J.; Virzonis, D.; Morkvenaite, I. Improving Industrial Robot Positioning Accuracy to the Microscale Using Machine Learning Method. *Machines* **2022**, *10*, 940. <https://doi.org/10.3390/machines10100940>

Academic Editors: Raul D.S.G. Campilho, Francisco J. G. Silva and Dan Zhang

Received: 11 August 2022

Accepted: 13 October 2022

Published: 17 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

Robots are used in various processes, including manufacturing, entertainment, services, and scientific research. To maintain a technical edge and thereby remain competitive, more and more businesses are applying advanced technology and programming solutions to their operational processes [1]. Such a wide application encourages the development of universal robotic systems and requires research of their capabilities and performance characteristics.

In general, industrial robots provide high-level static and dynamic positioning accuracy. Nevertheless, they must be maintained to ensure that they continue to meet the conditions for which they have been programmed and in which they operate [2]. Therefore, for each specific task, it is important to determine the following: (i) positioning accuracy (positioning error between stated and the real position of arm end effector); (ii) repeatability (positioning error between real positions of arm end effector performing repeating movements); (iii) other parameters, which are considered to be unique characteristics of the particular machine [3]. Positioning accuracy depends on a number of actions including, but not restricted to the following: (i) the parameters of the drives guiding the robot's movements; (ii) tolerances in manufacturing parts of the machinery; (iii) tolerances due

to the articulation of the robot's chains [4]; (iv) control algorithms [5,6]; (v) dynamic properties of the mobile platform [7,8], and (vi) robot arm properties [9,10]. Each of the factors may become important for the accuracy of the robot, depending on its type, lifting capacity, and operational conditions [4]. Due to the complex nature of robots, the increasing positioning accuracy of robotic installations develops challenging tasks. The positioning errors in offline mode can be determined using special algorithms [11] and mathematical models [12–15]. Common positioning accuracy problems in path and trajectory planning can be resolved by improving offline programming software when the code for a robot is generated automatically or by implementing online feedback [16–20]. However, typical compensation methods are limited by the chaotic nature of the positioning errors. Therefore, more adaptive methods, such as machine learning (ML), can increase positioning accuracy by compensating for positioning errors in particular cases [21,22]. Advances in ML with respect to simple error compensation are in the accountancy of feature-chaotic robot error distribution about positioning points. ML ensures respect for robot fluctuation and fits into the prescribed tolerance field. There are a few cases [23–26] of ML approaches compensating for the positioning errors. Chen and Zhang developed an ML kinematics-based positioning error compensation method for high-precision mechanical machining operations [26]. This positioning error compensation method provided good results, but it contains sophisticated procedures due to its complexity and use of excessive data. More to say, it combines the following: analytical modeling; extended Kalman filtering; spatial interpolation algorithm; an adaptive mesh division algorithm, and an inverse distance weighted interpolation algorithm. To use it at a few trajectory points, this method becomes too costly.

Moreover, ML could be implemented to improve the trajectory-planning process [27–29] or reduce vibrations [30,31].

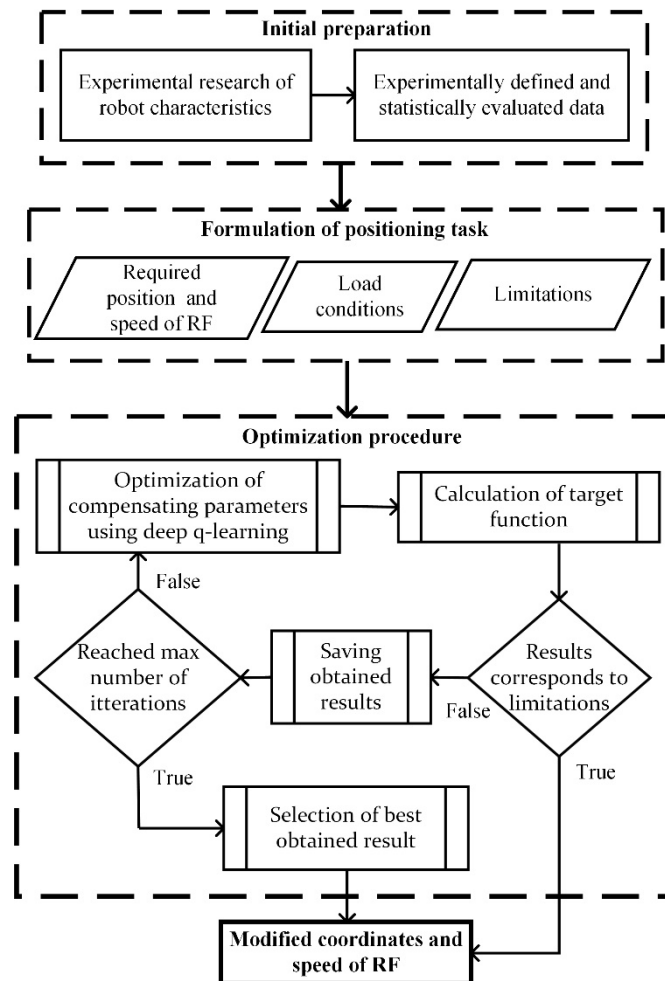
The main aim of our research is to create a methodology to increase industrial robot positioning accuracy and minimize robot end-of-arm vibrations by applying ML in online mode. This methodology will be used to compensate for positioning errors by shifting the destination point or altering the moving velocity of the end-of-arm reference point before defining the forward kinematic task. The method is focused on destination point approach accuracy in defined arbitrary robot positions. Moreover, our proposed approach suits well for industrial robots with typically closed controllers since position correction is performed externally with respect to the robot controller; thus, modified motion commands to target coordinates are processed in a standard way.

## 2. Concept of Research

This paper focused on creating a universal methodology for most types of industrial robots, which can increase positioning accuracy at the robot trajectory endpoints. One of the pillars of ML is mathematical optimization, which involves the numerical computation of parameters for a system designed to make decisions based on unseen data [32–34]. While the ML procedure is based on existing data collection, such a method all the time remains retrospective and reflects errors in previous applications with corresponding load cases. The use of the robot for precise machining or assembly operations requires data about the vibration level and settling time. This will be critically important in addition to static positioning error values and directions. The ML procedure enables a decrease in positioning error values, and end-of-arm vibration influences the resulting accuracy. Nevertheless, the robot workspace mapping procedure, which provides error values and direction at any point of the workspace, all the time remains inaccurate for each particular case.

It is possible to calculate optimal parameters for a given learning problem using currently available data [35]. The collection of data required for ML is a complex process, defined by the robot design, its characteristics, and the aim of ML implementation.

The procedure of implementing our proposed ML method is divided into the following three phases: (i) initial preparation, (ii) positioning task formulation, and (iii) optimization (Figure 1).



**Figure 1.** Simplified block diagram of positioning accuracy increase procedure by implementing ML method. RF—reference point.

The success of the ML procedure depends on the quantity and quality of data generated during initial preparation. This phase includes experiments for the definition of the main robot's characteristics, such as positioning accuracy, reference point vibration level, statistical evaluation, and analysis of obtained data. The initial preparation procedure should be performed for all robot trajectory points of interest. This procedure must be repeated in cases of mechanical wear, change of tool configuration, or essential variation of environmental conditions.

The obtained data shows the technical conditions of the robot since these data consist of a set of dependencies between positioning errors, vibration level, reference point position, operating time, robot speed, and load. In the general case, these dependencies can be expressed as follows:

$$\begin{cases} \Delta_{x,y,z} = F(x, y, z, V, M, t_{op}) \\ \varepsilon_{x,y,z} = F(x, y, z, V, M) \end{cases} \quad (1)$$

where:  $\Delta_{x,y,z}$  is the overall positioning error,  $\varepsilon_{x,y,z}$  is the overall level of reference point vibrations,  $x, y, z$  are coordinates of robot reference point,  $V$  is the speed of reference point,  $M$  is load,  $t_{op}$  is the time from the moment when a robot was switched on.

The second phase of our proposed ML method is the Formulation of the positioning task. In this phase, it is necessary to define the required position and speed of the reference point ( $x, y, z, V$ ) and load ( $M$ ). Moreover, it is necessary to specify the following limitations: acceptable overall positioning error  $\Delta_{min}$ , and overall vibration level of reference point  $\varepsilon_{min}$ . In general, case limitations can be identified as follows:

$$\begin{cases} \Delta_{x,y,z} \leq \Delta_{min} \\ \varepsilon_{x,y,z} \leq \varepsilon_{min} \end{cases} \quad (2)$$

Moreover, to ensure stable operation of the ML algorithm and avoid looping in the algorithm (when the desired accuracy cannot be achieved), it is necessary to include a parameter specifying the maximum number of iterations  $k_{max}$ .

The third phase is the Optimization procedure, which contains the cycle during which the optimization algorithm is used to obtain the most suitable compensation parameters in order to achieve the required positioning accuracy and an acceptable level of reference point vibrations.

The ML algorithm runs with each new positioning task; ML selects compensation parameters and predicts the expected positioning error  $\Delta_{x,y,z}$ , and reference point vibration level. This prediction is based on experimentally defined data obtained during the Initial preparation phase. After defining the expected positioning error  $\Delta_{x,y,z}$  and reference point vibration level  $\varepsilon_{x,y,z}$ , their values are compared with the limitations. If positioning error and reference point vibration level correspond to the limitations of Equation (2), then modified coordinates of the required reference point position and speed are transferred to the robot controller. In case if results do not correspond to the limitations, the data is saved and checked if the maximum number of iterations  $k_{max}$  are reached. The optimization of compensation parameters is performed many times until the limit of iterations is reached. In case if the maximum number of iterations is not achieved, the program selects the best result from all previous iterations.

#### Deep q-Learning Algorithm

The deep Q-learning algorithm was implemented to realize the mentioned ML procedure. The advantage of this algorithm in comparison to others is emphasized due to its efficiency in similar problems and its relatively simple implementation [36–38]. The deep Q-learning algorithm combines the Q-learning algorithm and a deep artificial neural network. The idea of Q-learning is based on the perception of the environment and state to take respective actions in order to achieve the maximum reward. A neural network enables the algorithm to operate in a much larger environment and optimize calculations procedure by enabling approximation features. It allows the algorithm to observe the pattern in the environment and discover the optimal sequences of actions instead of calculating and evaluating each state and the value of each action at each point in the environment. The same principle was used to optimize the definition of the algorithm loss function. We used the stochastic gradient descent method (SGD) [35] to define the function by the theoretical gradient calculated from randomly selected data points instead of calculating the actual gradient from the entire dataset. Such an approach diminishes the dependency of obtained data from strong dataset influence on functional distribution. Other parameters of the implemented algorithm were selected experimentally by evaluating their impact on positioning accuracy after many trials in a separate study. The configuration of the algorithm used in this study is provided in Table 1.

**Table 1.** Parameters of the implemented algorithm.

| Parameter | Activation Function | Optimizer | Hidden Neurons | Hidden Layers | Replay Memory | Temperature |
|-----------|---------------------|-----------|----------------|---------------|---------------|-------------|
| Value     | Softsign            | SGD       | 37             | 1             | 100,000       | 70          |

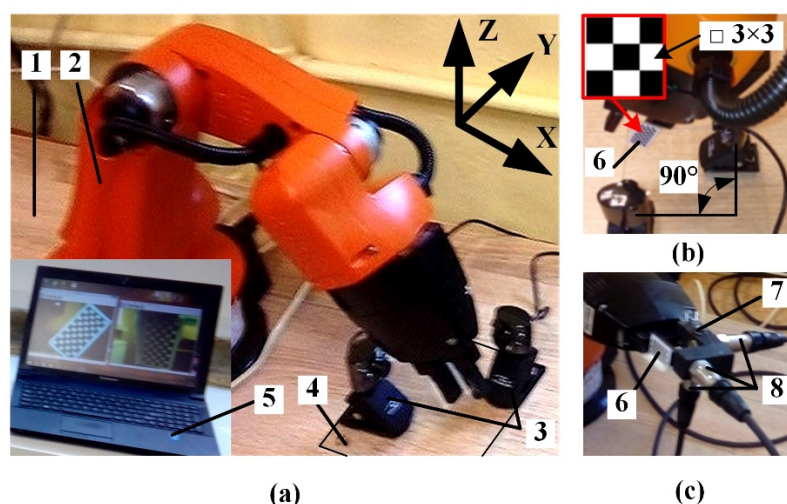
The parameters within Table 1 were defined through the experimental running of the algorithm. Process activation through the Softsign procedure corresponds to the robot learning mode, while there are possibilities for other functions. The activation margin (sensitivity to the activation condition) needs redefinition; the ML process converges correctly and, within 1000 iterations, achieves learning process influence saturation. The implemented optimizer—SGD—corresponds to our purposes and achieves the prescribed result. A number of neurons is chosen according to the desired process parameter resolution, and the number of hidden layers is naturally chosen as one according to the dimensionality of the ML process. The amount of replay memory is defined by the experimental test, and it occurs outside our paper scope as well as the temperature parameter.

Furthermore, this paper provides a detailed methodology for experimental research and data collection for ML.

### 3. Experimental Research

#### 3.1. Experimental Setup

The experiments were performed using the KUKA-YouBot robot (KUKA, Augsburg, Germany) as an industrial robot testbench; it was fixed to a special stable base. The robot's geometric parameters and main characteristics are provided in Table 2 [39]. Positions of the robot's gripper were detected using two USB cameras with a resolution of  $1920 \times 1080$ , a checkerboard matrix of  $4 \times 8$  mm size, and a user-defined function, implementing the detect Checkerboard Points procedure in MatLab (Figure 2). According to [40], the use of such a measurement method can ensure an average measurement deviation better than 0.0004 mm.



**Figure 2.** Experimental setup for measurements of positioning accuracy: (a) general view; (b) position of the cameras; (c) accelerometers mounted on robot gripper. 1—base; 2—robot; 3—USB cameras; 4—based on which are mounted cameras; 5—personal computer; 6—optical checkerboard matrix; 7—holder for cameras; 8—accelerometers.

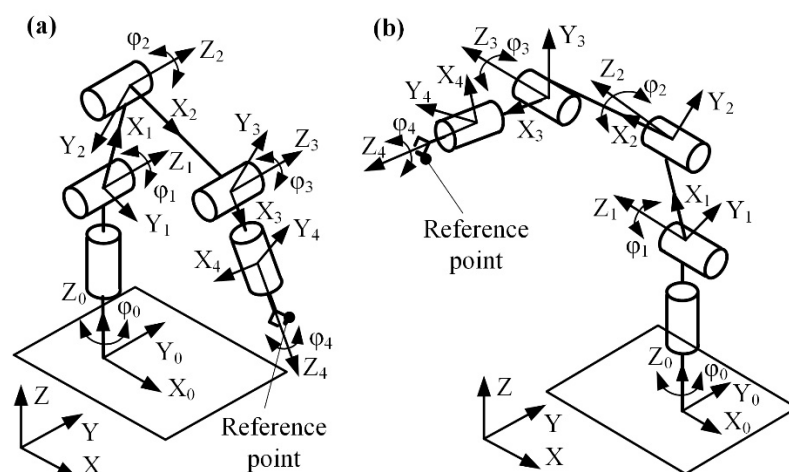
**Table 2.** General characteristics KUKA-YoubBot arm [39].

| General Information |                      | Axes Motion Range, Speed |                    |
|---------------------|----------------------|--------------------------|--------------------|
| Serial kinematics   | 5 axes               | Axis 0 (A0)              | +/-169°, 90°/s     |
| Height              | 655 mm               | Axis 1 (A1)              | +90°/-65°, 90°/s   |
| Work envelope       | 0.513 m <sup>3</sup> | Axis 2 (A2)              | +146°/-151°, 90°/s |
| Weight              | 6.3 kg               | Axis 3 (A3)              | +/-102°, 90°/s     |
| Payload             | 0.5 kg               | Axis 4 (A4)              | +/-167°, 90°/s     |

The cameras were fixed by a special holder positioned at a 90° angle relative to each other to resolve spatial coordinates of the end-of-arm reference point (Figure 2b). The holder with attached cameras was screwed to the same base as the robot. Two special checkerboard patterns (identification marks for cameras) were attached to the gripper (Figure 2b,c).

Absolute vibration accelerations of the end-of-arm reference point were measured using industrial accelerometers Ini 603C01 (PCB Piezotronics, Depew, NY, USA) with a measurement range of 0.5–10,000 Hz and an acceleration limit of 51 g. Accelerometers were attached to the gripper in three perpendicular directions, as shown in Figure 2c. Signals from accelerometers were collected using the data acquisition system USB-4432 (National Instruments, Austin, TX, USA).

Ambient temperature was measured using a digital thermometer MWF-DT-616CT (CEM Corporation, Matthews, NC, USA), with a resolution of 0.1 C. Loads were weighted using Silver crest HG01025 (Silvercrest, Corona, CA, USA) scales, with a resolution of 0.001 kg. For the determination of the required warm-up time, the unloaded robot moves at 50% of the max joint speed (which is 90°/s for all axes) to the trajectory endpoint (Figure 3), rotating joints by the following angles:  $\varphi_0$ —90°,  $\varphi_1$ —15.2°,  $\varphi_2$ —45.7°,  $\varphi_3$ —44.2°,  $\varphi_4$ —20°, and backward. Then the robot returns to the start position, waits for 3 s, and the image of the checkerboard matrix is captured. The experiment trials took place after long operation breaks of 12, 2, and 1 h. In the case of a 2- and 1-h operation break, the warming procedure was applied before the experiment. Firstly, the robot was warmed up using a 30 min warming program and 2- and 1-h operation breaks afterward, correspondingly.

**Figure 3.** Position of the end-of-arm reference point: (a) start (measuring) position; (b) endpoint of the trajectory.

The positioning accuracy measurements were performed in the same conditions, but in this case, experiments were carried out with 10%, 50%, and 100% of the maximum joint speed, without load, and loaded by 0.250 kg and 0.500 kg weights. Each measurement was repeated 20 times.

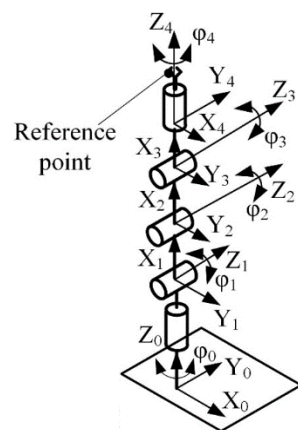
The proposed methodology for the online ML procedure was tested for 2 different robot configurations, imitating the pick-and-place task. In each individual case, using our online ML method, the procedure can be performed at any chosen position in the robot workspace.

### 3.2. Evaluation of the End-of-Arm Reference Point Vibrations

The dynamic parameters of the robot—the end-of-arm reference point vibration level and settling time when the robot stops at the desired positions—were evaluated using accelerometers (Figure 2c). There are the following three parameters for measuring vibrations: displacement, velocity, and acceleration. The evaluation of vibration acceleration provides the best sensitivity in higher frequencies and suits well for evaluating impacts caused by bearing damage, abnormal gears, and noise. Moreover, this method is recommended when it is necessary to evaluate forces and stresses acting on or rotating parts. Moreover, measurements of acceleration do not require reference points. Therefore, it is more convenient for industrial robots.

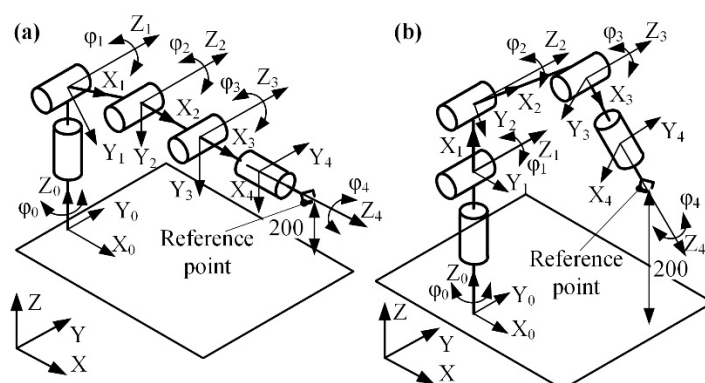
The measurements were carried out at the following three different speeds of the joints of the robot: at 10%, 50%, and 100% of the maximum ( $90^\circ/\text{s}$ ) speed of the joint, using the same robot movement trajectory as shown in Figure 3. The measurement was repeated three times.

The separate DOF effect on the robot end-of-arm vibrations was evaluated by rotating each robot joint separately. For this purpose, joints I–V at coordinates  $\varphi_0$ – $\varphi_4$  were rotated clockwise by a  $60^\circ$  angle at maximum ( $90^\circ/\text{s}$ ) speed from the start position Figure 4. The end-of-arm reference point vibrations were measured after the movement of each joint by  $60^\circ$  at a maximum ( $90^\circ/\text{s}$ ) speed. After 10 s, the joints were rotated back to the start position. Measurements were performed when the robot moved in forward and backward directions, and the procedure was repeated three times.



**Figure 4.** The start position of the robot for the measurements of separate joint impact to robot vibrations.

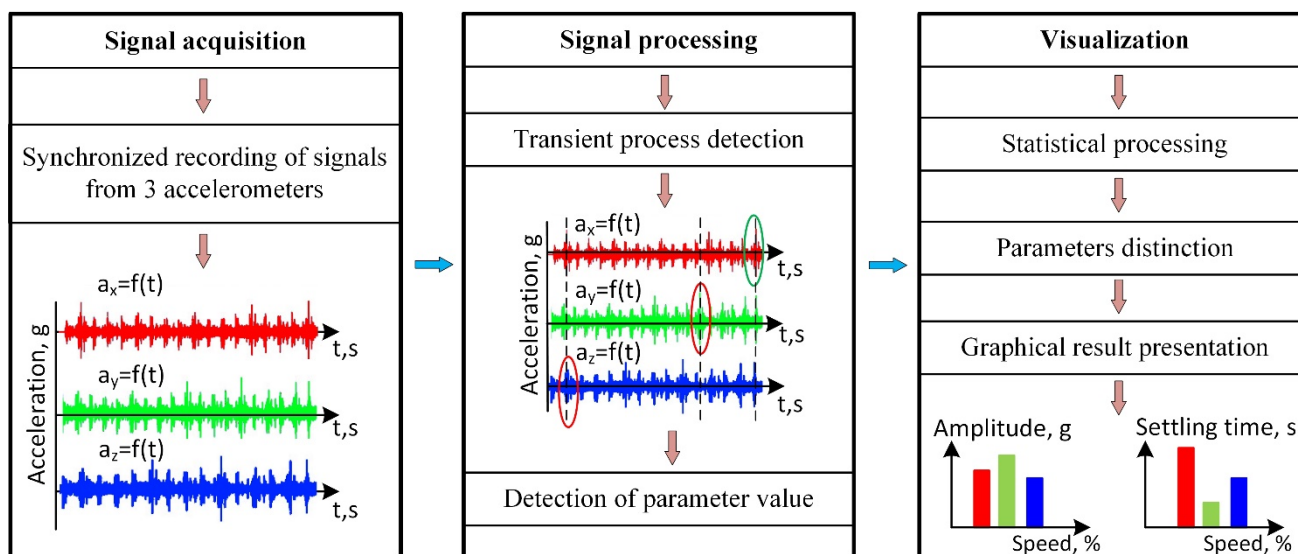
To define the impact of joint I (coordinate  $\varphi_0$ , Figure 5) on robot end-of-arm vibrations, joint I was rotated at  $60^\circ$  from the start position, and the vibrations were measured. The vibrations at the following two start positions were evaluated: (i) fully extended position (Figure 5a); (ii) compacted position (Figure 5b).



**Figure 5.** Start positions of the robot used for the measurements of gripper vibrations: (a) maximum peripheral reach on extended position, (b) compacted position. In both cases, the reference point was distanced from the z-axis origin by 200 mm.

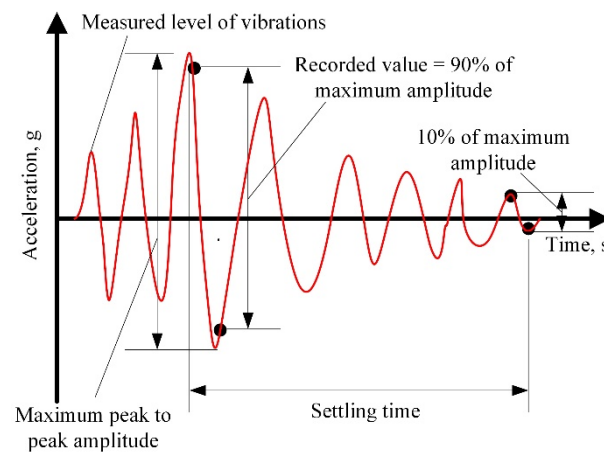
In a fully extended position robot's own weight fully loads joint I. Other joints remain unloaded and stay in a singularity position. The compacted robot position is represented by the situation when joint I is loaded by half of the own weight of other links.

All vibration measurements were performed according to the flowchart provided in Figure 6.



**Figure 6.** Flowchart of the vibration measurement process.

Signals from three accelerometers were recorded continuously during robot movement between the endpoints of the trajectory. Later acquired data were processed offline (Figure 7), defining transient processes due to robot stops at the trajectory endpoints.



**Figure 7.** Definition of vibration parameters from experimental data.

Main parameters (end-of-arm reference point maximum peak-to-peak vibration amplitude, settling time) were extracted from raw data, processed statistically, grouped according to testing conditions, and represented in graphical form.

### 3.3. Calculations

Overall positioning accuracy from measurements in separate axes was evaluated according to the methodology provided by ISO 9283:1998 [41]. It defines positioning accuracy  $AP_p$  as the deviation between a reference position and the mean of attained positions while approaching reference positions from the same direction as follows:

$$AP_p = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2}, \quad (3)$$

where  $x_c$ ,  $y_c$ ,  $z_c$  are coordinates of the prescribed end-of-arm reference point position, and  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$  are mean values of the resulting position.

General robot positioning error consists of accumulated values from all errors, and it is an object of ML-obtained compensation. General error from three-axis measurements for the experiments in 1, 2, and 12 h evaluated using the simplified equation, since target position defined as  $x_c = 0$ ,  $y_c = 0$  and  $z_c = 0$  as follows:

$$AP_p = \sqrt{(\bar{x})^2 + (\bar{y})^2 + (\bar{z})^2}. \quad (4)$$

where  $\bar{x}$ ,  $\bar{y}$ ,  $\bar{z}$  are calculated from experimental data as follows:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j, \quad \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j, \quad \bar{z} = \frac{1}{n} \sum_{j=1}^n z_j, \quad (5)$$

where:  $x_j$ ,  $y_j$ ,  $z_j$  are coordinates of the resulting position, and  $n$  is the measurement cycle number.

Robot positioning repeatability  $RP_l$ , also was evaluated using the methodology proposed in ISO 9283:1998. According to it,  $RP_l$  value is the radius of the sphere that defines the closeness of an attained position after  $n$  attempts to achieve the same position from the same direction as follows:

$$RP_l = \bar{l} + 3S_l, \quad (6)$$

here:

$$\bar{l} = \frac{1}{n} \sum_{j=1}^n l_j; \quad l_j = \sqrt{(\bar{x} - x_j)^2 + (\bar{y} - y_j)^2 + (\bar{z} - z_j)^2}; \quad S_l = \sqrt{\frac{\sum_{j=1}^n (l_j - \bar{l})^2}{n-1}}. \quad (7)$$

Robot positioning repeatability was evaluated using the same experimental data as used for the positioning accuracy measurement; values obtained after the 14th minute of the experiment were used.

### 3.4. Statistical Evaluation of Research Data

The confidence of results for robot gripper vibration level and settling time measurements were evaluated by statistical parameters. Measurement data was assessed by correlation-regression analysis. Arithmetic averages, their standard deviations, and confidence intervals at a 0.95 probability level were calculated according to [42].

In correlation-regression analysis, the difference between the measured result  $m_v$  and real measured parameter value  $r_v$  is defined as absolute measurement error as follows [42]:

$$\Delta m_v = m_v - r_v. \quad (8)$$

The measured parameter  $m_v$  have a prescribed probability, representing the real value of positioning error if the exact measurements are repeated  $n$  times. From several measurements, we calculated the arithmetic mean  $\overline{m}_v$  as follows:

$$\overline{m}_v = \frac{1}{n} \sum_{i=1}^n m_{vi} = \frac{m_{v1} + m_{v2} + \dots + m_{vn}}{n} \quad (9)$$

Absolute measurement error consists of systematic, random, and random errors of deduction. Using correlation-regression analysis, only random errors were estimated in statistical data evaluations. In our case, we evaluated only random errors, as if the methodology and measurement devices are far more accurate than expected error, systematic and accidental deduction errors are not significant and therefore cannot be considered [42]. To evaluate random error, it is necessary to calculate the experimental standard deviation  $\sigma$  of each individual measurement as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (m_v - \overline{m}_v)^2}{n-1}}. \quad (10)$$

The experimental standard mean deviation  $S_{md}$  is calculated by the following:

$$S_{md} = \frac{\sigma}{\sqrt{n}}. \quad (11)$$

The random error of the measured parameter is calculated by the following:

$$\Delta m_{v,n,p} = S_c \cdot S_{md}. \quad (12)$$

where:  $S_c$ —the value of the Student Criterion selected according to the number of experiment variables  $(n - 1)$  and the probability level ( $\alpha = 0.95$ ).

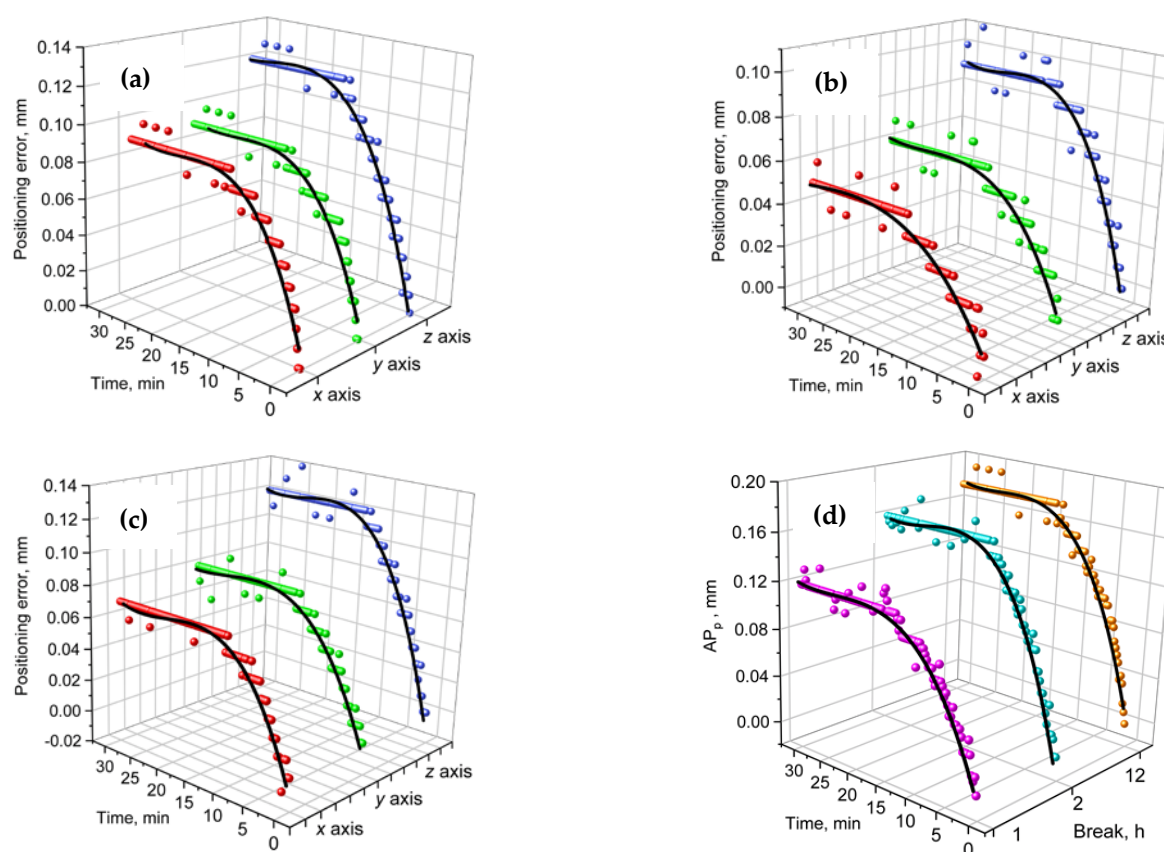
The final result of the measured ( $n$  times) value  $m_v$  is expressed as the sum of the arithmetic mean  $\overline{m}_v$  and the random error  $\Delta m_{v,n,p}$  as follows:

$$\overline{m}_v \pm \Delta m_{v,n,p}. \quad (13)$$

## 4. Results

### 4.1. Robot's Accuracy after a Long Operation Break

The robot's accuracy after a long operation break degenerates because of the temperature regime change and lubricant film disappearance from joint clearances. Thus, the measurement of the positioning error of the robot after a long operation break is important for determining the accuracy of restoring to operating time. To determine the robot's positioning accuracy in time, the positioning error measurement was performed in the  $x$ ,  $y$ , and  $z$ -axes when the robot starts to operate after a 12-h break (Figure 8a). In all axes, positioning error increases in the first 14 min of operation time. Positioning errors in the  $x$  and  $y$ -axes are almost identical as follows: after 14 min, errors are 0.09 mm in both the  $x$  and  $y$ -axis, while in the  $z$ -axis, the error is 0.12 mm.



**Figure 8.** Positioning error dependency on time after: (a) 12 h operation break; (b) 1-h operation break; (c) 2-h operation break; (d) general positioning errors  $AP_p$ , calculated by Equation (4) after 1, 2, and 12 h breaks. Trend lines—regression approximation.

Regression approximation parameters are provided in Table 3.

**Table 3.** Coefficients for the regression estimated robot positioning error.

| Operation Break | Axis   | $a_0$                    | $a_1$                   | $a_2$                    | $a_3$                   |
|-----------------|--------|--------------------------|-------------------------|--------------------------|-------------------------|
| 1-h             | $x$    | $2.8464 \times 10^{-4}$  | $5.5500 \times 10^{-3}$ | $-1.9554 \times 10^{-4}$ | $2.1655 \times 10^{-6}$ |
|                 | $y$    | $1.6700 \times 10^{-3}$  | $7.7700 \times 10^{-3}$ | $-3.3957 \times 10^{-4}$ | $4.9016 \times 10^{-6}$ |
|                 | $z$    | $-4.6000 \times 10^{-3}$ | $1.3590 \times 10^{-2}$ | $-6.3174 \times 10^{-4}$ | $9.5758 \times 10^{-6}$ |
|                 | $AP_p$ | $8.5400 \times 10^{-3}$  | $2.2560 \times 10^{-2}$ | $-1.0000 \times 10^{-3}$ | $1.4723 \times 10^{-5}$ |
| 2-h             | $x$    | $-7.3200 \times 10^{-3}$ | $1.2260 \times 10^{-2}$ | $-6.1872 \times 10^{-4}$ | $9.9370 \times 10^{-6}$ |
|                 | $y$    | $-5.8300 \times 10^{-3}$ | $1.1690 \times 10^{-2}$ | $-5.0854 \times 10^{-4}$ | $7.0670 \times 10^{-6}$ |
|                 | $z$    | $-9.7000 \times 10^{-3}$ | $1.8300 \times 10^{-2}$ | $-8.2850 \times 10^{-4}$ | $1.2126 \times 10^{-6}$ |
|                 | $AP_p$ | $-1.1120 \times 10^{-2}$ | $2.4470 \times 10^{-2}$ | $-1.1200 \times 10^{-3}$ | $1.6540 \times 10^{-5}$ |
| 12-h            | $x$    | $8.0000 \times 10^{-3}$  | $1.2340 \times 10^{-2}$ | $-6.1009 \times 10^{-4}$ | $9.8658 \times 10^{-6}$ |
|                 | $y$    | $7.5200 \times 10^{-3}$  | $1.1750 \times 10^{-2}$ | $-5.5304 \times 10^{-4}$ | $8.5685 \times 10^{-6}$ |
|                 | $z$    | $-2.7500 \times 10^{-3}$ | $1.5180 \times 10^{-2}$ | $-6.0424 \times 10^{-4}$ | $7.8195 \times 10^{-6}$ |
|                 | $AP_p$ | $4.5125 \times 10^{-4}$  | $1.5930 \times 10^{-2}$ | $-6.9624 \times 10^{-4}$ | $9.9989 \times 10^{-6}$ |

Results of experiments performed after a 1-h and 12-h operation break showed that the required warm-up period is 14 min (Figure 8b). However, with a 1-h break, positioning errors decreased compared to a 12-h break, causing the following errors: 0.05 mm in the  $x$ -axis, 0.06 mm in the  $y$ -axis, and 0.09 mm in the  $z$ -axis. This phenomenon can be explained by the fact that 1 h is not enough for all joints to cool down to the initial temperature and to sediment the lubricant layer. The uneven distribution of positioning errors

in the  $x$  and  $y$  axes requires an increase in the operational break. To check this assumption, the same experiments were performed after a 2 h break (Figure 8c).

It was observed that the time until positioning errors become stable is 12.5 min for the  $y$  and  $z$ -axes and 11.2 min for the  $x$ -axis. The stable positioning error in the  $x$ -axis is 0.07 mm, 0.08 mm in the  $y$ -axis, and 0.12 mm in the  $z$ -axis. This situation is quite similar to the case when the robot starts operating after a 12-h break. Small differences can be explained by assuming that after a 2-h break, the robot almost cools down, but some elements in separate joints may still have higher temperatures.

From the dependencies between positioning error fluctuations over time, we can define the minimum required warm-up period for the robot or predict the positioning accuracy in respect of operating time. Moreover, the obtained data allow us to define the robot's overall positioning accuracy and repeatability.

Results presented in Figure 8d correspond to results presented in Figure 8a–c where it is seen that after a lengthy break (12 h), overall positioning precision depends upon time. In the first 13 min of operation after the 1-h break, positioning error slightly increases up to 0.12 mm. After a 2-h break, the error reaches 0.16 mm, and finally 0.17 mm after a 12-h break. This allows us to state that the minimum warm-up time required for the robot to reach stable positioning accuracy values is more than 14 min.

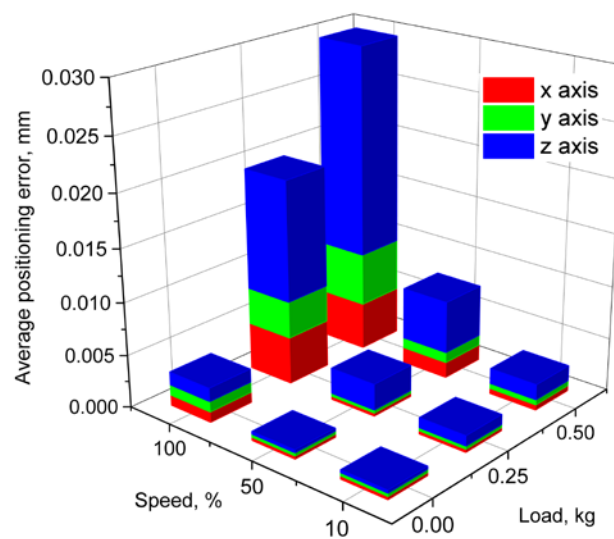
The repeatability error of the robot was specified using experimental data provided in Figure 8d. The overall evaluation of repeatability error after data processing is  $0.0146 \pm 0.00526$  mm (Equations (6) and (7)).

The collected data suggest that, if the robot stays idle for an hour or more, in order to achieve a higher level of accuracy and repeatability, it is recommended that within the first 14 min after the break, do not perform any operational action, but rather use some warm-up programs. In other cases, we can provide coordinates of the prescribed position, which should be adjusted according to the dependencies obtained during this research. The value and direction of defined positioning error become important parameters for compensation used in ML-based robot trajectory correction. The ML process will use the value and direction of the defined positioning error as parameters and use them as the target line for further process of ML. The orientation of the error vector also has meaning for error compensation. In the case of a simulation procedure rather than an experimental ML approach, there is a chance to significantly decrease learning time. Obtained data will have a value not only as an absolute achieved point but also as a beacon for further learning, especially when another loading configuration or load occurs.

We observed the positioning error dependencies over time and after a long operation break. These dependencies can be included in ML algorithms to compensate for positioning errors at different periods of the warming process operating times.

#### 4.2. Definition of Robot Positioning Accuracy

The accuracy of robot positioning was examined when the robot was moved at various speeds with different loads. Positioning errors were measured in the following three perpendicular directions:  $x$ ,  $y$ , and  $z$  (Figure 9). The average distribution of positioning errors was calculated using Equation (12) when the robot moves at the speed values of 10%, 50%, and 100% of maximum speed for every joint being unloaded or with 0.25 kg and 0.50 kg loads.



**Figure 9.** Positioning errors of the robot when it operates at various speeds and loads.

It was defined that in all researched cases, the highest positioning error value appears along the z-axis operating at maximum speed ( $90^\circ/\text{s}$ ). When the robot moves in unloaded mode, the average positioning error in the z-axis is 0.0014 mm. When the robot moves with a 0.50 kg load, this error increases up to 0.02 mm. A similar tendency is noticed when analyzing the accuracy of the x and y-axis, but in those cases, when the values of positioning errors are lower compared with values of the z-axis as follows: 0.001 mm when the robot moves unloaded and 0.005 mm when the robot moves with a 0.50 kg load.

The results of the measurements allow us to state that if the robot is warmed up, positioning accuracy of  $\pm 0.03$  mm can be achieved with a maximum (0.50 kg) weight at the highest speed ( $90^\circ/\text{s}$ ). At lower loads or at lower speeds, it is possible to achieve a positioning accuracy of up to  $\pm 0.01$  mm (Figure 9).

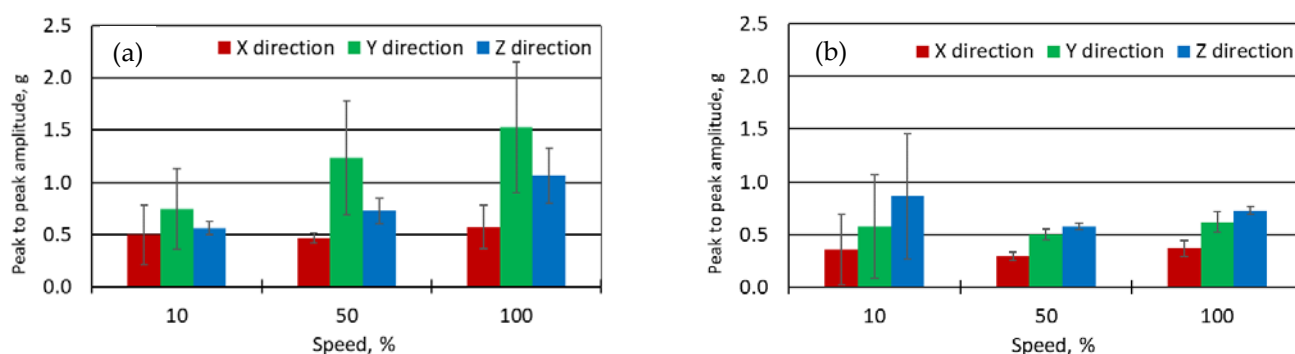
Results obtained from accuracy and repeatability measurements show that at desired conditions, the robot achieves better parameters compared to the ones declared by the manufacturer [39]. The achieved values of positioning error with prescribed load and speed let us compensate key-point coordinates through ML procedure and achieve actual positioning accuracy higher than guaranteed by the robot producer.

Moreover, ML can compensate for the dynamic error of the robot, but for this purpose, it is necessary to analyze the actual vibrations during the robot's movement. Measurements of robot vibrations allow us to fulfill information about positioning accuracy by evaluating such parameters as maximum vibration level and settling time when the robot stops at the desired position. The results of these researches are presented in the next sub-chapter. The error values are derived from experimental research as error ranges, and the direction of error known as well; therefore, the compensation is performed in advance for defined load and robot configuration.

#### 4.3. Measurement of the Robot Vibrations

Data obtained from robot vibration measurements allowed us to define the average vibration level of the end-of-arm reference point and settling time. These characteristics describe the robot's dynamics and have a great influence on its accuracy and operating speed. In addition, these characteristics allow indirect evaluation of position overshoots and show the minimum required time to reach a stable position after movement. This is extremely important in robotics for the assembly of precise objects and the cooperation of several robots.

Statistically evaluated results show the average level of the reference point absolute vibration acceleration values (Figure 10).

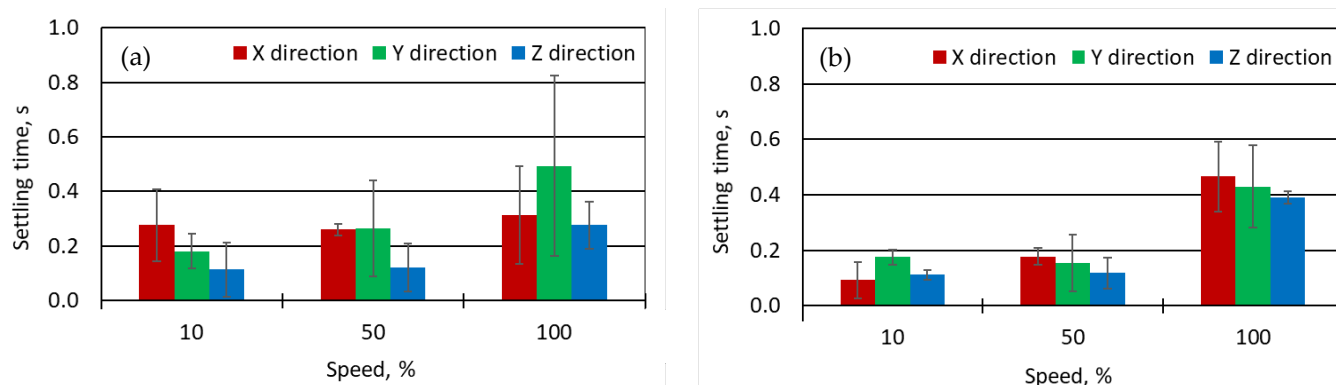


**Figure 10.** Peak-to-peak vibrations amplitudes: (a) at the start (measuring) position (Figure 3a), (b) at the endpoint of the trajectory (Figure 3b). Statistical evaluation was performed according to Equations (8)–(13).

From Figure 10a, it is seen that the average amplitude of the vibrations increases when the speed of the robot increases practically on all measured axes. The highest values of vibration levels were recorded at the highest speed. Comparing vibration levels in separate axes, we noticed that the highest values (0.75 g at speed 10% and 1.6 g at speed 100%) were defined on the *y*-axis.

Results presented in Figure 10b show the following inverse tendency: here, the highest values of vibration level occur when the robot moves at 10% speed. By analyzing the vibration level in the *z*-axis direction, it was 0.87 g at 10% speed, 0.6 g at 50% speed, and 0.75 g at 100% speed. A similar tendency was also defined by analyzing the *x* and *y*-axes.

Controversial results regarding Figure 10a,b can be explained by the fact that when the robot moves up from the start position, the gravitational force acts as an additional load; when the robot moves down from the endpoint of the trajectory, gravity acts in the same direction as robot displacement (Figure 2). This analysis was useful to evaluate the settling time of vibrations (Figure 11).



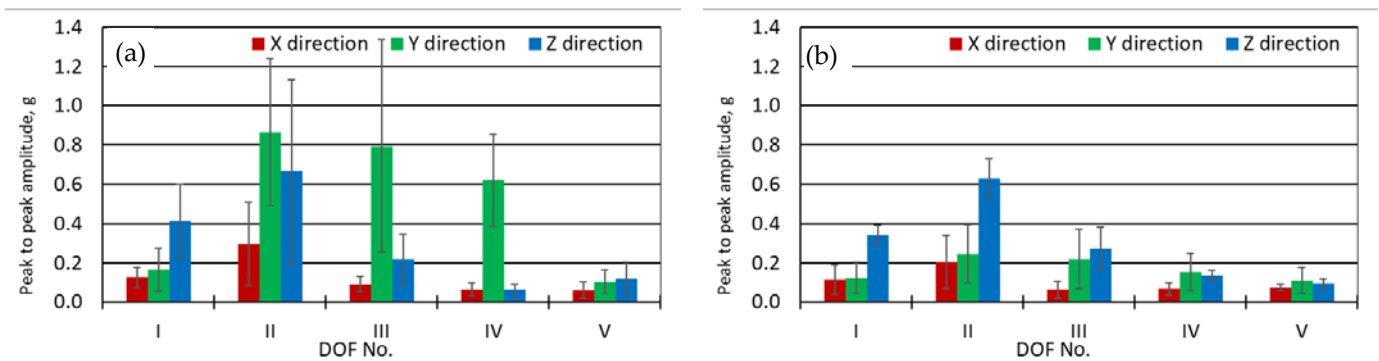
**Figure 11.** Settling time: (a) at the start (measuring) position (Figure 3a) and (b)—at the endpoint of the trajectory (Figure 3b). Statistical evaluation processed according to Equations (8)–(13).

Settling time for all axes increases with movement speed; however, the resulting settling time of the *x*-axis at 50% speed does not follow this tendency (Figure 11a). At the ends of the trajectory (Figure 3), the lowest defined settling time is 0.1 s, whilst the highest value is equal to 0.5 s. Comparing the results presented in Figure 11a,b, the impact of the direction of gravity on the direction of the robot's movement is demonstrated. Settling time in the case when the robot moves from its starting point is about 0.07 s. If the robot moves from its trajectory endpoint, it does not exceed 0.2 s. Results presented in Figures 10 and 11 allow us to state that the vibration level when the robot stops depends not only

on the movement speed but also on the movement direction. The results of all experimental research revealed the settling time did not exceed 0.5 s.

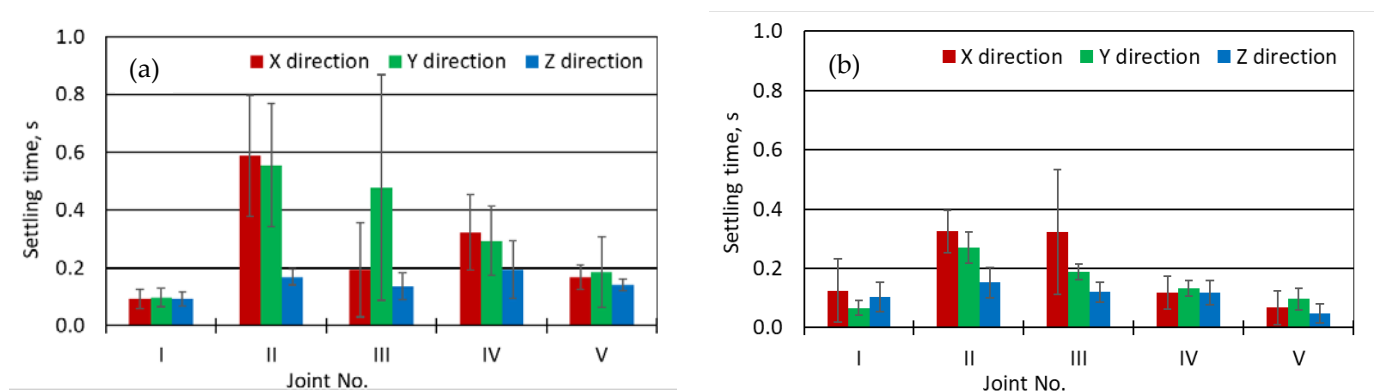
#### 4.4. Separate Joint Vibrations Evaluation

To evaluate the behavior of each joint, we performed a separate experiment, which allows defining the influence of each joint on the general vibrations of the entire robot (Figure 12).



**Figure 12.** Peak-to-peak vibrations amplitudes evaluated after rotation of separate joints in (a) clockwise direction; (b) anti-clockwise direction. Statistical evaluation processed according to Equations (8)–(13). Robot start position marked in Figure 3a.

Peak-to-peak vibration amplitudes were evaluated after the rotation of separate joints (Figure 4). The largest measured vibration amplitude was recorded at the  $x$  and  $y$  axes. The largest amplitude of all measurements was recorded when rotating joint II in a clockwise direction (0.6 g), and the smallest amplitude was recorded when rotating joint V (in a counterclockwise direction, 0.05 g). Probably some variations in clockwise and counterclockwise directions were produced by gravity, which was affecting the robot's movement. Results of measured settling time when separate joints were moved are presented in Figure 13.



**Figure 13.** Settling time evaluated after rotation of separate joints in: (a) clockwise direction; (b) counterclockwise direction. Statistical evaluation was performed by Equations (8)–(13). Robots start position is shown in Figure 3a.

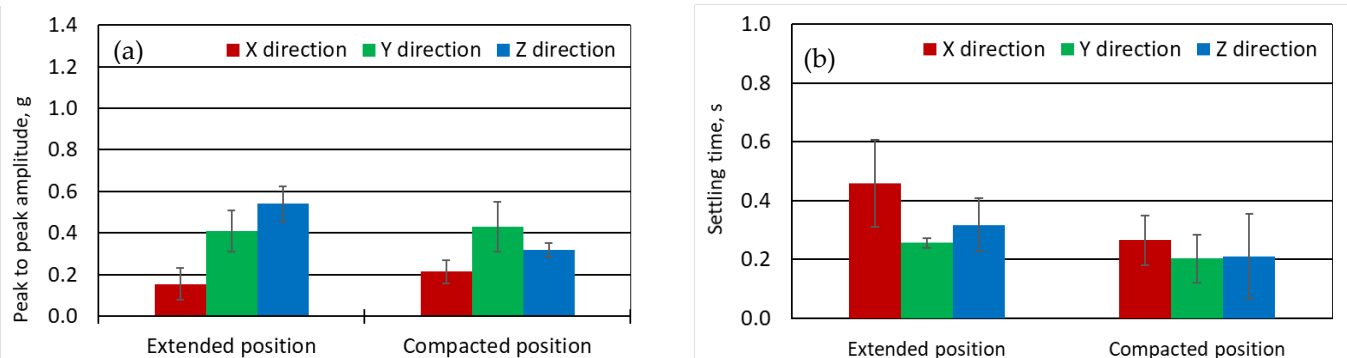
From Figure 13a, it is seen that settling time had the greatest value in the  $x$  and  $y$  directions when joint II was rotated. When the robot joints move clockwise (Figure 13b), the longest settling time was recorded when rotating the second and third joints. The shortest settling time was detected while moving joint I and joint V in both cases. Results obtained from this experiment show that dynamic robot characteristics are mostly affected

by the characteristics of the second and third joints, and the settling time at various positions can be more than 0.5 s.

#### 4.5. Extended Position Vibration Measurement

The impact of the first joint characteristics on the robot vibrations when this joint is loaded by maximum force due to an extended robot arm was evaluated (Figure 5). Under such conditions, the maximum vibration level is expected to be influenced by the gravity force direction.

From Figure 14, the highest level of vibration, 0.46 g, was detected in the  $x$ -axis when the robot is at an extended position; in the compact position case, this value was 0.27 g. The lower levels of vibrations were detected in the  $y$  and  $z$ -axes. The moderately high level of vibrations in the  $z$ -axis allows us to state that during the movement of the first joint, the whole structure of the robot is kinematically excited. In the extended position, the longest settling time of 0.56 s was defined in the  $z$ -axis, while in the compact position, the longest settling time of 0.44 s was detected in the  $y$ -axis. This can be explained by the fact that in various configurations, due to the distribution of mass centers, the variable loads and inertia forces in all joints can be detected. Thus, it could be concluded that in most uncomfortable configurations, settling time will not exceed 0.6 s.



**Figure 14.** Influence of the robot positions (extended position and compact position) in to (a) peak-to-peak vibrations amplitudes; (b) settling times. Statistical evaluation was performed by Equations (8)–(13).

#### 4.6. Experiment with Implemented Correction

The correction was applied to the real (Kuka YouBot) robot using processed experimental data. The flowchart of the performed procedure is presented in Figure 15. During the first run, algorithms take experimentally defined data representing robot characteristics and previous compensation values as input parameters and provide an output as a natural number that is later transformed into a new compensation value.

The compensation parameters were applied by shifting the stated start and endpoint positions of the trajectory provided in Figure 3. The efficiency of the proposed methodology was evaluated by measuring positioning accuracy at the start and endpoint of the robot trajectory (Figure 16). During the experiment, the robot was loaded with a 0.5 kg payload and moved at maximum velocity.

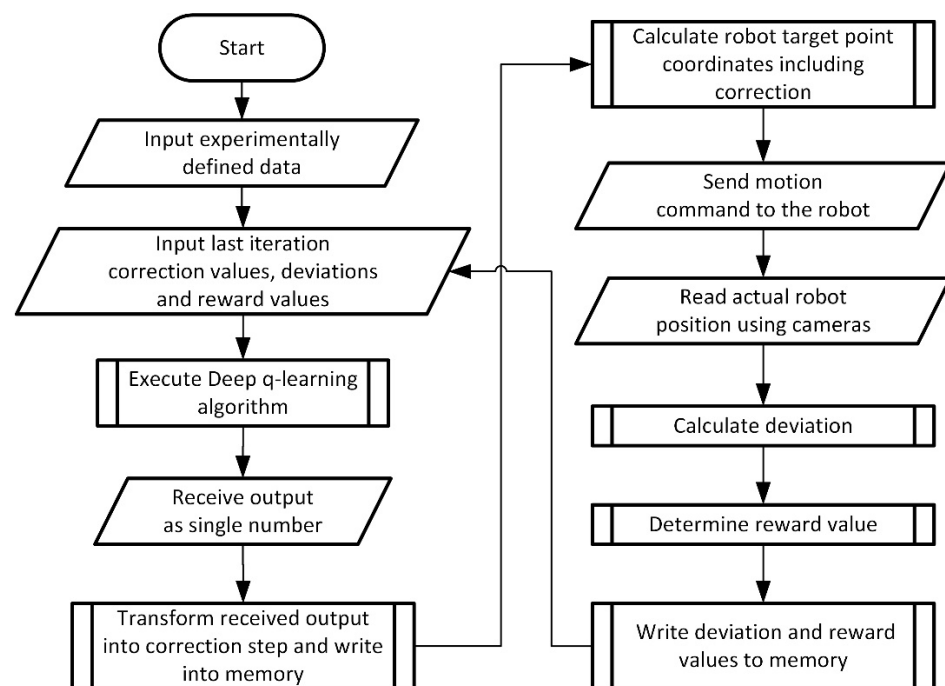


Figure 15. Flowchart of machine learning based positioning errors compensation procedure.

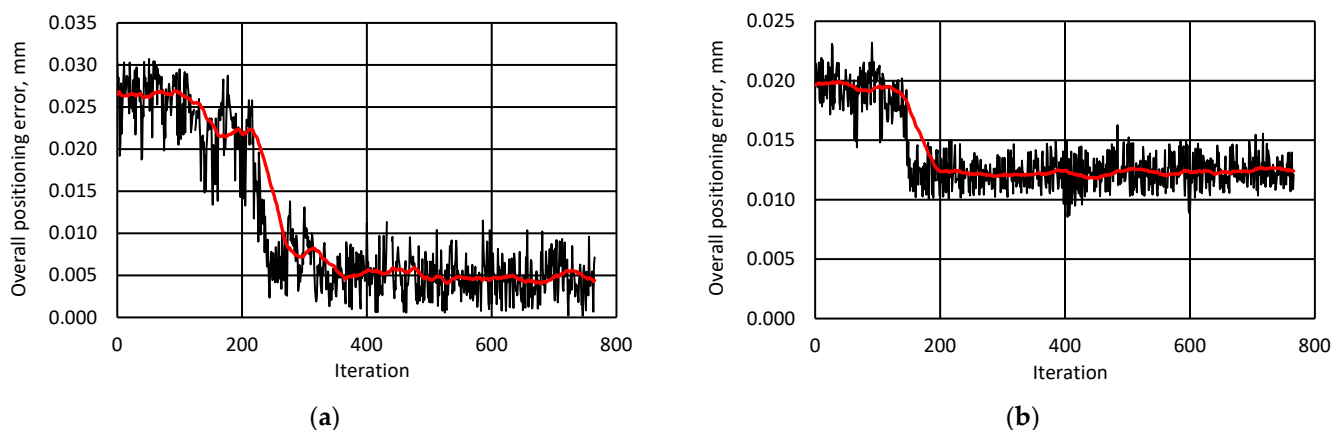


Figure 16. Comparison of robot positioning accuracy before applying correction procedure and after. (a) Experiment at the start point (Figure 3a); (b) experiment at the endpoint (Figure 3b).

The machine learning process was terminated after 780 iterations by the program because the general positioning error stops to decrease. The ML process has different progress for overall positioning error in the start (Figure 3a) and endpoints (Figure 3b) of the trajectory. The learning process at the start point of the trajectory progresses at the 180th iteration, while the process at the endpoint of the trajectory starts approaching the learning target at the 180th iteration and ends at the 240th iteration. Differences in the learning process are influenced by the gravity force depending on the robot links' configuration.

From the obtained results, we defined the following: the overall positioning error at the start point decreases by 40%: from 0.021 mm to 0.0125 mm; the overall positioning error at the endpoint decreases by 34%: from 0.028 mm to 0.0185 mm.

The proposed method deals with the complex improvement of robot accuracy in comparison with classic methods. Comparison of the proposed method to the existing classic ML methodologies has some scale problems. The online ML process, as proposed in the paper, can minimize robot positioning errors at the interesting point by compensating them in the operation online. Available offline learning procedures are focused on preparatory modes of robot's implementation. Offline methods, as well as statistical ones

(workspace mapping), are not sensitive to the fluctuations of the robot operating process. In such a case, robot installation in the production enterprise begs for readjusting for the time being, while the proposed procedure is valid for active robot operation. The complexity of the evaluated and compensated errors is hidden behind the process—we achieve a goal as a diminishing of robot error regardless of the actual coordinate position and configuration of the robot in the target point position. The distribution of the task components along the axes as it is performed for some methods here is not applicable. The disadvantage of the proposed method is some operational time for the ML procedure to set up compensation values, so actual disturbances of the robotic system can cause some periods of increased error values. The good point of the proposed method—initial data collection is possible to optimize. There are available synthetic (simulated) data implementations into the learning process and the use of advanced technical equipment for automated data collection.

## 5. Conclusions

The dynamic behavior of a robot is to a large extent affected by the robot's warming-up time, the load, the speed of movement, and the trajectory in the working space. The experimental research on warming-up time showed that the highest positioning error was 0.12 mm if the robot operated without warm-up. The optimal time for warming up is 14–15 min. The completed research allowed a conclusion that when the manipulator stays idle for 1 h or more, it is recommended to allow the manipulator to operate initially without performing any work. Where the use of a warm-up program is inconvenient, higher positioning accuracy can be achieved by adjusting the prescribed positions according to experimentally defined dependencies between positioning errors and operating time. Implementing ML here could compensate for positioning errors during the robot warming period if the learning procedure can be taken for many operation breaks. Robot repeatability measured after the warming-up process was  $0.0146 \pm 0.00526$  mm.

Research performed when the robot moves at various speeds with various loads showed that the highest 0.019 mm positioning error appeared in the z-axis when the robot moves at its highest speed with a 0.50 kg load. The obtained results significantly exceed the tolerance of  $\pm 1.0$  mm declared by the manufacturer. This allows us to state that ML can improve the accuracy of industrial robots with known operating and environmental conditions. ML is suitable for positioning static and dynamic error compensation for steady-state trajectory and fast accuracy improvement during initial or test operation. The obtained accuracy error value is better than that declared by the manufacturer, and it can even be improved for precision operations; the manufacturer could also implement this option using ML procedures.

Results obtained from vibration measurements show that gripper vibration amplitude mostly depends on the movement trajectory, speed, and direction; the highest measured peak-to-peak vibration amplitude, 1.6 g, was observed at maximum (90°/s) speed. Additional research performed with separate joints showed that the highest impact on the overall vibration level is on joint II and joint III if the robot operates in common configurations. When the robot operates in the closed position, overall vibration levels are mostly affected by joint I.

The results of the settling time measurements, in most cases, correspond to the results of vibration measurements. It was noticed that settling time in most of the tests depends on movement speed, while vibration amplitude is additionally affected by movement direction. The longest settling time was defined as 0.55 s at maximum speed. Therefore, this allows us to state that when operating at maximum speed, the settling time for the robot can last up to 0.6 s.

The set of results obtained from our research allows us to predict the behavior of the analyzed robot and to define its characteristics under various conditions. These results can be used to improve robot parameters for precise operations. Moreover, this result and methodology can be used as a basic database for creating and implementing in practice

various ML algorithms. To compensate for positioning errors, the data about the current robot operation must be collected and processed in a way suitable to further use in ML. First, positioning accuracy and vibrations under various conditions in positions that are important for the performing task and in which the accuracy is diminished (for example, near the singularity) should be determined experimentally. Second, the measurements of vibrations on the end-of-arm reference point, including measurements of separate joint impact to general vibration level, should be performed. Then the collected data can be processed using ML algorithms.

**Author Contributions:** Conceptualization, V.B., A.D. and D.V.; methodology, M.S. and I.M.-V.; software, S.P. and E.S.; validation, M.S., I.M.-V. and A.D.; formal analysis, I.M.-V. and E.S.; investigation, M.S.; resources, M.S.; data curation, D.V.; writing—original draft preparation, A.D., I.M.-V. and J.B.; writing—review and editing, J.B., A.D. and I.M.-V.; visualization, A.D. and S.P.; supervision, V.B.; project administration, M.S.; funding acquisition, V.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is part of the AI4DI project, receiving funding from the Electronic Components and Systems for European Leadership Joint Undertaking in collaboration with the European Union's H2020 Framework Programme (H2020/2014-2020) and National Authorities, under grant agreement n° 826060.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to an internal project consortium agreement.

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

## Nomenclature

|                       |                                                                    |
|-----------------------|--------------------------------------------------------------------|
| $g$                   | Gravitational acceleration equivalent to $\sim 9.81 \text{ m/s}^2$ |
| $\varphi$             | Actual joint rotation angle, deg                                   |
| $AP_p$                | Overall positioning accuracy                                       |
| $RP_i$                | Robot positioning repeatability                                    |
| $m_v$                 | Measured result                                                    |
| $r_v$                 | The real measured parameter value                                  |
| $\Delta m_v$          | Absolute measurement error                                         |
| $\sigma$              | Experimental standard deviation                                    |
| $S_{md}$              | Experimental standard mean deviation                               |
| $\Delta m_{v,n,p}$    | The random error of the measured parameter                         |
| $\Delta_{x,y,z}$      | Overall positioning error                                          |
| $V$                   | Speed of reference point                                           |
| $M$                   | Load                                                               |
| $t_{op}$              | Time from the moment when the robot was switched on                |
| $k_{max}$             | Maximum number of iterations                                       |
| $\varepsilon_{x,y,z}$ | The overall level of reference point vibrations                    |

## References

1. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *J. Ind. Inf. Integration* **2017**, *6*, 1–10. <https://doi.org/10.1016/j.jii.2017.04.005>.
2. Jin, Y.; Chen, I.-M.; Yang, G. Kinematic design of a family of 6-DOF partially decoupled parallel manipulators. *Mech. Mach. Theory* **2009**, *44*, 912–922. <https://doi.org/10.1016/j.mechmachtheory.2008.06.004>.
3. Bhangale, P.; Agrawal, V.; Saha, S. Attribute based specification, comparison and selection of a robot. *Mech. Mach. Theory* **2004**, *39*, 1345–1366. <https://doi.org/10.1016/j.mechmachtheory.2004.05.020>.
4. Park, K.-J. Flexible robot manipulator path design to reduce the endpoint residual vibration under torque constraints. *J. Sound Vib.* **2004**, *275*, 1051–1068. <https://doi.org/10.1016/j.jsv.2003.07.001>.
5. Wu, K.; Krewet, C.; Kuhlentötter, B. Dynamic performance of industrial robot in corner path with CNC controller. *Robot. Comput. Manuf.* **2018**, *54*, 156–161. <https://doi.org/10.1016/j.rcim.2017.11.008>.
6. Sintov, A.; Shapiro, A. Dynamic regrasping by in-hand orienting of grasped objects using non-dexterous robotic grippers. *Robot. Comput. Manuf.* **2018**, *50*, 114–131. <https://doi.org/10.1016/j.rcim.2017.09.009>.

7. Ponce-Hinestroza, A.N.; Castro-Castro, J.A.; Guerrero-Reyes, H.I.; Parra-Vega, V.; Olguin-Diaz, E. Cooperative redundant omnidirectional mobile manipulators: Model-free decentralized integral sliding modes and passive velocity fields. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 June 2016; Volume 2016, pp. 2375–2380. <https://doi.org/10.1109/ICRA.2016.7487387>.
8. Adamov, B.I. Influence of mecanum wheels construction on accuracy of the omnidirectional platform navigation (on example of KUKA youBot robot). In Proceedings of the 25th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2018-Proceedings, Petersburg, Russia, 28–30 May 2018; pp. 1–4. <https://doi.org/10.23919/ICINS.2018.8405889>.
9. Huckaby, J.; Christensen, H.I. *Dynamic Characterization of KUKA Light-Weight Robot Manipulators Technical Report GT-RIM-CR-2012-001*; Georgia Institute of Technology: Atlanta, GA, USA, 2012.
10. Mohamed, K.T.; Ata, A.; El-Souhily, B.M. Dynamic analysis algorithm for a micro-robot for surgical applications. *Int. J. Mech. Mater. Des.* **2011**, *7*, 17–28, <https://doi.org/10.1007/s10999-010-9146-y>.
11. Urrea, C.; Pascal, J. Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot. *Comput. Electr. Eng.* **2018**, *67*, 791–806, <https://doi.org/10.1016/j.compeleceng.2016.09.004>.
12. Abele, E.; Weigold, M.; Rothenbücher, S. Modeling and Identification of an Industrial Robot for Machining Applications. *CIRP Ann.* **2007**, *56*, 387–390, <https://doi.org/10.1016/j.cirp.2007.05.090>.
13. Sharifi, M.; Chen, X.; Pretty, C.; Clucas, D.; Cabon-Lunel, E. Modelling and simulation of a non-holonomic omnidirectional mobile robot for offline programming and system performance analysis. *Simul. Model. Pr. Theory* **2018**, *87*, 155–169, <https://doi.org/10.1016/j.simpat.2018.06.005>.
14. Li, R.; Zhao, Y. Dynamic error compensation for industrial robot based on thermal effect model. *Measurement* **2016**, *88*, 113–120, <https://doi.org/10.1016/J.MEASUREMENT.2016.02.038>.
15. Shang, D.; Li, Y.; Liu, Y.; Cui, S. Research on the Motion Error Analysis and Compensation Strategy of the Delta Robot. *Mathematics* **2019**, *7*, 411, <https://doi.org/10.3390/math7050411>.
16. Borisov, O.I.; Gromov, V.S.; Pyrkin, A.A.; Vedyakov, A.A.; Petranovsky, I.; Bobtsov, A.; Salikhov, V.I. Manipulation Tasks in Robotics Education. *IFAC-Pap.* **2016**, *49*, 22–27, <https://doi.org/10.1016/j.ifacol.2016.07.147>.
17. Wang, L.; Chen, L.; Shao, Z.; Guan, L.; Du, L. Analysis of flexible supported industrial robot on terminal accuracy. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418793022. <https://doi.org/10.1177/1729881418793022>.
18. Mueggler, E.; Faessler, M.; Fontana, F.; Scaramuzza, D. Aerial-guided navigation of a ground robot among movable obstacles. In Proceedings of the 12th IEEE International Symposium on Safety, Security and Rescue Robotics, SSR 2014-Symposium Proceedings, Hokkaido, Japan, 27–30 October 2014. <https://doi.org/10.1109/SSRR.2014.7017662>.
19. Sharma, S.; Kraetzschmar, G.K.; Scheurer, C.; Bischoff, R. Unified closed form inverse kinematics for the KUKA youBot. In Proceedings of the 7th German Conference on Robotics, ROBOTIK 2012, Munich, Germany, 21–22 May 2012; pp. 139–144.
20. Li, B.; Tian, W.; Zhang, C.; Hua, F.; Cui, G.; Li, Y. Positioning error compensation of an industrial robot using neural networks and experimental study. *Chin. J. Aeronaut.* **2021**, *35*, 346–360, <https://doi.org/10.1016/j.cja.2021.03.027>.
21. Cao, C.-T.; Do, V.-P.; Lee, B.-R. A Novel Indirect Calibration Approach for Robot Positioning Error Compensation Based on Neural Network and Hand-Eye Vision. *Appl. Sci.* **2019**, *9*, 1940, <https://doi.org/10.3390/app9091940>.
22. Yuan, P.; Chen, D.; Wang, T.; Cao, S.; Cai, Y.; Xue, L. A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot. *Adv. Mech. Eng.* **2018**, *10*, 2018. <https://doi.org/10.1177/1687814018763411>.
23. Zhu, W.; Li, G.; Dong, H.; Ke, Y. Positioning error compensation on two-dimensional manifold for robotic machining. *Robot. Comput. Manuf.* **2019**, *59*, 394–405, <https://doi.org/10.1016/j.rcim.2019.05.013>.
24. Marchal, P.C.; Sörnmo, O.; Olofsson, B.; Robertsson, A.; Ortega, J.G.; Johansson, R. Iterative Learning Control for Machining with Industrial Robots. *IFAC Proc. Vol.* **2014**, *47*, 9327–9333, <https://doi.org/10.3182/20140824-6-za-1003.00550>.
25. Pane, Y.P.; Nagesh Rao, S.P.; Kober, J.; Babuška, R. Reinforcement learning based compensation methods for robot manipulators. *Eng. Appl. Artif. Intell.* **2018**, *78*, 236–247, <https://doi.org/10.1016/J.ENGAPPAL.2018.11.006>.
26. Qi, J.; Chen, B.; Zhang, D. A calibration method for enhancing robot accuracy through integration of kinematic model and spatial interpolation algorithm. *J. Mech. Robot.* **2021**, *13*, 061013, <https://doi.org/10.1115/1.4051061>.
27. Jing, W.; Goh, C.F.; Rajaraman, M.; Gao, F.; Park, S.; Liu, Y.; Shimada, K. A Computational Framework for Automatic Online Path Generation of Robotic Inspection Tasks via Coverage Planning and Reinforcement Learning. *IEEE Access* **2018**, *6*, 54854–54864, <https://doi.org/10.1109/access.2018.2872693>.
28. Schaal, S.; Atkeson, C.G. Learning Control in Robotics. *IEEE Robot. Autom. Mag.* **2010**, *17*, 20–29, <https://doi.org/10.1109/MRA.2010.936957>.
29. Sumanas, M.; Bucinskas, V.; Vilkonciene, I.M.; Dzedzickis, A.; Lenkutis, T. Implementation of Machine Learning Algorithms for Autonomous Robot Trajectory Resolving. In Proceedings of the 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 25 April 2019; pp. 1–3. <https://doi.org/10.1109/eStream.2019.8732166>.
30. A Fahmy, A.; Kalyoncu, M.; Castellani, M. Automatic design of control systems for robot manipulators using the bees algorithm. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2011**, *226*, 497–508, <https://doi.org/10.1177/0959651811425312>.
31. Liu, K.P.; Li, Y.C. Vibration suppression for a class of flexible manipulator control with input shaping technique. In Proceedings of the 2006 International Conference on Machine Learning and Cybernetics, Dalian, China, 13–16 August 2006; Volume 2006, pp. 835–839. <https://doi.org/10.1109/ICMLC.2006.258481>.
32. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260, <https://doi.org/10.1126/SCIENCE.AAA8415>.

33. Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **2015**, *521*, 452–459, <https://doi.org/10.1038/nature14541>.
34. Dimiduk, D.M.; Holm, E.A.; Niezgoda, S.R. Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. *Integr. Mater. Manuf. Innov.* **2018**, *7*, 157–172, <https://doi.org/10.1007/s40192-018-0117-8>.
35. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311, <https://doi.org/10.1137/16m1080173>.
36. Adam, B.; Smith, I.F. Reinforcement Learning for Structural Control. *J. Comput. Civ. Eng.* **2008**, *22*, 133–139, [https://doi.org/10.1061/\(asce\)0887-3801\(2008\)22:2\(133\)](https://doi.org/10.1061/(asce)0887-3801(2008)22:2(133)).
37. Qiu, J.; Wu, Q.; Ding, G.; Xu, Y.; Feng, S. A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**, *2016*, 67. <https://doi.org/10.1186/s13634-016-0355-x>.
38. Arimoto, S.; Kawamura, S.; Miyazaki, F. Bettering operation of Robots by learning. *J. Robot. Syst.* **1984**, *1*, 123–140, <https://doi.org/10.1002/rob.4620010203>.
39. YouBot Detailed Specifications. Available online: [http://www.youbot-store.com/wiki/index.php/YouBot\\_Detailed\\_Specifications](http://www.youbot-store.com/wiki/index.php/YouBot_Detailed_Specifications) (accessed on 14 October 2022).
40. Wang, Z.; Wu, Z.; Zhen, X.; Yang, R.; Xi, J.; Chen, X. A two-step calibration method of a large FOV binocular stereovision sensor for onsite measurement. *Measurement* **2015**, *62*, 15–24, <https://doi.org/10.1016/J.MEASUREMENT.2014.10.037>.
41. ISO 9283:1998; Manipulating Industrial Robots—Performance Criteria and Related Test Methods; ISO/TC 299 Robotics. Technical Committee: Geneva, Switzerland, 2003.
42. Cowan, G. *Statistical Data Analysis*; Clarendon Press: Oxford, UK, 1998.