Small Batch Size Robot Programming with Human in the Loop

Béla Takarics, Gábor Sziebig, Péter Korondi

Abstract:

Robot programming methodologies are moving from the traditional to more human friendly, rapid and flexible programming alternatives. In this paper a new robot programming methodology is introduced. The proposed programming methodology physically disconnects the robot programmer from the robot itself. Applications of the proposed methodology in grinding and welding operations with experimental results as well are presented.

Keywords: *robot programming, vision system, sme robots, human-machine communication.*

1. Introduction

Increasing requirements to productivity and working environment push the need for industrial robots in the industry. This is especially important in heavy physical work were sickness, absence and recruiting problems are often the most limiting factors on the productivity. Today, industrial robots represent the best solution for both productivity and flexibility.

In manufacturing engineering, man-machine interaction has gone from typical online programming techniques into virtual reality based offline programming methodologies [1]. Today, a wide range of offline software tools is used to imitate, simulate and control real manufacturing systems [2]-[5]. However, also these new methodologies lack capability when it comes to human-machine communication.

Typically, today communication with the virtual environment is done *via* a keyboard/mouse interface while feedback to the operator is given from the computer screen. These desktop reality systems uses only a limited spectre of human senses and there is quite a low sensation of "being inside" the system. Programming is still done on the premises of the machines.

As humans receive most of their information visually, significant part of the neurons in the human brain is dedicated to visual processing; probably the best way of programming robots is visual programming, without any restrictions. Previously visual programming was only used in computer programming [6], [7], but with new series of robots and robot controllers a new era in robot programming had started [8]. Mentioned above, these solutions are usually limited to keyboard/mouse input and feedback [9].

Thus, this article focuses on a new interactive robot programming methodology where the knowledge and flexibility of the human operator is combined with information from CAD models and simulations of real robot movements. This 3D interactive environment is especially suitable for planning robotic operations.

The organization of the paper is as follows: Section 2. gives an overview of the programming methodology while Section 3. and 4. presents the application of this methodology in two different robotic cases (grinding and welding) including experimental results. Section 5. concludes the paper.

2. Methodology

By introducing modern technologies, like motion capturing and augmented reality principles, with the goal of adding human representations into the programming environment it is possible to create a new programming concept where the human operator can interfere with machines in a cognitive manner. Thus, the key-idea of the proposed methodology is to capture the knowledge of a skilled operator and make an automatic knowledge transfer to a robot system. Fig. 1. summarizes the proposed methodology.

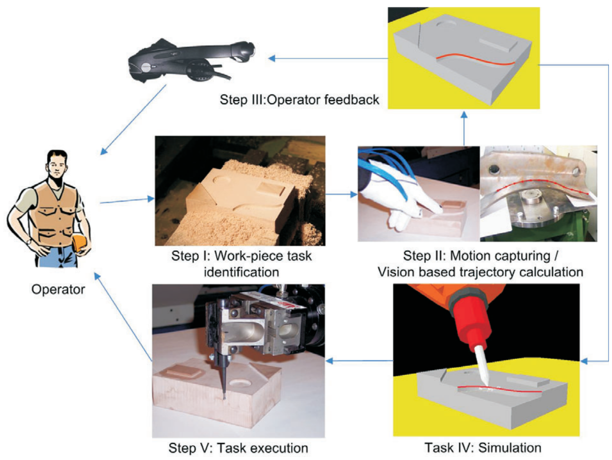

Fig.1. Overview of methodology.

The proposed methodology allows a human operator to freely move around in an industrial environment (e.g. shop-floor) and identify the work-pieces which are necessary to modify via some robot action. As before mentioned, such a typical situation could be a robot grinding process following a moulding process where the work-pieces suffer from some irregularities (burrs) or a welding operation, where the trajectories may differ from work-piece to work-piece. By sight, the human operator can easily identify the area of modification but cannot exactly quantify the error.

The methodology is composed from the following steps:
Step I: An operator based on his/her own expertise decides, which work-piece needs modification. This decision

can also be supported by measurement systems (e.g. CMM machine) and CAD drawings. Not only the work-piece itself, but also the necessary task is chosen (e.g. grinding, welding).

Step II: The operator highlights the modifications with using one of the following tools: a) using a software component on a PC connected to a camera vision system the operator identifies the necessary path for the operation (using keyboard/mouse) b) the operator is connected to a motion capture system, which allows free movement in the shop-floor environment and by pointing on the surface of the given work-piece the rough path is stored on a PC.

Step III: The path from the previous step is transferred onto to the ideal work piece (e.g. CAD-model in a simulation environment) in order to verify that the defined path is located as expected. The path is automatically projected (as a line) onto the surface of the ideal CAD model. By transferring the robot path directly onto the CAD model actually defines what the final end-geometry should look like. Further, the line from the CAD model is fed-back to the operator either through computer screens or using a Head Mounted Display (HMD). This ability of instant feed-back makes it possible for the operator to relocate the operation path according to what he/she sees in the screens. By placing the operator "into the loop" of adjusting/relocating the generated path, the accuracy requirements of the human-machine interfaces (e.g. mouse, motion capture system) are minimized.

Step IV: Simulation of the complete robot motions are carried out in order to check for singularities, out of range limitations, and possible collisions.

Step V: The robot operation is executed.

3. Monocular vision based trajectory generation

Instead of a costly, task specific, but fully automated solution (e.g. cast part grinding/deburring), which works only from CAD model of the work-piece and other low level sensory input (e.g. contact sensor), an operator is involved in the process of robot programming. The result is a 90% automated solution with the expertise of the worker. The proposed vision-based robot programming for grinding applications, adds the required information into the Offline programming environment. Thus, location and shape of any irregularities can be identified and the necessary grinding/deburring trajectory created. It is based on a single camera facing the grinding surface, a software component for defining the grinding path and an Offline simulation environment.

Following the methodology described in Section 2, the system structure is the following: In the beginning the robot task is defined (grinding) from an image, acquired by a camera facing the work-piece. The goal is to reach the optimum in accuracy and complexity; the robot moves along these points, which describes a geometrical form. This form can be a line, described by a start and an end point, but it can be also a Bezier curve described by 500 or more points. It is the operator's role to make the decision, using a graphical user interface (GUI). Using an operator here is a simplification, but not in the mean of retardation of quality. Instead of accuracy drop, the operator makes the system as accurate (or more) as it could be with using teaching or online programming, but without the time consumption. The simplification makes robot programming not absolutely automated, but it can be done remotely, far away from the robot.

An existing robot simulation solution for the design and offline programming of complex workplaces fitted with several systems, called IGRIP (Interactive Graphics Robot Instruction Program) is used for rest of the steps (Step III - IV). The work-piece data and CAD models are imported to the virtual environment in IGRIP and the simulation is run there. The source code provided by IGRIP is compiled to machine code and uploaded to the robot for machining. More details are discussed in [10].

3.1. Results

Experiments have been carried out with an ABB IRB-2000 industrial robot and a distant operator. For these tests a low resolution web camera (640*480 pixels) were used to capture a hand drawn line (grinding path) on a sculptured surface (work-piece). After the image processing the robot was instructed to follow the generated path. The result is shown in Fig. 2.

a) *b)*

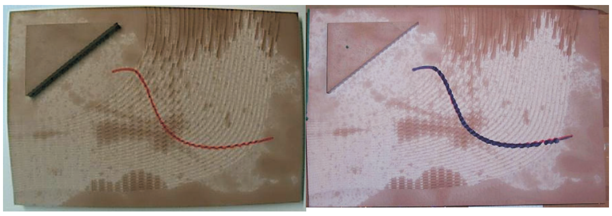

Fig. 2. Ideal hand-drawn path a) and the resulting path b) executed by the robot.

The deviation between the hand drawn and the robot generated answer was approximate 1 mm. A machining accuracy of less than a millimeter is acceptable in grinding and deburring operations.

4. Stereo vision based welding trajectory generation

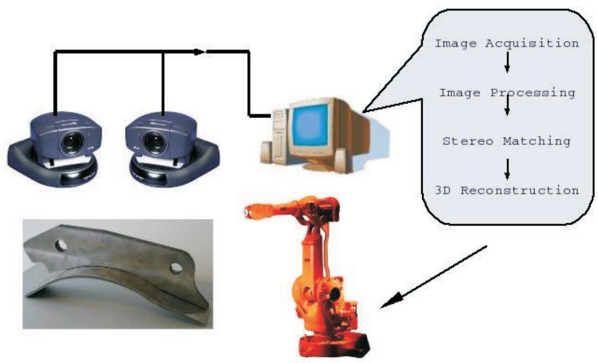

Fig. 3. Welding system overview and data flow.

In the following a fully automated robotic welding application will be introduced. The robot has to weld excavator buckets. The batch size is very small, only few work pieces fall in a batch, which means that conventional robot programming methods are not economic in this case. The welding path is usually obtained from the CAD model of the work pieces. Unfortunately, in this case the CAD models of the excavator buckets cannot be used as references for welding as the tolerances of the work pieces are high. The solution came from methodology mentioned in Section 2. The proposed system has the following components: a vision system, user driven software component and a welding robot (see Fig.3.).

The welding trajectory generation is based on camera vision and image processing. The trajectory reconstruction is done in two steps: first, the welding curve is recognized in the image by image processing (see Fig. 4.), second, the 3D curve is reconstructed with the principles of stereo vision. The vision system was implemented in Matlab and was tested in industrial environment. Simulation software is used to transform the welding path from the world coordinate system to the actual robot coordinate system. This way the whole system is able to discover the welding path in the space and to generate the welding trajectory for a given industrial robot. The trajectory is first checked against errors (e.g. singularities) and only after successful simulation will be executed in real operation. More details are discussed in [11].

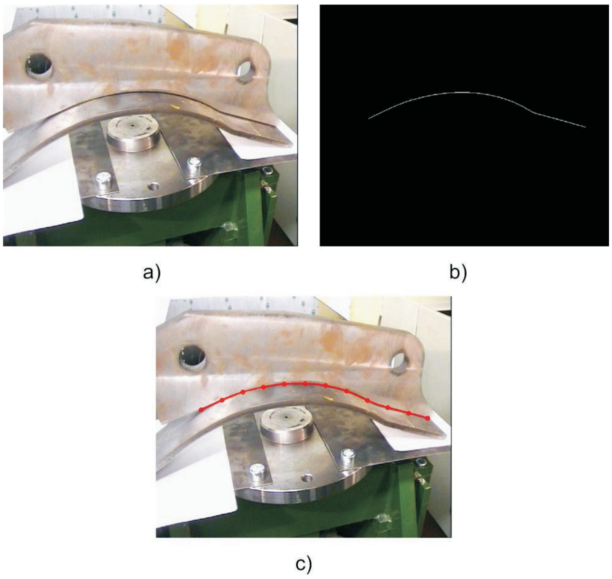

Fig. 4. a) Original image; b) edge detection step; c) spline fit on the original image.

4.1. Results

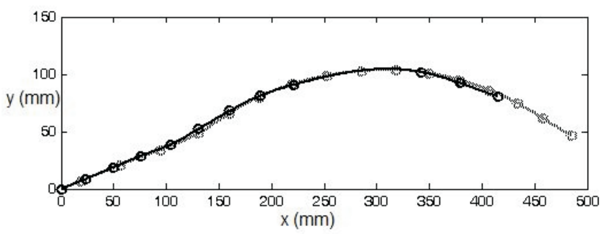

Fig. 5. Comparison of the measured (gray) and the generated (black) trajectory.

The vision system's accuracy was checked in the following way: 20 points of the ideal (expert defined) curve was measured on the real work-piece and from these points a spline is constructed. This ideal curve is compared with the result of the image processing and 3D reconstruction (automatically generated spline). The result can be seen in Fig. 5.

As in this case the welding curve is planar, first the distance between the measured and the reconstructed welding plane was checked. The average distance between the two planes is 1.96 mm. After this the vertical distance between each point of the splines were calculated. The distan-ces are shown in Fig. 6. and the average distance is 1.29 mm.

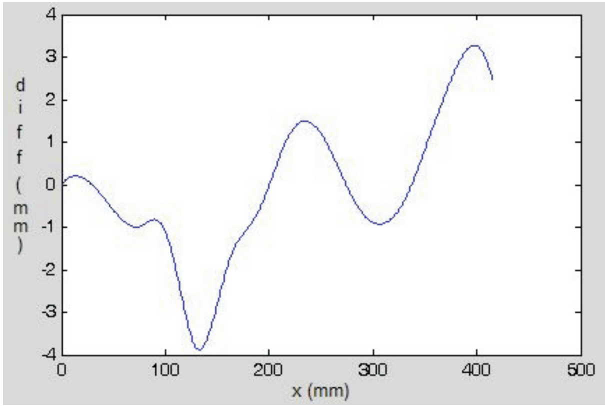

Fig. 6. Distances in millimetres between the measured and generated points in vertical direction.

The accuracy provided by the system is approximately the same as an average industrial welding robot has, thus it is enough for excavator bucket welding. The tests were made only with one type of work-piece, thus there is no experience with different sizes. The work-pieces have to take as much part of the images as possible, this way the image processing system can take full advantage of the camera resolution. It is known that resolution limits the precision of the vision. The cameras are placed at 1 m fixed distance from the work pieces to avoid collision with the robot. Larger distance is unnecessary as the larger zoom infers optical distortions. The system was tested in real industrial environment. This proves its robustness, which is a must for industrial applications.

5. Conclusion

In this paper a new type of robot programming methodology has been presented. The proposed system makes robot programming human friendly, rapid and flexible. The methodology uses advanced human-machine interfaces (e.g. motion capturing) and 3D feedback to the operator that enables the operator to immediately adjust his actions. The generated path can be simulated before machining in order to take account for any limitations in the real robot movements. The operator is physically disconnected from the robot and can move freely around in any shop-floor environment. Results in two important application domains of industrial robots are also shown.

ACKNOWLEDGMENTS

The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency.

AUTHORS
Béla Takarics* - Department of Mechatronics, Optics and Applied Informatics, Budapest Univ. of Technology and Economics, Budapest, 1521, Hungary.
E-mail: takarics@get.bme.hu.
Gábor Sziebig - Department of Industrial Engineering, Narvik University College, Narvik, 8514, Norway.
E-mail: gabszi@hin.no.
Péter Korondi - Computer and Automation Research Institute, Budapest, 1111, Hungary and Dept. of Mechatronics, Optics and Applied Informatics, Budapest Univ. of Technology and Economics, Budapest, 1521, Hungary.
E-mail: korondi@sztaki.hu.
* Corresponding author

References

[1] Sett A., Vollmann K., "Computer based robot training in a virtual environment". In: *Proc. of IEEE International Conference on Industrial Technology (ICIT'02)*, volume 2, 2002, pp. 1185-1189.
[2] Jaramillo-Botero A., Matta-Gomez A., Correa-Caicedo J., Perea-Castro W., *Robomosp. IEEE Robotics & Automation Magazine*, vol. 13, no.4, 2006, pp. 300-302.
[3] Chang-Sei K., Keum-Shik H., Yong-Sub H.H., Soo-Ho K., Soon-Chang K., "Pc-based on-line programming using vrml for welding robots in shipbuilding". In: *Proc. of IEEE Conference on Robotics, Automation and Mechatronics*, volume 2, 2004, pp. 949-954.
[4] Biegelbauer G., Pichler A., Vincze M., Nielsen C., Andersen H., Haeusler K. "The inverse approach of expaint [robotic spray painting]". *IEEE Robotics & Automation Magazine*, vol. 12, no. 3, 2005, pp. 24-34.
[5] Bruccoleri M., D'Onofrio C., Commare U.L., "On-line programming and simulation for automatic robot control software generation". In: *Proc. of 5th IEEE International Conference on Industrial Informatics*, 2007, pp. 491-496.
[6] Okamura T., Shizuki B., Tanaka J., "Execution visualization and debugging in three-dimensional visual programming". In: *Proc. of 8th International Conference on Information Visualisation (IV2004)*, 2004, pp. 167-172.
[7] Broil W., Herling J., Blum L., "Interactive bits: Prototyping of mixed reality applications and interaction techniques through visual programming". In: *Proc. of IEEE Symposium on 3D User Interfaces*, 2008, pp. 109-115.
[8] Han K.S., Wook J.J., "Programming lego mindstorms nxt with visual programming." In: *Proc. of International Conference on Control, Automation and Systems (ICCAS'07)*, 2007, pp. 2468-2472.
[9] Lourens T., "Tivipe - tino's visual programming environment". In: *Proc. of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004)*, volume 2, 2004, pp. 10-15.
[10] Solvang B., Korondi P., Sziebig G., and Ando N. "Sapir: Supervised and adaptive programming of industrial robots". In: *Proc. 11th IEEE International Conference on Intelligent Engineering Systems (INES'07)*, Budapest, Hungary, June 2007, pp. 281286.
[11] Takarics B., Szemes P.T., Nemeth G., Korondi P., "Welding trajectory reconstruction based on the Intelligent Space concept". In: *Proc. of IEEE Conference on Human System Interactions (HSI 2008)*, 2008, pp. 791-796.