Offline Path Correction System for Industrial Robots

Philipp Roebrock
Univ. of Applied Sciences Gelsenkirchen
Department of Informatics
Neidenburger Strasse 43,
45877 Gelsenkirchen/Germany

Kay Böhnke
Univ. of Applied Sciences Gelsenkirchen
Department of Informatics
Neidenburger Strasse 43,
45877 Gelsenkirchen/Germany

Abstract: The paper describes the design of an offline path correction system for industrial robots using laser stripe sensors. Our approach offers high flexibility because of the systems ability to self-calibrate and a high speed measurement because of the usage of approximate positioning of the robot. The paper further reviews the advantages and disadvantages of online and offline path correction systems.

Key–Words: Industrial Robot, Sensor, Offline Path Correction

1 Introduction

Most applications with industrial robots especially those in the automotive industry cannot be realized with only static robot programs due to higher accuracy requirements. Different approaches provide sensor based solutions for this problem achieving different accuracies.

Besides the accuracy of the manipulator system itself including tool and potential external axes there are two components that are most relevant for the accuracy of the complete system, the accuracy of the conveyor system and the production tolerances of the work object. This leads to two different strategies to increase the systems accuracy. For high tolerances in the conveyor system it is common to make a global *position correction* that determines the position change of the work object relative to a certain setup position. This can be done using multiple cameras each viewing certain features of the work object. The position change can be calculated using bundle adjustment techniques (see [10]). If there are high production tolerances in the work object the usage of robot *path correction* might be in order. A path correction corrects the robot path usually by using local sensor systems mounted at the robots hand.

The Fig. 1 illustrates the usage of both approaches. During system setup the work object is posed into the nominal position in the robot cell and the application program for the robot is taught (a). The nodes along the robots path are the path points taught in the robot program and referring to the work object coordinate system. During normal system operation the conveyor places a new work object with certain production tolerances in the robot cell at a slighly different than the nominal position (b). With a position correction the work object coordinate system can be corrected to have the robot path run along the work object again (c). To compensate the production tolerances of the work object a path correction of the single path points is advisable (d).

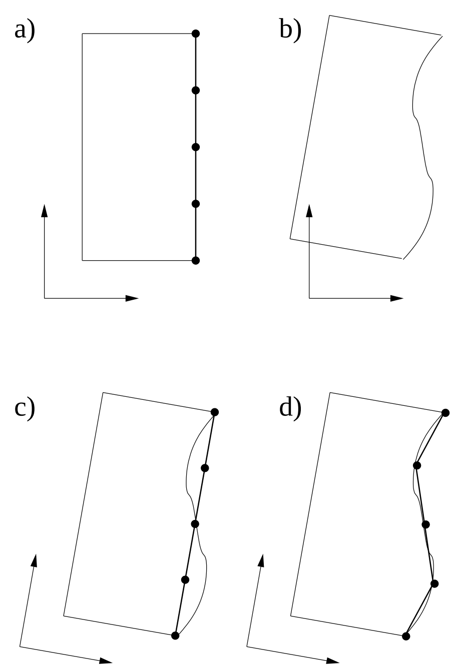

Fig. 1: Position Correction and Path Correction

There is a difference between *online path correction systems* and *offline path correction systems*. Online path correction systems execute measurement and correction calculation continuously in a loop and provide the robot permanently with corrections during his applications run. On the opposite the offline path correction divides the production process into a measurement run and an application run. During the measurement run the measurement data is collected and after calculating the corrections the robot starts its application run with the previously calculated corrections without any further measurement.

This paper describes the design of an offline path correction system with an universal robot interface, self-calibration capabilities and a fast measurement run with compensation of the robot approximate positioning. The sensor used is a laser stripe sensor with hardware trigger feature.

Section 2 introduces the system and its single components. Section 3 gives important aspects of the setup process and Section 4 of the normal system operation. In Section 5 we show some related work connected with the subject of offline robot control with sensors. The paper closes with Section 6 which compares online and offline path correction solutions.

2 System Overview

The main components of the system are a standard industrial robot, a laser stripe sensor attached to the robots hand and an industrial PC (IPC) with the path correction software installed on it. The communication between robot and IPC is packet-based and can use different transportation media like industrial buses or serial protocols. It includes commands for calibration (see Section 3), measurement signaling and transmission of corrections (see Section 4). The robot controls the process and initiates the single process steps, the role of the software system is passive.

The sensor used in this system is a laser stripe sensor, consisting of a laser module and a matrix camera in a compact and ready calibrated sensor module. The laser module projects a laser line on the target, the diffuse reflected light is received by the camera. From the brightness distribution in the camera picture a microprocessor calculates the metric 2D coordinates of a number of points on the target object in the measurement plan ($\hat{=}$ laser projection plane) using the triangulation principle.

To correct the robot path the robot positions the sensor at certain points on his application path creating sensor scans of parts of the work object. To be able to calculate corrections for these points an evaluation algorithm has to extract certain features from the sensor scans. This can be the coordinates of a pivot point of an object edge or the angle under which the object is seen in the sensor scan. These features are the result of a measurement. Fig. 2 shows as an example on the left side a step-like structure on the edge of the work object, on the right the scan of the edge. Marked with an arrow is the pivot point that has to be extracted. The coordinates of the pivot point s_1 and s_2 and the angle of the base line s_3 are referred as the result of a single sensor measurement:

$$\mathbf{s} = (s_1, s_2, s_3) \tag{1}$$

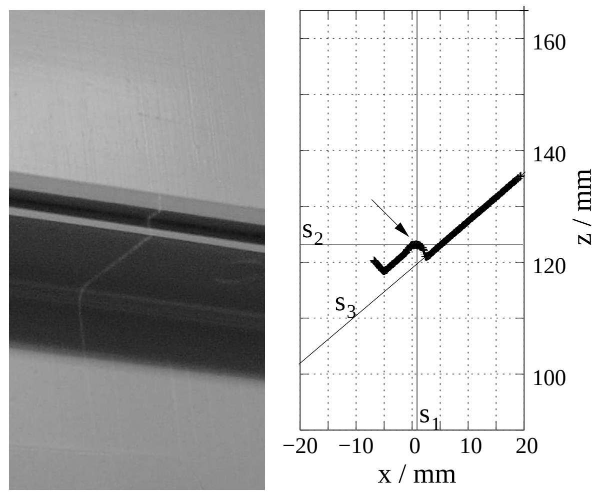

Fig. 2: Object Edge and Laser Scan

The sensor should be attached to the robots tool in a way that the center of the sensors measurement range is as near as possible at the tool center point (*TCP*) of the tool. This reduces additional errors. To make teaching of the robot programs much more easier and to ensure easy replacement of the sensor in case of sensor failure the TCP of the sensor should be determined. For methods of TCP calibration of laser stripe sensors see [9].

The sensor has an ethernet connection to the IPC to control the sensor and to transfer the measurement data. The robot has to execute a measurement while traversing a measurement point. To do this the sensor has to support a hardware trigger feature to ensure measurement at realtime speed. The hardware trigger input of the sensor is therefore directly linked with a digital output module in the robot controller. This way the robot is able to execute measurements directly out of the robot control language even when using approximate positioning moves.

3 System Setup

During system setup the work object is posed at the nominal position in the robot cell. The application program of the robot is created first, which is the basis for the training and measurement programs too. From all path points in the application program the robot programmer chooses the path points which should be corrected with the path correction system. On all these correction points a training run is executed. As the first step of a training run a measurement gives the nominal sensor values $\mathbf{s_0}$ at the nominal position $\mathbf{p_0}$ of the correction point. Then the local relation between robot movements and resulting sensor values changes is determined by executing a series of elemental movements around $\mathbf{p_0}$ and by measuring at these points. Fig. 3 shows as an example the results of training movements of $\pm 4\,\mathrm{mm}$ in x. The sensor values at $x = 0$ are the nominal sensor values. A set of three points is linearly interpolated, the slope of the line is the partial derivative of a sensor value in x, in the example:

$$\frac{\partial s_1}{\partial x} = -0.58 \quad \frac{\partial s_2}{\partial x} = +0.38 \quad \frac{\partial s_3}{\partial x} = +0.01 \qquad (2)$$

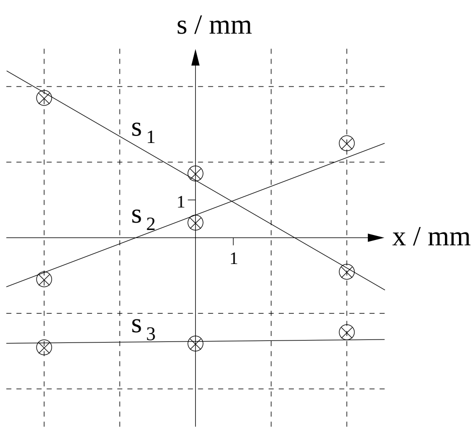

Fig. 3: Training in x-direction

Arranging the partial derivatives for all sensor values and all degrees of freedom to be corrected (up to six: three translational x, y, z and three rotatory α, β, γ) in form of a matrix gives the *sensitivity matrix* $\mathbf{S}$, for example:

$$\mathbf{S} = \begin{pmatrix} \frac{\partial s_1}{\partial x} & \frac{\partial s_1}{\partial y} & \frac{\partial s_1}{\partial z} \\ \frac{\partial s_2}{\partial x} & \frac{\partial s_2}{\partial y} & \frac{\partial s_2}{\partial z} \\ \frac{\partial s_3}{\partial x} & \frac{\partial s_3}{\partial y} & \frac{\partial s_3}{\partial z} \end{pmatrix} \qquad (3)$$

To calculate correction values the relation between sensor values and robot movements has to be known which is represented by the inverse of $\mathbf{S}$, called the *Jacobian matrix* $\mathbf{J}$:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial s_1} & \frac{\partial x}{\partial s_2} & \frac{\partial x}{\partial s_3} \\ \frac{\partial y}{\partial s_1} & \frac{\partial y}{\partial s_2} & \frac{\partial y}{\partial s_3} \\ \frac{\partial z}{\partial s_1} & \frac{\partial z}{\partial s_2} & \frac{\partial z}{\partial s_3} \end{pmatrix} \qquad (4)$$

Because $\mathbf{S}$ is not necessarily a square matrix, $\mathbf{J}$ can be calculated using the Moore-Penrose inverse of $\mathbf{S}$:

$$\mathbf{J} = \mathbf{S}^{+} = \left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{S}\right)^{-1} \cdot \mathbf{S}^{\mathrm{T}} \qquad (5)$$

The Moore-Penrose inverse is usually calculated using the singular value decomposition (SVD) of $\mathbf{S}$. The SVD helps with the analysis of the condition of the sensitivity matrix and to detect ill-posed configurations, e.g. degrees of freedom that should be corrected but cannot be detected with the current sensor configuration. For calculation of the SVD take a look at the books [2] and [8].

Result of the training process are the nominal sensor values $\mathbf{s_0}$ at the nominal position $\mathbf{x_0}$ and the Jacobian matrix $\mathbf{J}$ for each correction point.

4 System Operation

Using offline path correction the complete production cycle is separated into a measurement run and an application run. During the measurement run the robot moves with the sensor as its tool along the work object and triggers the measurements on each correction point. After calculation of the corrections and transfer to the robot the applications run can start with the real application tool.

First step for the calculation of a single correction is to determine the deviation of the measured sensor values during the measurement run $\mathbf{s}$ from the sensor values at the nominal position $\mathbf{s_0}$:

$$\Delta\mathbf{s} = \mathbf{s_0} - \mathbf{s} \qquad (6)$$

The correction $\Delta\mathbf{x}$ relative to the nominal position $\mathbf{x_0}$ is now obtained by a multiplication with the Jacobian matrix:

$$\Delta\mathbf{x} = \mathbf{J} \cdot \Delta\mathbf{s} \qquad (7)$$

The calculation above requires the robot to position the sensor exactly in the nominal position at each correction point during the measurement run. So it would be unavoidable for the robot to stop at each correction point for measurement. This unnecessarily slows down the measurement run. Standard industrial robots support the concept of approximate positioning. Instead of an exact positioning at each path point the robot approaches the path point and before reaching it the robot heads for the next path point. How exact a path point is reached can be configured with different criteria, e.g. with a distance criterion. Approximate positioning changes the robot path from a polygon to a line sections connected with parabolas leading to faster robot motion.

When using approximate positioning during measurement run, there is no significant loss of accuracy if the approximate positioning happens in a direction that is not corrected by the path correction system, for example if there are multiple path points in a line along an object edge and the robot uses approximate positioning for the inner points but the correction is calculated for the two directions orthonormal to the edge. In other cases a special compensation for the approximate positioning is needed. It is possible for the robot to get its current position $\mathbf{x}$ when triggering the sensor and to send this information to the path correction system. With this information the calculated correction including approximate positioning compensation is:

$$\Delta\mathbf{x} = \mathbf{J} \cdot \Delta\mathbf{s} - (\mathbf{x_0} - \mathbf{x}) \tag{8}$$

5 Related Work

There are numerous works on online controlled robots with different sensors like [1], [4], [5], [6], [7] or [12] to mention some important. They connect the robot controller and sensors of different kind in a close control loop to reduce position errors of the robot. Offline path correction systems by contrast are not that popular, the number of publications is much smaller.

An offline correction system that uses a similar to our approach is described in the patent [3]. It uses a laser stripe sensor and a pre-determined Jacobian matrix for correction of calculations too, but provides no approximate positioning compensation. Application purpose of this patent is the automotive industry.

The work of [11] states a correction system for the robot end effector and therefore does a global correction of the end effector and does not refer to a work object. The sensors used are ultra sound sensors which are much cheaper than laser stripe sensors but less accurate and have their problems in environments with changing temperatures and movements of air. The sensor calibration uses linearisation with sensitivity and Jacobian matrices too.

6 Conclusion

The offline path correction system described in this paper has been implemented in an industrial application and has proofed his usefulness in practice. Based on the special requirements the system has to meet and the design decisions we made it has different advantages and disadvantages compared to online path correction solutions.

The most important general advantage of offline path correction systems is that because of the separation of measurement and application both processes do not influence each other. This is interesting for applications like welding or sealing where the measurement can be affected by bright light, dirt or sealing paste. The other way around the application process does not have to be considerate of the measurement process, e.g. in respect of the movement speed of the robot or the need of synchronous movement of sensor and application tool. In offline path correction systems the location of the measurement and of the application can easily be the same. Offline path correction systems are less time critical applications (no real time control) with lower hardware requirements and they offer a very good traceability of the correction process because every correction can be mapped exactly to a robot path point. This helps to increase safety and stability of the system.

The division of the production process into measurement and application run can be an advantage or a disadvantage depending on the application. If the application run itself is separated into multiple steps (like welding, cleaning and inspection of the weldseam) possibly involving multiple robots, the corrections of an offline correction system can be used multiple times. In cases where the application run is just a single run, the offline path correction solution is usually more time consuming.

On the downside an offline path correction solution usually obtains a lower accuracy. One reason is that the robot path is sampled into discrete correction points with a lower density than this would be possible with online correction systems. Provided that the sensor and the evaluation are fast enough, an online solution is able to feed the robot controller with new corrections every single interpolator (IPO) cycle[1], reaching a high density of correction points. So this is an important fact especially for work objects with high tolerances. Another reason is the robots accuracy. If the same robot moves twice to the same position, once for measuring, once for application the repeatability of the robot has to be taken into consideration which is quite good with perhaps one or two tenths of a millimeter. But if one robot does the measurement and another the application, the absolute accuracy of the robot is of importance, which adds up to multiple tenths of a millimeter.

Besides the general pros and cons of offline correction solutions the main characteristics of the system we presented in this paper is its ability to self-calibrate which makes the system flexible and setup process easy and the fast measurement feature with compensation of the approximate positioning.

References:

[1] M. Fridenfalk, G. Bolmsjö: *Design and validation of a sensor guided robot control system for welding in shipbuilding*; International Journal for the Joining of Materials; 14(3/4):44-55; 2002

[2] G. H. Golub, C. F. Van Loan: *Matrix Computations*; The Johns Hopkins University Press; Third Edition 1996; Chapter 8.6, Pages 448-456

[3] B. Huang, V. Milenkovic: *Method and system for correcting a robot path*; U.S. Patent Number 4945493; applied for 1988-09-26, received 1990-07-31

[4] J. P. Huissoon, D. L. Strauss: *System and method for tracking a feature on an object using a redundant axis*; U.S. Patent Number 5465037; applied for 1994-01-13, received 1995-11-07

[5] F. Lange, G. Hirzinger: *A universal sensor control architecture considering robot dynamics*; International Conference on Multisensor Fusion and Integration for Intelligent Systems MFI 2001; Pages 277-282; 08/2001

[6] N. Miyake, M. Sumita, S. Sarugaku: *Method and apparatus for controlling tracking path of working point of industrial robot*; U.S. Patent Number 4954762; applied for 1990-01-30, received 1990-09-04

[7] N. Nayak, A. Ray: *An integrated system for intelligent seam tracking in robotic welding Part I+II*; Proceedings of the IEEE International Conference on Robotics and Automation, 1990; Pages 1892-1903; 1990

[1] usually around 10 ms for todays robot controllers

[8] W. H. Press, A. S. Teukolsky, W. T. Vetterling: *Numerical Recipes in C*; Cambridge University Press; Chapter 2.6: Singular value decomposition; 01/1993

[9] Philipp Roebrock: *TCP Identification of Contactless Measurement Systems*; Symposium of Electronics and Telecommunications; "Politehnica" University of Timişoara/Romania; 09/2006

[10] B. Triggs, Ph. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon: *Bundle Adjustment - A Modern Synthesis*; Springer Verlag; 2000

[11] Y. Veryba, L. Kourtch, B. Verigo, V. Murashko: *Method and system for robot end effector path correction using 3-D Ultrasound sensors*; Proceedings of the 3rd World Congress on Intelligent Control and Automation; Pages 1240-1243; 2000

[12] J. Zeller: *Sensorplanung und schnelle Sensorregelung für Industrieroboter*; Hanser Fachbuchverlag; 01/1998