Improving the Absolute Accuracy by Online Interpolation Technique of Industrial Robots

Darpan Patel¹, Lars Lienenlüke¹*, Simon Storms¹, Christian Brecher¹ and Johannes Wied²

¹Department Automation and Control, Chair of Machine Tools, Laboratory for Machine Tools and Production Engineering (WZL, RWTH Aachen), Campus-Boulevard 30, 52074 Aachen, Germany
²Technology factory, Press shop, Mercedes-Benz Plant Sindelfingen, 71059 Sindelfingen, Germany

e-mail address corresponding author: l.lienenlueke@wzl.rwth-aachen.de

Abstract. This paper describes a novel method to improve the accuracy of industrial robot by calibration technique. Robot calibration is broadly classified into model-based and modeless techniques. In this paper, a modeless technique capable of interpolating the deviation along the Z-axis is defined which doesn’t consider any kinematic model, instead adapts interpolation method to compensate errors. The robot’s deviation is measured with a laser tracker and the deviation controlling parameters are calculated. The controlling parameters interpolate the deviation and with an online compensation program this deviation is compensated in real time scenario. This modeless technique is experimentally tested and it demonstrates a substantial improvement in the accuracy of the robot.

1. Introduction

Precision machining is widely practiced in automobile and aerospace industries. A conventional approach for precision machining is with machine tools. They have been widely used in the industries and are preferred over the articulated robots. One of the main reasons of machine tools being chosen over robots is due to its better accuracy. While articulated robots are known for its repeatability, accuracy remains a topic of discussion. Most manufacturing industries use the robot manipulators for pick and place tasks which they perform adeptly. In comparison with the special purpose machines, the robots offer a greater work volume, flexibility and are relatively cost effective [1, 2].

Modern manufacturing industries demand a flexibility in their manufacturing operations. Since industrial robots can perform several machining operations and can easily adapt to different work volumes, they offer a great potential for flexibility in manufacturing industries. However, the use of robots for precision machining is compromised due to its inability to perform with high accuracy. Due to this, efforts are being taken to improve the accuracy of the industrial robots and to fulfil the demands of precision manufacturing. However, robots consist of a complex assembly system and a small change in kinematic model adversely affects the robot’s performance. Therefore, inaccuracy of robots is intrinsic. Hence, it is necessary to understand the robot kinematics to perform calibration tasks. There are several robot calibration companies that offer calibration solutions which are often insufficient. These model based approaches mainly consider axis length, joint angle and joint position
deviations, but neglect important factors like gear backlash or joint stiffness. So the considerable effort for calibration and compensation brings only small improvements in absolute accuracy. The approach that is presented in this paper leads to a significant improvement that is easy to implement. It is also difficult to validate the results of calibration offered by these companies, as there are several aspects which may be ignored during calibration. To elaborate on this issue, the calibration experiment might be conducted under a specific loading condition and therefore when different loading conditions are applied on the robot, it is difficult to comment on inaccuracy observed even after calibration. Similarly, the ambient temperature of the robot needs to be monitored and maintained to accurately calibrate a robot. Therefore, the above-mentioned aspects are to be considered to develop a compensation technique capable of being installed on several robots is to be developed.

The research work is further distributed in the following manner. The initial survey of the recent calibration technologies are explained in section 2. Section 3 presents measurement technique and the methods to estimate the amplitude of deviations automatically with Matlab and Simulink to determine the parameters required further for compensation. In section 4, the concept of the modeless calibration method developed is described. The working principle of the implemented modeless technique is explained in the section 5. The compensation technique is further verified experimentally by conducting measurements with the laser tracker and validated by performing milling process in section 6. The interpretation of the results along with the conclusion is discussed in section 7.

2. State of the Art
The robot consists of a complex assembly structure. The accuracy of the robot is usually affected due to reasons like prolonged use of robots, assembly, installation tolerances, stiffness, gear backlashes as well as software errors like calculation error of the TCP (Tool Center Point) position inserted in robots control system. Finding the location of such errors is a great challenge. Therefore, altering the design or disassemble and reassembling the robot can create new challenges. Different techniques and methods have been developed to calibrate robots which is based on different models. In an effort to further elaborate calibration techniques performed on robots, it is broadly classified as model-based and modeless methods [3]. The model-based calibration technique encompasses implementation of a kinematic model for the robot, conducting measurements by moving the robot inside the work volume and then with the modified kinematic parameters, compensation is performed [4]. The advantage of a model-based calibration is that the kinematic parameters can be determined for a relatively large work volume. All poses lying within this working volume can be compensated with these parameters [3, 5]. This implies that as the work volume is reconfigured, a new set of parameters need to be determined. The disadvantage of this model is that one requires a thorough knowledge of kinematical modeling as well as determining its parameters is a complex process. This knowledge might remain limited to few experts in kinematical modeling and is not easily interpretable for field engineers [3, 5]. On the contrary, a modeless method does not consider any kinematic modeling and parameter estimation. In this method, measurements are taken throughout the robot’s workspace and the errors are recorded. With the help of these errors, the robot’s workspace is divided into a number of 2D squares and 3D cubes. The errors along these 2D squares and 3D cubes are known from previous measurements and errors can be compensated by simple interpolation techniques [6]. Even though this method consists of a relatively large measurement, its simple implementation and effectiveness have led many industries to adopt this technique [3, 5].

Since the use of robots in machining industries has increased, several researchers have taken up the challenge to develop accurate kinematic models. A linear model and complex model were developed which could accurately compensate the first-order and second-order error effects. Since then several kinematic models such as Hayati et al. models, Stone and Sanderson’s S-model and Zhuang et al. model have been developed [4, 7]. But the most recent research has led to the implementation of the Screw method and the Denavit-Hartenberg (DH) method out of which DH method is widely practiced in the industries. Based on the short comings of DH method and the complexity involved in matrix calculation, an alternative method called Direct Coordinate System method is developed [8]. A common approach to all the above-mentioned methods is to accurately develop a kinematic model based on the robot design and determining the optimized parameters based on the errors. With the help
of these optimized parameters, a backward transformation is carried out to redefine the robot joint angles.

A common approach for measuring the accuracy of the robot is with a laser tracker or by a stereo camera. Normally, a single reflector is attached to the spindle to record its coordinates. However, this imposes a few restrictions to move the TCP in certain positions and poses. This can be overcome with a square tool consisting of 4 reflectors on each side. Hence ensuring that at least one reflector is detected by the laser tracker. Subsequently, the TCP can be guided to several positions and with different poses. By conducting several calibration tests with this tool, the kinematic parameters of joints 2, 3, 4 and 5 can be determined along with the present compliance [9]. Another method to evaluate the DH parameters is by the stereo vision system. By placing 2 CCD cameras perpendicular to one another, a virtual TCP can be generated. This virtual TCP is moved along the measurement points manually within the range of calibration volume. A machine vision algorithm is developed to determine the robot joints and to further rectify these joints [10]. Another approach to finding the optimized joint parameters is to place a marker on the TCP which can be detected by 2 or more cameras placed at different positions [4].

On the other hand, the modeless approach does not consider any kinematic model, instead, it adapts interpolation methods to compensate error. Several techniques developed to compensate error with modeless approach were found in the literature. The TCP is moved along the work volume and the position of it is continuously measured with a CMM (Coordinate Measuring machine) device. The work volume of the robot is divided into a number of 2D and 3D cells and the error measured along these cells is noted [6]. A method is developed to statistically analyze the error and to develop a histogram function based on the data. An algorithm is developed with this histogram function to improve the performance of robot [11]. Another method is developed by dividing the workspace into fan shaped cells with each cell consisting of 8 points. In this technique, the error along each point of the cell is measured and the position of the target point is estimated by linearly interpolating the position based on the error calculated at the grid points [6]. Subsequently, the interpolation of the target points is proposed and implemented with fuzzy logic. A fuzzy logic incorporates a membership function for each of the measured output and desired target output. The difference between the two membership function is defined by the error calculated. A set of control rules are required to be defined to determine the position error. That implies, to interpolate the error in a cube defined by 8 points, 8 control rules are required [5].

3. Experimental Setup

In this research, the authors have developed a modeless calibration technique which incorporates a work volume of 1000 mm×4200 mm×500 mm. The deviation along Z-axis within this work volume is measured with a Leica laser tracker. The laser tracker is a highly accurate measuring device with an accuracy of ±15 μm (+6μm for high distances). The robot used for the experimentation is a KUKA KR500 L480-3 MT. These measurements are further processed in Matlab to determine the inaccuracy of the robot. Further, with the help of Simulink models, parameters controlling the inaccuracy are determined. With these parameters, the error is interpolated based on the TCP position and further compensated. The causes of deviation and its effect on machining primarily focused on the issue of deviation observed in the Z-axis. The deviation along Z-axis accounts for deteriorating the surface quality during machining operations like milling and laser processing. It is observed that the error present in first 3 joints contributes a higher percentage of error in comparison to the errors of the other 3 joints [6]. Based on the motion of the first 3 joints of a robot, a technique to determine the amplitude of this deviation is developed.

3.1. Measurement technique

A curved path test is defined which takes into account the work volume of the robot during machining (refer Figure 1). The test is necessarily performed by guiding the TCP along a path defined by 24 points in one plane. During the measurement the laser tracker is placed stationary at a corner of the robot cell. To completely encompass the whole work volume, joint 1 is rotated from +50° to −50° by a constant increment of 25°± 5°. During this motion of joint 1, the reflector needs to be continuously
rotated for the laser tracker to detect it. The first curve is measured by rotating the joint 1 with a radius of \(X = 2400\) mm. In all, 5 curves are measured with each curve exceeding the previous one by 200 mm in positive X-axis. Whereas, the limits of Y-axis are approximately \(\pm 2000\) mm. The measurement is carried out for each plane’s starting from \(Z = 500\) mm to 800 mm. The defined measurement volume is preferred since the machining operations are conducted within the illustrated volume. As previously mentioned, the errors of joints 1, 2 and 3 are to be measured. For this reason, the joints 4 and 6 are not changed at all and the spindle is maintained perpendicular to the plane of measurement. Further, the joint 5 moves itself by a minimal angle as compared to the first 3 joints to maintain the spindle perpendicular to the plane of measurement.

Further, a 6-points slant test is illustrated by 6 points positioned diagonally to one another. This test is conducted only in the positive Y-axis once symmetric deviation is observed along negative and positive Y-axis. Additionally, all the 6 points cannot be measured in the negative Y-axis since the laser tracker is placed in this corner of work volume. Hence, points 5 and 6 would not be detected in negative Y-axis by the laser tracker as they lie too close to the measuring volume of the laser tracker. The first point is measured at \(X = 2200\) mm and \(Y = 0\) mm. Subsequent points are measured by gradually travelling in Y-axis and increasing distance in X-axis by 200 mm in comparison to the previously measured point. The limit of Y-axis is gradually increased from 0 to 2000 mm. The measurement is carried out for \(Z = 500\) to 800 mm.

![Figure 1. Curve path test](image)

### 3.2. Data evaluation

The points measured by the laser tracker are measured with respect to the laser coordinate system (LCS). Therefore, to accurately evaluate the measured data, the points measured should be represented in the robot coordinate system (RCS). A Matlab program is written which helps perform the coordinate transformation. During the test, the TCP travels to different points in the defined work volume. The coordinate transformation of these points is calculated from LCS to RCS. A Matlab function performs a complex calculation to determine translation, rotation, scaling and reflection matrix with the best fit method and stores this data for further calculation. In addition, a Matlab function determines the quality of transformation by displaying the standard deviation between the theoretical and measured point. Subsequently, the measured coordinates in RCS are used for further calculation to identify the parameters affecting the deviation. To elaborate on this issue, different Simulink models are created based on the type of deviations. These Simulink models calculate the controlling parameters which are required for the compensation technique.

### 3.3. Types of deviation

A curved path test is initially conducted to measure the deviation along every plane. After the evaluation of the measured data in Matlab it is evident that the TCP does not travel along a straight
plane. Subsequently, after an initial survey of evaluated data, it was observed that the deviation in the positive Y-axis is greater than the deviation in negative Y-axis. This phenomenon was common to all planes. The difference of deviation in Z-axis is observed by comparing the deviation of a point in positive Y-axis with its corresponding point in negative Y-axis. Hence, this confirms the presence of inclination error.

In addition to inclination error observed, a considerable deviation is observed when the TCP travels along the X-axis. When the TCP moves further from one curve to another, a radial distance of 200 mm is added to the previous radius and ensures that each curve is equally spaced to one another. After comparing the deviations observed for all points in one curve with their corresponding points to the adjacent curve, it is observed that the deviation gradually increases for each curve. This deviation is referred as the forward deviation. Furthermore, as the TCP travels in the positive and negative Y-axis along the curve, the deviation increases forming a curved surface which is denoted as sideway deviation.

In addition to the forward and sideway deviation, a non-linear pattern is observed while comparing the deviation for each plane. This implies, that the deviation is found to be above the intended plane of travel and at times below the intended plane of travel. A general observation for the deviation of all planes demonstrates that the deviation is negative to the expected plane till Z = 500 mm and below whereas, the deviation is positive to the expected plane from Z = 700 mm and higher. In the plane between 500 to 700 mm, a mixed deviation of positive and negative to the expected plane is observed. Furthermore, not only the deviation is either positive or negative but also the amplitude of deviation varies for every plane.

It should be noted that all the tests were conducted under same loading conditions and therefore different loads cannot be attributed to the cause of such a deviation. However, it can be observed that as the robot arm extends and contracts to travel along different curves, the force acting at the end effector might enhance the effects of joint clearance and drive backlash. This phenomenon of deviation can be attributed to joint clearance and drive backlash of the joint 2 and 3 which are altered for each curve. Therefore, the curve path test is capable of evaluating 3 different types of errors along X-axis (forward deviation), Y-axis (sideway deviation) and Z-axis (positive and negative deviation). The amplitude along X and Z-axis can be appropriately determined by this method. However, to determine the amplitude of sideway deviation the 6-points slant test is conducted.

4. Error Compensation Based on Modeless Method
A modeless compensation, as mentioned in section 2 divides the work volume of the robot into several parts and the error for each part is noted. A similar method is implemented by conducting measurement tests in accordance with section 3.1. The curve path test conducted in a single plane measures the error in a 2D plane along X and Y-axis. The test is further conducted by changing only the coordinates of Z-axis and the coordinates of the X and Y-axis are unchanged. This ensures the error for a point with the same X and Y coordinates is obtained at different planes. In accordance to that, an error along a confined 3-dimensional volume is measured. This measurement technique enables to examine the behavior of the robotic arm and its corresponding deviation along different work volumes. Based on the deviations measured, a prominent modeless compensation approach can be adopted. A traditional approach to implement a modeless compensation is by implementing various interpolation techniques.

4.1. Parameter estimation
A huge deflection is observed known as inclination error between the points measured to the left of the robot and to the right. This deflection is observed throughout the work volume. Further, the amplitude of the deviation also remains almost constant for all the measured planes. Therefore, the slope of inclination can be calculated between the points to compensate the error. To accurately calculate the amplitude and the slope of inclination, a Simulink model is constructed. The Simulink model determines this difference along all the plane and computes an average of the difference. The output of the Simulink model is further used in the Kuka Robot Language (KRL) program to interpolate the
difference. Since this inclination error is observed along the Y-axis the angle is calculated based on the distance travelled along Y-axis in the work volume.

![Graphical representation of robotic arm as cantilever beam](image)

**Figure 2.** Graphical representation of robotic arm as cantilever beam

A common phenomenon observed along all the planes is the forward deflection along X-axis and the sideway deflection along Y-axis. This phenomenon can be compared to deflection observed in a cantilever beam (refer Figure 2). In the presence of different geometric positions and under variable loading condition as well as gravity effect, the robotic manipulator experiences a deformation [12]. In this case, the deformation is attributed to the deflection caused in the Z-axis. Subsequently, as the robotic arm extends gradually along its work volume, the weight of the end effector under the effect of gravity causes an additional deflection [13]. The results of a curve path tests reveal that under a constant load due to self-weight of the robot end effector, a forward and a sideways deflection is obtained for each curve. To accurately determine this deviation as the robot arm extends, the deviation is calculated by an equation of cantilever beam acted upon by the moment. In this case, the deformation is attributed to the deflection caused in the Z-axis. Subsequently, as the robot joint extends gradually along its work volume, the weight of the end effector under the effect of gravity causes an additional deflection [13]. The results of a curve path tests reveal that under a constant load due to self-weight of the robot end effector, a forward and a sideway deflection is obtained for each curve. To accurately determine this deviation as the robot arm extends, the deviation is calculated by an equation of cantilever beam acted upon by the moment. All the robot joints are transformed into a free body diagram representing a cantilever beam. However, the measured deviation is a known factor in this case, as well as the length of the beam can be compared to the stretch of the robotic arm. Therefore, the only decisive factor to be resolved is the \( \theta \) which is known as the angle of deflection. To determine \( \theta \) by an equation of cantilever beam acted upon either by a force or moment is inconsequential. But this equation should be later contemplated to interpolate the deviation with KRL programming. In accordance to that, the \( \theta \) is determined with a moment equation implemented with a Simulink model. The Simulink model computes a value for theta for the individual plane and further calculates an average value of theta for all planes. With this method, two different \( \theta \)'s are computed for each sideway and forward deflection. The curve path test is prescribed to determine the forward theta. Whereas, the 6-points slant test is performed to calculate the sideway theta.

As the robot travels further along the Z-axis the motion of robot joint 2 and 3 causes a non-linear deviation. To summarize the behavior of error, the error for lower planes is negative and the error escalates perpetually in positive Z direction for higher planes. This implies that the TCP travels either above or below the designated plane of travel. Furthermore, the amplitude of deviation for each plane also varies. This phenomenon is observed even when the measurements are conducted after compensating the inclination error, forward and sideway deflection with the KRL program. To compensate such an error, a method is established with which the robot TCP travels along the designated plane with minimum deviation along the Z-axis. A Simulink model is designed to determine the amplitude of error for each plane and to determine a compensating value for the error. For this, the curved path test is performed after compensating the inclination, forward and sideway deflection. Only the coordinates of Z-axis for all the measured planes are fed into the Simulink model as the input. The average deviation of each plane is calculated in the model. Further, the average of one plane is divided with an average deviation of the subsequent plane. The output of this calculation ensures a multiplying factor is computed between two points which acts as a slope of inclination between the errors of two planes. This multiplying factor is treated as an angle to define the slope of deviation between the two points. The number of planes to be considered for calculating the multiplying factor reckons on the behaviour of the error observed. It should be noted that if the
amplitude of error for two planes is approximately same, the value of multiplying factor would eventually be reduced. This is because the slope of inclination between the two points would be less. At times, when the multiplying factor has a relatively low value, the error is not accurately compensated for the planes displaying a considerable error.

4.2. Compensation of error by interpolation

With the angle of inclination calculated for inclination error, the amplitude of deviation can be interpolated in real time with a KRL program. This deviation can be obtained by:

$$\Delta = Y_{\text{present}} \times \tan(\theta_{\text{inc}})$$  \hspace{1cm} (1)

where, $Y_{\text{present}}$ is the Y coordinate of the point to be interpolated and $\theta_{\text{inc}}$ is the inclination slope. This interpolated deviation given by $\Delta$ is further added to the corresponding Z coordinate of the point (refer Figure 3 (1)). Furthermore, the amplitude of deviation depends upon the coordinate of Y-axis and this deviation is added to the present coordinate value of the Z-axis in real time scenario. Since the deviation of errors was found to be greater along positive Y-axis as compared to the negative Y-axis, the inclination error compensation is implemented along the positive Y-axis to maintain a proportional deviation along positive and negative Y-axis.

As mentioned in section 4.1, two different thetas are computed for each forward and sideway deflection. The interpolation of the forward deviation in KRL is represented as:

$$L = \text{sqrt}\left(\left(X_{\text{present}}\right)^2 + \left(Y_{\text{present}}\right)^2\right)$$  \hspace{1cm} (2)

$$\Delta_f = \frac{L \times \theta_{\text{forward}}}{2}$$ \hspace{1cm} (3)

$$\Delta_s = \frac{Y_{\text{present}} \times \theta_{\text{sideway}}}{2}$$ \hspace{1cm} (4)

where, $L$ is the length of the cantilever beam corresponding to the robotic arm and $X_{\text{present}}$ is the X coordinate of the point to be interpolated. Further, $\theta_{\text{forward}}$ and $\theta_{\text{sideway}}$ are forward and sideway deflection slopes computed previously. Once $\Delta_f$ and $\Delta_s$ are computed, the interpolated deviations are added to the corresponding Z coordinate of the point (refer Figure 3 (2)).

Once the Simulink model computes the value of inclination slope for non-linear deviation, the cosine of the slope value is multiplied with the Z coordinate of the designated plane of travel. Herewith, an amplitude of error for the following plane is interpolated which can be either added or subtracted to the actual Z coordinate depending on the form of error measured previously (refer Figure 3 (3)). The interpolation is computed as follows:

$$\Delta = Z_{\text{present}} - \left(Z_{\text{present}} \times \cos(\theta_i)\right)$$ \hspace{1cm} (5)

here $\theta_i$ is the inclination slope calculated to compensate the non-linear deviation. It should be noted that all the above-mentioned interpolations are calculated for each point defining the motion of the robot.

5. Compensation Working Principle

This research work is focused towards developing a compensation technique without utilizing any external control device. Based on these limitations, a method to compensate the deviations in real time scenario is developed. A common practice in the industry to machine with an industrial robot is by programming the operation in CAM (Computer Aided Machining) software like Mastercam and developing a code from a post-processor software like Robotmaster. Further, the post-processor
creates program referred as main program which can be directly run in the robot controller to perform machining operation. A compensation program is developed in KRL by integrating the available syntaxes for a KR C4 controller. Further, this compensation program is presumed to perform an interpolation utilizing the parameters estimated with the Simulink. Consequently, a function is defined which is capable of performing the necessary calculation and returning the calculated value in a specified format defined in the function. It is noted that the input and output format for a function should be defined in the same format. Subsequently, a function with name CorrectPos is defined with its input as PresentPos in E6POS format. To make PresentPos be accessible for subsequent calculation, it is declared as an object or variable in E6POS. The E6POS is a data type to represent a point in cartesian workspace along with the orientation angle of the robot. This implies in addition to X, Y, Z coordinates the corresponding A, B and C angles are defined. Further, this data type consists of S (Status) and T (Turn) which denote the axis precisely [14]. The input variables for the function are transferred from the main program. Every program generated by Robotmaster is treated as the main program. This main program consists of every point described by either LIN, PTP or CIRC syntaxes. Motion programming is defined with these syntaxes. These syntaxes include the coordinate information associated with the defined points. The main program generated by Robotmaster is usually represented in LIN format as:

\[
\text{LIN} \{X \ 100, \ Y -20, \ Z \ 250, \ A -180, \ B -0.1, \ C -180\}
\]

In order to transfer the parameters from the main program to the developed function, the abovementioned point should be represented as:

\[
\text{LIN} \ \text{CorrectPos} \ (\{X \ 100, \ Y -20, \ Z \ 250, \ A -180, \ B -0.1, \ C -180\})
\]

This alteration enables the coordinate data to be transferred into the function for further processing. Once the coordinates are transferred into the function, they can be accessed by PresentPos variable. Further, the interpolation of error is executed in the following order:

1. Inclination interpolation
2. Forward and sideway deflection interpolation
3. Non-linear deviation interpolation

![Flowchart for KRL compensation program](image)

**Figure 3.** Flowchart for KRL compensation program

The CorrectPos function calculates the necessary deviation and updates the Z coordinate for every single point defined in the main program. The Z coordinate is updated in the PresentPos and the function concludes by returning PresentPos in E6POS format to the robot controller. The compensation function is so defined that the KR C4 controller first calculates the compensation and then travels to the point with the updated coordinates.
6. Validation
The compensation function developed had to be tested to check the functionality. This was performed with the aforementioned curve path test as per section 3.1. In addition, this test was performed to compare the results with and without the influence of compensation. Therefore, the measurement was carried at the same planes without changing any of the coordinates. The advantage of this procedure is, due to good repeatability properties of the robot, the deviation at each point can be distinctively observed. The amplitude of deviation after every compensation was noted, and it was found that the error compensated at one stage has an adverse effect on the parameters controlling the successive compensation. This implies, in case the inclination error is over compensated the successive sideway compensation parameter is affected. The sideway theta would reduce in this case. Similar observations were noticed between the forward and non-linear deflection.

Table 1. Deviation compensated in mm after each interpolation for planes along Z-axis

| Plane       | 500 mm | 600 mm | 700 mm | 800 mm |
|-------------|--------|--------|--------|--------|
| Without     | -1.604 | -0.992 | 0.973  | 1.52   |
| Inclination | -1.488 | -0.828 | 1.064  | 1.565  |
| Forward & Sideway | -1.176 | -0.516 | 0.672  | 1.87   |
| Non-linear  | 0.076  | 0.46   | 0.144  | 0.584  |

After comparing the measurement results with and without the compensation program, it is evident the compensation function substantially compensated different types of deviations observed previously (refer Figure 4, Figure 5).

![Figure 4. Measurement result without compensation for plane Z=500 mm](image)

The accuracy of the compensation function is to be tested while performing machining operations. For this reason, a milling operation is chosen to validate the performance of the KRL program. A Zinc aluminum alloy workpiece was used on which the milling operation was performed. In order to validate the results accurately, it was necessary to perform milling operations with robot arm as extended as possible. This would ensure that the deviations in Z-axis would be greater when the robot arm was further extended. Therefore, a workpiece with dimensions $1550 \times 2795 \times 510$ mm was selected. Further, locking pins with 50 mm height were placed between the working table and the workpiece and this changed the Z dimension to 560 mm. The workpiece was placed as far as possible in the X-axis.
Two paths were defined to mill the workpiece along a straight line starting from the extreme left of the workpiece and travelling towards the right with a depth of cut 0.5 mm. Both the paths were defined as far as possible in the X-axis so that the robot arm is extended. One of the paths was programmed with the compensation whereas, the other path was programmed without compensation. Further, two U-section paths are defined, each to the extreme left and right of the workpiece with respect to the robot. The path defined to the left of the robot was programmed without the compensation whereas, the path to the right was defined with the compensation. The idea was to observe a deviation between the depths of cut for both the paths. The milling operation was programmed to perform a depth of cut of 1 mm for both the paths. The milling process was conducted with the same tool and constant feed-rate for all the paths. Later, the depth of cut was measured with an absolute height gage to compare the effect of a milling process with and without compensation. Nevertheless, a substantial difference between the depths of cuts is observed between the path 1 (compensation) and 2 (without compensation) as well as path 3 (without compensation) and 4 (compensation). The depth of cut was measured along different points of the path to observe a pattern of deviation (refer Figure 6).
was observed to be uniform along the path 1. Theoretically, due to the extension of the robot arm to travel to path 1, the depth of cut was expected to be deeper. As for the curve path test, every curve was shifted by 200 mm and a deviation for every curve was observed in this case. However, the compensation function fairly maintained a uniform deviation along the path 1. Points corresponding to the path 2 were measured along path 1 to compare the results and observe the effects of compensation.

Comparing the results of path 1 and 2, it is evident that the depth of cut for path 2 is greater and non-uniform as compared to the depth of cut of path 1. However, the programmed depth of cut was 0.5 mm and the measured depth of cut for both the paths confirmed the presence of deviation.

| Table 2. Comparison of depth of cut for path 1 and 2 |
| Path | Right corner | Center | Left corner |
|------|--------------|--------|------------|
| 1    | -1.59 mm     | -0.88 mm | -1.46 mm   |
| 2    | -0.67 mm     | -0.86 mm | -0.41 mm   |

To further investigate the deviations in the area near to the robot, milling across path 3 and 4 were conducted. The parallel paths defined for the U section were maintained at a distance of 350 mm apart. This distance has no significance in terms of the functioning of the compensation program but, the reason was to observe a difference in depths between the two parallel paths. Nevertheless, paths 3 and 4 were programmed symmetrically about the centre of the workpiece to compare the results. Once the milling process was performed, the depth of cut along the path was measured. For path 3 as anticipated, a difference between the depths of the cut was observed for the two parallel paths in the U-section. This implies, that as the robot arm extended gradually while milling across the U-section, the deviation in Z-axis could be clearly observed in terms of depth of cut. The measurements were conducted on the path 4 to compare the results. It was observed that the depth of cut was nearly uniform for the parallel lines. This was tested by measuring several points along the path and not only the corresponding points measured about path 3. Therefore, this uniform depth of cut can be attributed to the functioning of the compensation program.

7. Conclusion

The aim of this research work is to investigate the possibilities of developing a technique to compensate the deviation in Z-axis using industrial robots in machining operations without the application of any external controller. For this, the deviations were measured with a laser tracker and depending upon the pattern deviations a compensation program is created in KRL. Later, measurements were conducted with the compensation, and results were compared to the initial measurement. It was evident that the absolute accuracy was substantially improved from ± 1.19 mm to ± 0.47 mm in terms of standard deviation. Validations were performed experimentally to examine the functionality of compensation technique. For this, milling process was performed on a workpiece with and without the compensation function. The depth of cut for all the milling processes was compared and the deviation in depth of cuts with and without compensation was compared. The results of laser tracker measurement and validation displayed a substantial improvement in the absolute accuracy by almost 60%. However, the reaction forces during cutting operations or the vibrations were not taken into account during validation and future work should include a method to compensate these forces. Further, the errors caused due to an external linear axis are also not considered during this research work and should be considered for future work.

8. References

[1] Lienenlücke L, Gründel L, Storms S, Brecher C. Model-based process planning for milling operations using industrial robots. In: International Conference on Control and Robotics Engineering; 2018, p. 37–44.

[2] Ceccarelli M, Glazunov VA. Advances on Theory and Practice of Robots and Manipulators: Proceedings of Romansy 2014 XX CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators. 2014th ed. Cham: Springer International Publishing; 2014.
[3] Bai Y, Zhuang H. Modeless robots calibration in 3d workspace with an on-line fuzzy interpolation technique. In: Wieringa P, editor. IEEE International Conference on Systems, Man & Cybernetics theme: Impacts of emerging cybernetics and human-machine systems conference proceedings. Piscataway (N.J.): IEEE; op. 2004, p. 5233–5239.

[4] Zhang X, Song Y, Yang Y, Pan H. Stereo vision based autonomous robot calibration. Robotics and Autonomous Systems 2017.

[5] Bai Y, Wang D. On the comparison of an interval Type-2 Fuzzy interpolation system and other interpolation methods used in industrial modeless robotic calibrations. In: IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications - CIVEMSA: July 27-29, 2016, Hotel Novotel Budapest Centrum, Budapest, Hungary proceedings. Piscataway, NJ, Piscataway, NJ: IEEE; 2016, p. 1–6.

[6] Guo Y, Yin S, Ren Y, Zhu J, Yang S, Ye S. A multilevel calibration technique for an industrial robot with parallelogram mechanism. Precision Engineering 2015.

[7] Marwan A, Simic M, Imad F. Calibration method for articulated industrial robots. Procedia Computer Science 2017.

[8] Zhou S, Song Q, Wang X. A kinematics modeling method of linkage robot based on euler spinning method. In: Xu X, editor. Proceedings of 2017 IEEE International Conference on Unmanned Systems (ICUS): Oct. 27-29, 2017, Beijing, China. Piscataway, NJ: IEEE; 2017, p. 104–109.

[9] Nubiola A, Bonev IA. Absolute calibration of an ABB IRB 1600 robot using a laser tracker. Robotics and Computer-Integrated Manufacturing 2013.

[10] Švaco M, Sekoranja B, Šuligoj F, Jerbić B. Calibration of an Industrial Robot Using a Stereo Vision System. Procedia Engineering 2014.

[11] Zhuang H, Wu X. Membership function modification of fuzzy logic controllers with histogram equalization. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics a publication of the IEEE Systems, Man, and Cybernetics Society 2001.

[12] Mao Y, Jing F, Liang Z, Fang Z. Positional accuracy analysis of welding robot under mechanism clearance and elastic deformation. In: 12th World Congress on Intelligent Control and Automation (WCICA). Piscataway, NJ: IEEE; 2016, p. 3253–3258.

[13] Shi S, Wu H, Song Y, Handroos H, Li M, Cheng Y et al. Static stiffness modelling of EAST articulated maintenance arm using matrix structural analysis method. Fusion Engineering and Design 2017.

[14] KUKA AG. KUKA System Software 8.3: Operation and Programming Instructions for System Integrators; 2017.

Acknowledgement
The authors of this research would like to acknowledge the support of Daimler AG, Sindelfingen for funding this work and letting us conduct calibration experiments on their robots. The IGF-project 20057 N (HORuS®) of the research association FVP (Forschungsvereinigung Programmiersprachen für Fertigungseinrichtungen e.V.) was supported via the AiF within the funding program „Industrielle Gemeinschaftsforschung und –entwicklung (IGF)“ by the Federal Ministry of Economic Affairs and Technology (BMWi) due to a decision of the German Parliament.