Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments

Thomas Dietz, Ulrich Schneider, Marc Barho, Susanne Oberer-Treitz, Manuel Drust, Rebecca Hollmann, Martin Hägele
Fraunhofer Institute for Manufacturing Engineering and Automation, Germany

Abstract

The use of robots in SME production environments requires programming systems that decrease programming effort and can be used by process experts rather than robotics experts. The core sources of information available to the robot system are human input, sensor data and model data. However most programming systems only connect one or two of these information sources to generate robot programs. The presented programming concept featuring hand-guided robot operation for intuitive robot instruction, use of workpiece CAD-models to retrieve nominal model information and a triangulation sensor for detection of workpiece contours features combination of all three sources of information and promises efficient robot programing for deburring applications in SME production. Finally a kinematic compensation approach is investigated which provides good positioning accuracy which is in particular important when dealing with workpieces of large dimensions.

1 Introduction

Robots are vital tools allowing production at low cost and with high quality. Although robots are now widely used in mass production they still cannot be used cost-effectively for a wide range of production tasks in small and mediums sized enterprises (SME) [10]. As SMEs typically produce in smaller lot sizes and produce many product variants the production processes are characterized by the requirement for frequent changeover and reconfiguration. This requirement can also be expected to make an imprint on mass production in the years to come as the trend for product individualization to customer requirements finds its way into mass production. While in mass production the robot system is in use for a long production phase after being developed and commissioned the lifecycle of robot systems in SME production is characterized by frequent reconfiguration and changeover as illustrated in figure 1. Therefore, in mass production the focus for increasing the efficiency of the robot system is mostly on improving the performance for the production task. Opposed to that the complete cycle of reconfiguration and production needs to be taken into account for SME production.

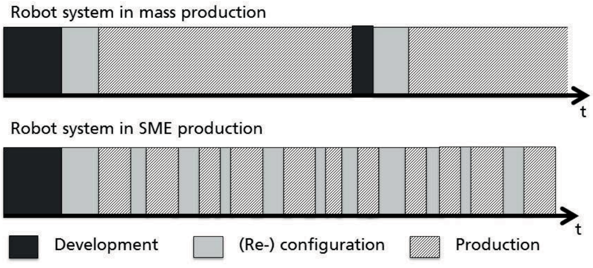
Robot system in mass production
t
Robot system in SME production
t
Development
(Re-) configuration
Production

Figure 1: Comparison of phases of robot use in mass production and SME production.

To allow rapid reconfiguration of robot systems to new products and to make that reconfiguration process reasonably simple so it is workable for the robot operator the focus has to lie on the adaptation of the robot programming tool. Today robot users either depend on online lead-through programming methods where the robot is programmed in the field by jogging it with a teach pendant and locking program points manually. While the basics of lead-through programming can be learned very quickly a lot of experience is required to generate good programs as the interface cannot offer information on the environment of the robot. On the other side online programming tools are used for the planning and programming of mass production lines. While offline programming tools offer advanced functionality operators are required to undergo extensive training to use them. Further the investment cost is usually significant [13]. The problem with both approaches for robot programming is that they are usually not tailored for the process executed by the robot and therefore do not offer specific functionality to speed up programming. Therefore a new generation of robot programming tools is currently developed in research that will improve the cost effectiveness of robots in SME production dramatically. This paper surveys the use of advanced programming systems for a deburring use-case commonly found in SME production.

Section 2 will discuss the investigated process and analyze basic requirements for its execution. The generation of robot programs and related technological aspects are discussed in section 3. This chapter addresses intuitive methods for obtaining human input, sensor, input from other sources such as workcell and product models and technologies to combine the information from different input sources into a robot program. Section 4 focuses in particular on big workpieces providing an approach to increase global robot accuracy. Section 6 presents conclusions and gives an outlook.

2 Process Description

Deburring is one of the most common robotic processes and has been widely studied in research [21, 14, 17, 11]. The high reach and orientation capability of industrial robots can be perfectly utilized for this process. For example it is commonly found in finishing tasks in the casting industry. Foundries usually manufacture a very large spectrum of cast parts which can be produced depending on customer demand. Deburring operations are used for finishing the cast parts before delivery, but also for finishing sand cores used for casting prior to the casting process. The large part spectrum prohibits automation without flexible programming systems as typical production volumes are small. Following the core technical requirements of a deburring system for SME foundry production resulting from end-user and system integrator interviews are laid out:
Technical core requirements:

- Average machine operator needs to be able to use the system with less than two hours of training.
- Programming system has to correct user input based on burr location.
- Workpiece referencing process needs to be at least partly assisted.
- System has to be able to operate in harsh and dusty environment.
- System needs to be robust to variations of thickness and shape of the burr.
- Safety measures for human-robot-collaboration have to be effective and compliant with current standards.

These requirements can only be matched through a combination of model information, sensor data and human input. Human input is required to structure the overall task, e.g. to guide the robot close to burrs and define edges to be deburred. Sensor data is required to track the exact location of the workpiece and model information is required to establish a relation between sensor data and human input to correct the imprecise motion of the operator.

3 Program Generation System

Figure 2 shows three types of information that are vital for program generation and how they are interrelated. On the one hand knowledge about the structure of the environment and nominal properties of devices and objects involved allows building a model of the workcell. The core advantage of offline programming lies in the ability to use this kind of model for programming. This allows the extraction of information that can be used for program generation. The models are usually built from CAD data and mathematical/parameter-based descriptions of the involved objects. Secondly the actual properties of the workcell define the reaction of the real-world system to any action. Actual properties can be measured with some degree of uncertainty and to a limited extend by using sensors. Thirdly the structure of the tasks defines how the robot is intended to execute the process. Usually this type of information is obtained from human input as humans are capable of structuring complex tasks and teach them to the robot. Offline programming tools usually rely on model information and human input to generate tag points and robot programs. However, programs generated from offline programming systems almost always require calibration on the real-world before they can be used for production. During calibration inaccuracies resulting from deviations between the model of the environment and the real-world system are corrected. This means that with offline tools the actual environment cannot be taken into account during programming. Online lead through programming relies on human input and the reaction of environment to the actions of the robot system. These reactions are either detected through sensor systems or the human programming the robot. However, model information cannot be used in this programming method. All three aspects, i.e. model, actual measurement and human input on task structure play a vital role for programming. For SME production pure offline programming is usually not feasible due the high cost and specialized training required. Therefore it appears to be logic to combine all three aspects in a single programming system to program the robot on the shop floor.

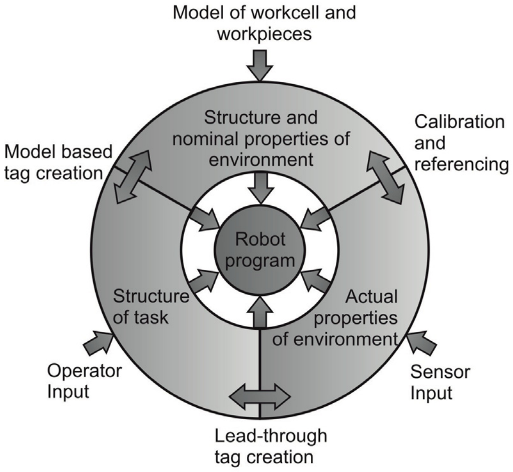
Model of workcell and workpieces
Structure and nominal properties of environment
Calibration and referencing
Model based tag creation
Robot program
Structure of task
Actual properties of environment
Operator Input
Sensor Input
Lead-through tag creation

Figure 2: Relation of different input methods for robot programming.

3.1 Human Input

As mentioned the human is capable of breaking down and structuring a mission into tasks and actions that can be understood and executed by the robot. Therefore a human operator is required for complex robot operations. For online programming systems human input is collected using standard PC-interfaces such as keyboard, mouse, 6D-mouse. For online programming the state of the art is jogging of the robot through buttons or 6D-mouse and touchscreen operation of the robot controller. Both methods are not natural instruction methods for humans and need to be learned before being able to use the robot.

A very intuitive input method that can be learned in a few seconds is programming by demonstration through hand-guiding. This input methods allows the operator to move the robot by guiding it with a handle mounted to its end-effector. The forces applied by the operator are measured and transformed into robot motion. During the motion the travelled path is recorded and stored by the robot controller and is subsequently available for post-processing.

The core aspect of programming by demonstration is to ensure safety for the operator. As specific limits for the robot's speed have recently been dropped from the robotics safety standards ISO 10218 [7] the burden for finding suitable thresholds is now increased on the side of end-user and system integrator. Injury mechanisms and the quantification of the resulting hazard level are generally very complex [15]. Although the severity of collisions between human and robot has been addressed in research [12, 3] it yet has to be broken down to be usable for end-users in robotic applications. For the programming system at hand this is accomplished by measuring the impact forces and pressure during collision using a device specially developed for this purpose (see figure 3). In this process it is possible to obtain an upper threshold value for the robot's speed.

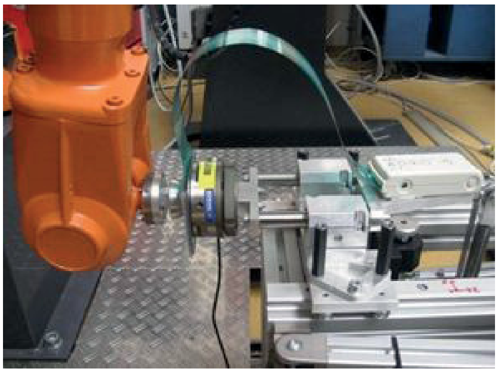

Figure 3: Device for measuring impact forces and pressure in human robot collaboration.

3.2 Sensor Input

To feed information about the workpiece and burr back to the robot system for program adjustment a sensor system capable of withstanding the harsh environment conditions of foundries is required. Line laser triangulation sensors are also used for welding tasks [9]. This type of sensor was chosen is favor of a camera based system due to its good robustness with respect to environmental conditions and the ease of integration into the robots end-effector. During programming by demonstration the sensor is used to record the height profile of the workpiece. Commercial software packages allow edge and surface detection .

3.3 Model Input

In order to use the sensor data for program generation for the deburring process it has to be set relation to the expected shape and properties of the workpiece and the robots environment. For the investigated deburring operation it is assumed that the robot is strong enough to withstand the process forces to remove the complete burr without the need for several passes. Therefore the deburring process can be observed purely geometrically. For this purpose a CAD model of the workpiece is used. The used CAD-library CADability is used to extract the desired information from the CAD model. It has to be noted that the deburring process may rely on CAD models as they usually are or can be expected to be widespread available in coming years for SME casting production.

3.4 Input Fusion for Program Generation

The core of the investigated programming system is how to set the three sets of information - human input, sensor data and CAD data - into relation to ease the programming process. Figure 4 shows the use of the information of the system for referencing of the workpiece. The worker selects an edge of the workpiece to be used for referencing in the cell model. Subsequently the robot is moved to the respective edge by hand-guiding. The sensor data from the robots position measurement system and the triangulation sensor in combination with edge detection can then be used to find the position of a point that is on the selected edge. This constraint is added to the model. The process is repeated until the workpiece is localized. The procedure can also be applied with flat surfaces of the workpiece to establish additional constraints.

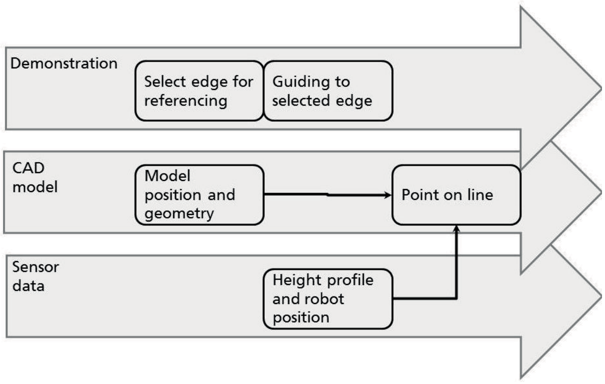
Demonstration
Select edge for referencing
Guiding to selected edge
CAD model
Model position and geometry
Point on line
Sensor data
Height profile and robot position

Figure 4: Process for finding reference points on the workpiece.

After the workpiece is referenced the process shown in figure 5 is applied to program the deburring process. Here the operator guides the robot close to the edged to be deburred. The position information of the robot is used in the CAD model retrieve the nominal shape of the part after deburring. This information can be used by the program generation system to correct the recorded robot position and orientation to optimize the burr removal. In combination with the data from the line sensor the burr removal strategy regarding robot orientation can be improved and safety checks for validation of correct referencing and workpiece tolerances can be applied. The information can further be used for quality control and documentation.

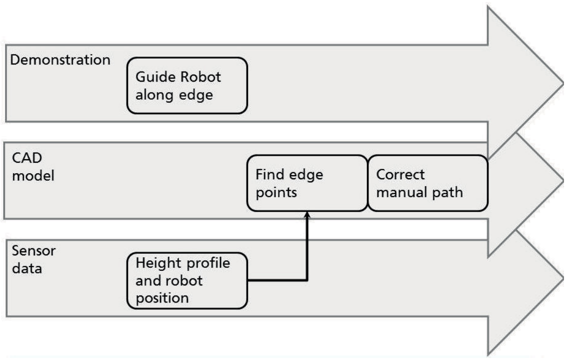

Figure 5: Process for programming of deburring task.

4 Improving accuracy of industrial robots by kinematic compensation

One major benefit of using industrial robots in deburring applications is there huge work space. Compared to conventional tooling machines industrial robots are much more flexible and can easily deal with dimensions up to 2 m. However for big workpieces the referencing approach described in Section 3.2 only guarantees local accuracy. In the following Section a kinematic compensation approach is described providing not only local but global accuracy for industrial robots [5].

4.1 Kinematic descriptions

In order to move robots cartesianly in space a transformation from the base of the robot to its effector is set up containing the geometry of the robot and the movements of the six rotational joints. The change from the frame of one joint to the following is described with one transformation. Popular descriptions of this transformation are the relativ description [1], the parameter set defined by Denavit and Hartenberg [2] and a description introduced by Hayati [4]. Whereas Denavit, Hartenberg and Hayati use four parameters to describe this transformation, the relative description uses six parameters. Vaishnav and Magrab [20] even propose a model using nine parameters in order to take into consideration non-orthgonal systems.
According to [16] good identification properties can be achieved with a combination of the descriptions of [2] and [4].

4.2 Identification of kinematic parameters

After setting up an appropriate model which describes the behavior of the robot precisely and at the same time enables good identification conditions, parameters g of the model need to be identified in order to set up an error compensation.
By means of one taken measurement Pos_{meas} and using

$$m = Pos_{meas} - Pos_{bot} \tag{1}$$

with the robots internal position Pos_{bot}, constraints can be set up describing the unknown kinematic parameters g:

$$m = F(g) \tag{2}$$

where $F(g)$ represents a function of the kinematic parameters g. As in general one measurement supplies less conditions than unknown parameters, several measurements need to be taken at different robot poses. This overconstrained system can then be solved by minimizing a norm. Most commonly least squares are used. Alternatively a Kalmanfilter can also solve the problem [18].
In this case the pseudoinverse is used to solve the overconstrained system. Detailing Equation 2 and separating $F(g)$ one receives for one measurement m_l

$$m_l = A_l(g_l)dg \tag{3}$$

where dg is the difference between the assumed and the true kinematic parameters and A is the Jacobian containing the derivatives of the measured coordinates with respect to the kinematic parameters:

$$A_l = \begin{pmatrix} \frac{\partial Pos_1(g)}{\partial g_1} & \cdots & \frac{\partial Pos_1(g)}{\partial g_i} \\ \vdots & & \vdots \\ \frac{\partial Pos_j(g)}{\partial g_1} & \cdots & \frac{\partial Pos_j(g)}{\partial g_i} \end{pmatrix} \tag{4}$$

with i the number of kinematic parameters and j the number of measurement signals. Using Equation 3 for one measurement we receive for k measurements

$$M = Sdg \tag{5}$$

with

$$M = \begin{pmatrix} m_1 \\ \vdots \\ m_k \end{pmatrix} \tag{6}$$

and with the sensitivity matrix S

$$S = \begin{pmatrix} A_1(g_1) \\ \vdots \\ A_k(g_k) \end{pmatrix} \tag{7}$$

Using the pseudoinverse S^+ of S leads to

$$dg = S^+ M \tag{8}$$

The actual kinematic parameters can then be calculated as

$$g = g_{ini} + dg \tag{9}$$

with the initial kinematic parameters g_{ini}.

4.3 Evaluation of robot poses for identification

However an appropriate model and a good identification algorithm as described in Sections 4.1 and 4.2 don't guarantee good identification results. The identifiability of the kinematic parameters strongly depends on the chosen robot poses [18] [8]. Different observability indeces have been proposed evaluating the set up conditions [18] [19]. Most of these indeces refer to the singular values of the sensitivity matrix S in Equation 7.

5 Implementation and Experimental Results

The strategy presented in Section 4 will be verified in experiments. The following Subsection will describe the components of the experiments and the experimental setup.

5.1 Components and parameters

A robot cell is build up and a robot is equipped with the necessary hardware for milling. The model identification will be done in the target environment in order to cover all possible sources of errors. A KR125 robot from KUKA is used together with a TwinCAT CNC-control of BECKHOFF where updated kinematic parameters can be implemented easily. As the robot is 13 years old, wear contributes to a limited precision in the case without compensation mechanisms.

In essential component of model identification is the measurement equipment. The quality of the identified models depends strongly on the precision of the measurement device. Especially due the large work space of the robot and the required precision only a few devices are worth to be considered. A Leica Absolute Tracker AT901 is chosen because of its three dimensional measurements, its tracking functionality and its error of $err_{LT} < 20\mu\text{m}$ for the chosen area.

5.2 Identification of the kinematic model

Robot positions used for identification of kinematic parameters are chosen so that they represent the work space which is relevant for machining.

The result of the optimisation procedure is presented in Table 1.

Kinematic parameters	Initial value	Offset
θ_1	-	+0.00917 °
θ_2	-	-0.10019 °
θ_3	-	+0.04798 °
θ_4	-	+0.05242 °
θ_5	-	-0.21941 °
θ_6	-	-0.21941 °
L_1	865 mm	+1.01343 mm
L_2	410 mm	+1.48882 mm
L_3	1000 mm	+0.39679 mm
L_5	1000 mm	-0.50421 mm

Table 1: Offsets of kinematic parameters resulting from optimisation

5.3 Positioning improvements

The positioning accuracy for industrial robots, also named absolute accuracy, is defined in [6]. Besides the definition of accuracy a representative set of measurement points is indicated.

According to the procedure described in [6] measurements were performed for the uncompensated and the compensated robot (see Table 2).

	uncalibrated robot	calibrated robot
Point 1	5.70 mm	0.54 mm
Point 2	5.55 mm	0.15 mm
Point 3	5.64 mm	0.41 mm
Point 4	5.63 mm	0.43 mm
Point 5	5.53 mm	0.38 mm

Table 2: Positioning errors of the uncalibrated and the calibrated robot according to [6]

6 Conclusions and Outlook

For the use of industrial robot in typical SME processes a reduction of the programming effort is required to allow cost-effective deployment also for medium and small lot sizes and series volumes. The programming has to be sufficiently intuitive to be carried out by machine operators without special training. To achieve this goal information on the nominal structure of the robots environment, sensor data reflecting the actual properties of the environment and human input for coordination of the robots tasks need to be combined in smart manner. The elementary building blocks (hand-guiding, CAD-model data extraction, edge detection from triangulation sensor data) of the described system have been implemented and tested. This paper describes a novel approach on how to connect these building blocks to create a programming system for finishing tasks in SME-production. Additionaly the deburring of large workpieces has been investigated. A kinematic compensation approach has been developped and evaluated.

Acknowledgements

Part of the research work reported here is supported by the European Comission under the Seventh Framework Programme (FP7/2007-2013) within the Project COMET (http://www.cometproject.eu/) under grant agreement #258769, the Project ROSETTA (http://fp7rosetta.org/) under grant agreement #258769 and the Project SMEROBOTICS (http://www.smerobotics.org/) under grant agreement #287787.

References

[1] Craig, J. Introduction to robotics: mechanics and control.

[2] Denavit, J. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics 22* (1955), 215–221.

[3] Haddadin, S., Albu-Schaffer, A., and Hirzinger, G. Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research 28*, 11-12 (2009), 1507–1527.

[4] Hayati, S., and Mirmirani, M. Improving the absolute positioning accuracy of robot manipulators. *Journal of Robotic Systems 2*, 4 (1985), 397–413.

[5] Hollerbach, J., and Wampler, C. The calibration index and taxonomy for robot kinematic calibration methods. *The international journal of robotics research 15*, 6 (1996), 573.

[6] ISO, I. 9283: manipulating industrial robots-performance criteria and related test methods, 1998.

[7] ISO 10218-1:2011. Robots for Industrial Environments - Safety Requirements - Part 1: Robot. Tech. rep., International Organization for Standardization, Geneva, Switzerland, 2011.

[8] Li, T., Sun, K., Xie, Z., and Liu, H. Optimal measurement configurations for kinematic calibration of six-dof serial robot. *Journal of Central South University of Technology 18*, 3 (2011), 618–626.

[9] Meyer, C. *Aufnahme und Nachbearbeitung von Bahnen bei der Programmierung durch Vormachen von Industrierobotern*. PhD thesis, 2011. 129 S.

[10] Meyer, C., and Schraft, R. D. An intuitive teaching method for small and medium enterprises. In *Intelligent Production Machines and Systems* (2006), Elsevier, Amsterdam u.a., pp. 568–571.

[11] Murphy, K. M., Norcross, R. J., and M., P. F. CAD directed robotic deburring. In *Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education and Applications* (1988).

[12] Oberer-Treitz, S., Puzik, A., and Verl, A. Measuring the collision potential of industrial robots. In *Proceedings for the joint conference of ISR 2010 and ROBOTIK 2010* (2010).

[13] Pan, Z., Polden, J., Larkin, N., Duin, S. V., and Norrish, J. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing 28*, 2 (2012), 87 – 94.

[14] Pan, Z., and Zhang, H. Robotic machining from programming to process control: a complete solution by force control. *Industrial Robot: An International Journal 35*, 5 (2008), 400–409.

[15] Schmitt, K.-U., Niederer, P., Muser, M., and Walz, F. *Trauma Biomechanics - Accidental Injury in Traffic and Sports*. Springer, Zürich, 2010.

[16] Schröer, K. *Identifikation von Kalibrationsparametern kinematischer Ketten*. PhD thesis, Technische Universitat Berlin, 1993.

[17] Shimokura, K., and Liu, S. Programming deburring robots based on human demonstration with direct burr size measurement. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on* (may 1994), pp. 572 –577 vol.1.

[18] Siciliano, B., and Khatib, O. *Springer handbook of robotics*. Springer-Verlag New York Inc, 2008.

[19] Sun, Y., and Hollerbach, J. Observability index selection for robot calibration. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* (2008), IEEE, pp. 831–836.

[20] Vaishnav, R., and Magrab, E. A general procedure to evaluate robot positioning errors. *The International journal of robotics research 6*, 1 (1987), 59.

[21] Zhang, H., Chen, H., Xi, N., Zhang, G., and He, J. On-line path generation for robotic deburring of cast aluminum wheels. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (oct. 2006), pp. 2400 –2405.