Force control experiments for industrial applications: a test case using an industrial deburring example.

Gabriel Afonso[1], J. Norberto Pires[2], Nelson Estrela[3]

[1]Technical School of Viseu, Portugal
[2]Mechanical Engineering Department, University of Coimbra, Portugal
[3]IVO Cutelarias, Portugal

Abstract

This paper reports the development of a robotic system designed for automatic deburring applications. Since a good model of the environment is difficult to obtain, namely on industrial applications, a simple strategy was designed to obtain the relevant parameters leading to an acceptable performance. Consequently, the system implements an indirect force control strategy as a way to use actual robot controllers, explore the computing power of external personal computers, and the advanced features of modern force-torque sensors. The proposed strategy is presented in some detail and further discussed using a few test-case experiments.

I. INTRODUCTION

In many industrial robotic applications, force control [1-7] is an important issue in order to guarantee certain process requisites and the quality of the final product [10]. In this framework, the majority of the robot tasks require contact with the surrounding environment, i.e., in the process of fulfilling the task the robot tool interacts physically with the working objects and surfaces. That interaction generates contact forces that should be controlled in a way to finish the task correctly, not damaging the robot tools and working objects. Those contact forces depend on the stiffness of the tool and working objects/surfaces and should be properly controlled. The option for a particular control technique depends on identifying if [1, 7]:

1) The contact forces should be controlled to achieve task success, but is sufficient to keep them inside some safety domain: Passive Force Control.
2) The contact forces should be controlled because they contribute directly to the success of the task: Active Force Control [1, 7].

In the first case, contact forces are an undesirable effect of the task and it is generally sufficient to keep them inside some safety domain. They are not necessary for the task, so usually the strategy is adding flexibility to the end-effector with the objective of damping all the possible impacts and increase the tolerance to positioning errors, complemented with detailed and careful planning of flying trajectories and object approach. There are many solutions in the market to add flexibility to the end-effector, and in fact this is currently the standard approach in industry.

In the second case, the contact forces are necessary to finish the task correctly, i.e., controlling the contact forces making them assume some particular value or more generally to follow a predefined force profile is part of the task.

II. OBJECTIVES

The main objective of this paper is to report the development of an indirect force control strategy designed to operate with industrial robotic deburring applications. More specifically, the system reported here is developed to debur high-quality knives that incorporate innovative design from well-known authors (fashion designers). Therefore, these products are very difficult to manufacture and have quality requirements incompatible with human-based deburring, since humans introduce too many unacceptable deviations as a consequence of their incapacity to maintain concentration for long periods of time.

Consequently, to test and further develop the proposed strategy (section III) a set of experiments, clearly related with the planned final application, were designed and executed. The selected experiments are very simple:

1) The robot is commanded to follow a line path between two selected points.
2) An obstacle is placed somewhere in the path.
3) The robot should detect and contour the object applying a user selected contact force normal to the object surface. Contact should be always maintained under a contact force set by the user, until it reaches the original path again and keeps moving toward the original endpoint. The coordinates of the contact points should be kept in every experiment and they should correspond to the object contour.

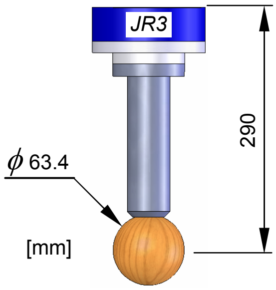


Fig. 1 Sensing tool attached to the JR3 sensor.

The experimental setup used to perform these experiments is composed of:

1) One industrial robot ABB IRB2400/S4C+/M2000.
2) One personal computer running Microsoft Windows Xp.
3) One JR3 force/torque sensor with accelerometers equipped with a PCI receiver and processing board installed on the PC PCI bus.
4) One especially designed sensing tool attached to the force/torque sensor and to the robot wrist.
5) A local area network (LAN), Ethernet and TCP/IP based, used for robot-PC communication. The network is isolated from the laboratory traffic using a properly programmed high-speed (100 Mbps) network switch.

III. FORCE CONTROL STRATEGY

A collision between the sensing tool and an object can be easily detected just by monitoring the actual force readings. The actual force components obtained from the sensor should correspond to the components of the object normal reaction force and are used to calculate the robot deviation trajectory $\left(\hat{d}\right)$. This is achieved by computing the cross product between the normal reaction force unitary vector $\left(\hat{n}\right)$ and the unitary vector $\left(\hat{p}\right)$ perpendicular to the trajectory plane.

$$\hat{d} = \hat{n} \times \hat{p} \qquad \textbf{(1)}$$

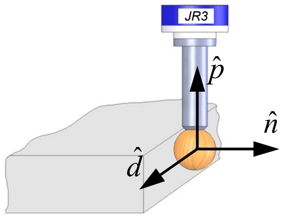


Fig. 2 Representation of the vectors $\hat{n}$, $\hat{p}$ and $\hat{d}$.

In theory, the vector obtained by this process should be tangent to the object contour, but due to the surface rugosity, friction forces, and noise, the so called normal force obtained from the sensor might deviate from the contact surface normal and consequentially the escape vector might not be tangent to the object contour surface. To overcome this problem, a correction process was implemented. This process consists in adding to the calculated deviation vector a component with the direction of the projection of the actual force on the trajectory plane, which will be called from now on, *correction vector* $\left(\vec{c}\right)$.

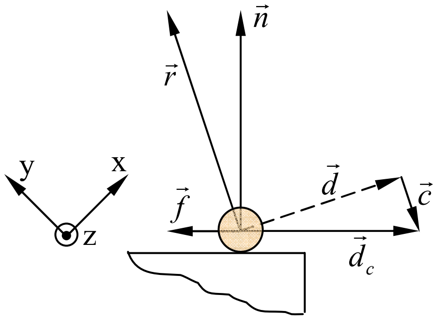


Fig. 3 Representation of considered vectors do define the deviation trajectory.

The vect ectory are (Fig. 3):
$\vec{n}$ - normal reaction from the object surface
$\vec{f}$ - friction forces
$\vec{r}$ - resultant force obtained from the sensor
$\vec{d}$ - deviation vector calculated from the force reading and the tool direction $\left(\hat{p}\right)$
$\vec{c}$ - correction vector, whose orientation is defined by projecting $\vec{r}$ on the trajectory plane, and the modulus is defined by the correction factor $\vec{c} = cf \cdot \hat{r}_{plane}$.

The idea is to add a component towards the object when the force reading is lower than the desired contact force, or add a component that points away from the object when the force reading is higher. The problem then consists on finding the correct intensity of the correction vector, in order to obtain a corrected deviation vector that is consistent with the contact settings. In the following, the intensity of the correction vector will be referred to as *correction factor* (*cf*).

The definition of the *correction factor* is based on three user selected parameters:

- **Desired contact force (**$F_{contact}$**)**: this value specifies the desired contact force during the contour motion. The software designed for the presented application forces the user to select a value in the 5.0 - 20.0 N range.

- **Force on collision detection (**$F_{collision}$**)**: this value specifies the force threshold used to identify collisions. This parameter is necessary in order to account for inertial forces detected by the sensor due to acceleration and deceleration of the robot, avoiding in this way false collision detections. The software designed for the presented application forces the user to select a value in the 0.25 - 1.0 N range.

- **Maximum correction factor (*Mcf*)**: this value specifies the maximum allowed correction factor.

The three above defined parameters are then used to compute the correction factor using the following linear function,

$$cf\left(\left|\vec{r}\right|\right)=\frac{Mcf}{F_{contact}-F_{collision}}\cdot\left|\vec{r}\right|+cf_{offset} \qquad \textbf{(2)}$$

where the initial value for cf_{offset} is given by:

$$cf_{offset}=\frac{-Mcf\cdot F_{contact}}{F_{contact}-F_{collision}} \qquad \textbf{(3)}$$

Fig. 3 represents graphically the *cf* function obtained just by using the following user defined parameters:

$F_{contact} = 10$ N, $F_{collision} = 0.3$ N, $Mcf = 0.15$

The obtained function can be adjusted by the user using the developed operation software.

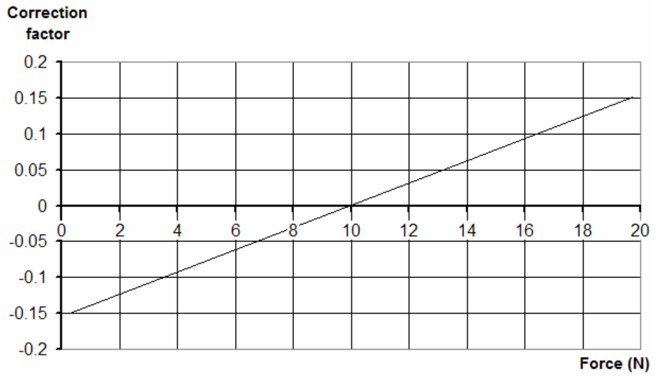


Fig. 4 Example of a user defined function to compute the *correction factor*.

In the case presented in Fig. 4, a contact force above 10 N would result in a positive *correction factor*, leading to a correcting vector component pointing away from the object. Similarly, for a contact force less than 10 N the *correction factor* would be negative resulting on a correcting vector component pointing to the object in order to increase contact force. It is also important to adjust the initial value of cf_{offset}. This can be done automatically just by comparing the actual force (from the sensor) and the user defined contact force. In fact, if the absolute value of the difference between the actual force reading and the user defined contact force exceeds a certain user pre-defined value from the contact detection force for a certain number of cycles, then a sample of the force signal is collected and the mean value is calculated. Subsequently, a new offset is defined by Equation 4 so that the mean value obtained matches the user established contact force.

$$cf_{offset}=\frac{Mcf}{F_{contact}-F_{collision}}\cdot\left(\left|\vec{r}\right|_{mean}-F_{contact}\right)+cf_{offset}^{'} \qquad \textbf{(4)}$$

where $cf_{offset}^{'}$ is the previous value of cf_{offset}.

The obtained correction factor is then used to compute the correction vector and to adjust the deviation vector ($\hat{d}$) guaranteeing that

$\left(\sqrt{1-cf^{2}}\cdot\hat{d}+cf\cdot\hat{r}_{plane}\right)$ is a unitary vector defining the direction that must be followed by the robot. Multiplying that vector by a user defined path step (*delta*), the actual position accommodation ($\vec{d}_c$), that should be commanded to the robot is obtained,

$$\vec{d}_c = delta\cdot\left(\sqrt{1-cf^{2}}\cdot\hat{d}+cf\cdot\hat{r}_{plane}\right) \qquad \textbf{(5)}$$

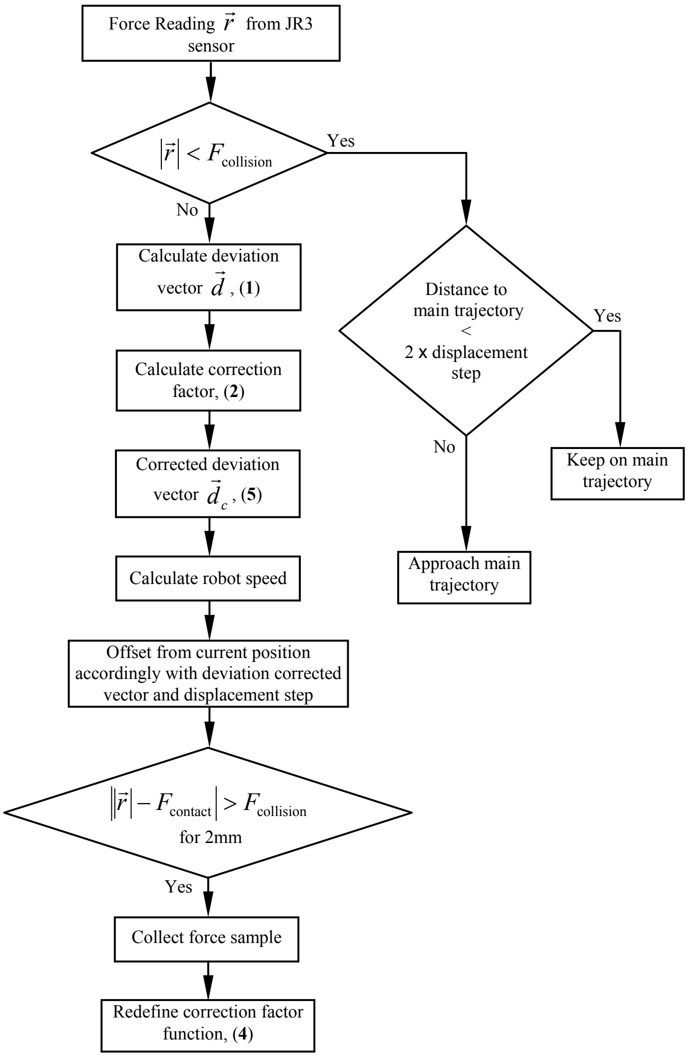
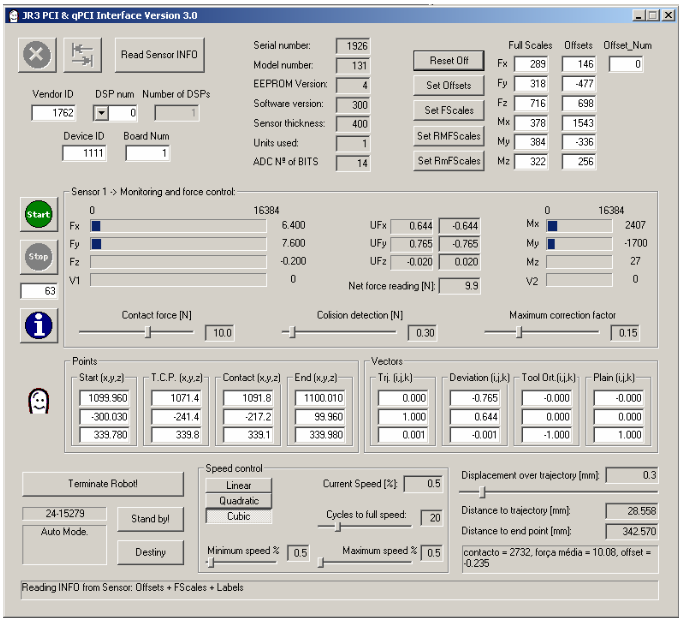


Fig. 5 Simplified flowchart of the force control used to define the robot motion.

IV. VALIDATION EXPERIMENTS

To execute the planned force control experiments a software application was developed to interface both with the sensor and the robot controller (Figure 6). The software is based on two software components: an ActiveX control named PCROB.NET2003 [8,9] and a C++ library (JR3PCI.lib) [11,12]. The application was developed using the Microsoft Visual Studio .NET 2003 programming suite, namely the C++ compiler. As already mentioned, the user interface allows the selection of the following parameters:

- **Contact force**[1]
- **Collision detection force**[1]
- **Maximum correction factor**[1]
- **Displacement step (delta):** path increment to be executed by the robot in each movement, ranging from 0.1 mm to 2.0 mm
- **Minimum speed:** robot minimum speed
- **Maximum speed:** robot maximum speed[2]
- **Linear, Quadratic, Cubic speed defining function:** in the developed software the robot speed was made a function of the correction factor. The objective is to have greater speeds whenever small or no corrections are necessary and lower speeds when using large correction factors. Selecting between one of the three presented choices defines the type of function that represents speed between the minimum and maximum speed.



Serial number: 1926
Model number: 131
EEPROM Version: 4
Software version: 300
Sensor thickness: 400
Units used: 1
ADC Nº of BITS 14

Vendor ID 1762
DSP num 0
Number of DSPs 1
Device ID 1111
Board Num 1

	Full Scales	Offsets	Offset_Num
Fx	289	146	0
Fy	318	-477	
Fz	716	698	
Mx	378	1543	
My	384	-336	
Mz	322	256	

Sensor 1 -> Monitoring and force control:

0 – 16384					0 – 16384	
Fx	6.400	UFx	0.644	-0.644	Mx	2407
Fy	7.600	UFy	0.765	-0.765	My	-1700
Fz	-0.200	UFz	-0.020	0.020	Mz	27
V1	0				V2	0

Net force reading [N]: 9.9

63

Contact force [N] 10.0
Colision detection [N] 0.30
Maximum correction factor 0.15

Points

Start (x,y,z)	T.C.P. (x,y,z)	Contact (x,y,z)	End (x,y,z)
1099.960	1071.4	1091.8	1100.010
-300.030	-241.4	-217.2	99.960
339.780	339.8	339.1	339.980

Vectors

Trj. (i,j,k)	Deviation (i,j,k)	Tool Ort.(i,j,k)	Plain (i,j,k)
0.000	-0.765	-0.000	-0.000
1.000	0.644	0.000	0.000
0.001	-0.001	-1.000	1.000

24-15279
Auto Mode.

Speed control
Current Speed [%]: 0.5
Cycles to full speed: 20
Minimum speed % 0.5
Maximum speed % 0.5

Displacement over trajectory [mm]: 0.3
Distance to trajectory [mm]: 28.558
Distance to end point [mm]: 342.570

```
contacto = 2732, força média = 10.08, offset =
-0.235
```

Reading INFO from Sensor: Offsets + FScales + Labels

Fig. 6 User interface

There are several parameters that might influence the software behavior when the robot is following a certain path. In order to test the software and the adopted force control strategy, a few experiments were executed to further evaluate the influence of the chosen parameters. The first parameter tested was the robot speed, and the second was displacement step value (delta). The procedure was:

1) The starting and end points were defined belonging to the same horizontal plane.
2) A square table was placed on the robot path, oriented approximately at an angle of 45º from the initial trajectory. This object was selected due to its rigidity and weight and also to test the behavior of the system in the presence of sharp edges.
3) A contact force of 10.0 N was defined; the collision detection force was set to 0.3 N and to the *maximum correction factor* was given a value of 0.15.
4) The robot was ordered to move following a linear path between the two previous defined points, using a path step (delta) of 0.1 mm.
5) Keeping the above mentioned parameters constant, three different constant speeds were tested: 0.1 mm/s, 1 mm/s and 2 mm/s.
6) For the speed exhibiting the best results, three different path step (delta) displacements were then tested: 0.3 mm, 0.5 mm and 0.8 mm.
7) The data for the tool center point (TCP) position, contact point (CP) position, tool speed, measured force, and all the user defined parameters is saved into a file during each experiment. Contact points are calculated during execution using the TCP positions read from the robot controller and the dimensions of the sensing tool.

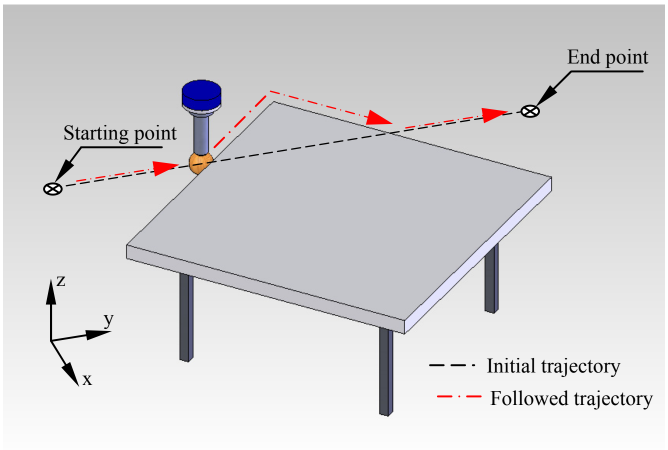


Fig. 7 Schematic representation of the conducted experiments.

V. RESULTS AND DISCUSSION

The obtained results are presented next using a graphical representation that shows:

1) The xy plane according with the robot global coordinate system.
2) The robot path using the TCP positions.
3) The robot path using the computed CP positions.
4) The obtained force profile during the experiment.

---

[1] Parameter already mentioned in section III.
[2] When the maximum speed is made equal to the minimum speed, constant speed is achieved.

Experiment 1:
$F_{contact}$ = 10 N, delta = 0.1 mm, speed = 0.1 mm/s

Fig. 8 represents the results for this experiment. Two force peaks are visible: when the tool collided with the table, and when the tool moved around the table corner. A good force tracking is observed, with actual force values maintained between 9 and 11 N. It is also interesting to observe that the calculated contact points form a precise representation of the object contour.

The table corner caused a dispersion of points around that region with large variations of the force value (Fig. 9). This dispersion results from the uncertainty on the resultant force orientation associated with the tool motion. The growing friction near the corner of the table is significant, and the obtained force reading deviates from the expected normal force, which causes a significant error on the contact point position.

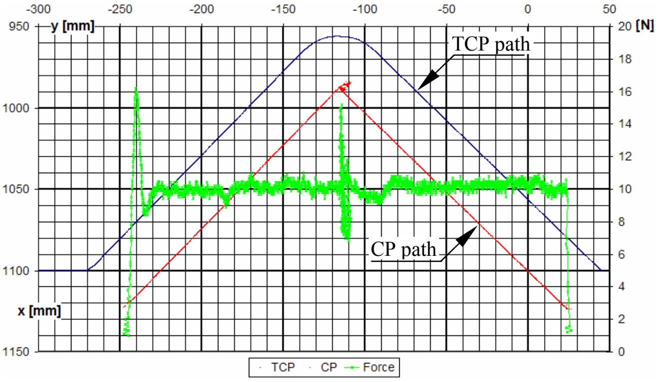


Fig. 8 Contour following using a 10 N desired contact force, 0.1 mm path step and 0.1 mm/s constant robot speed

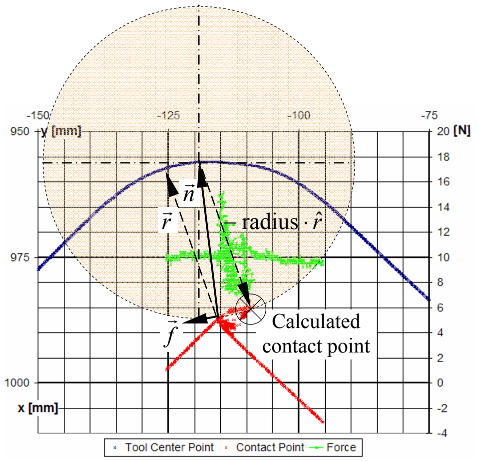


Fig. 9 Close up on contact points dispersion.

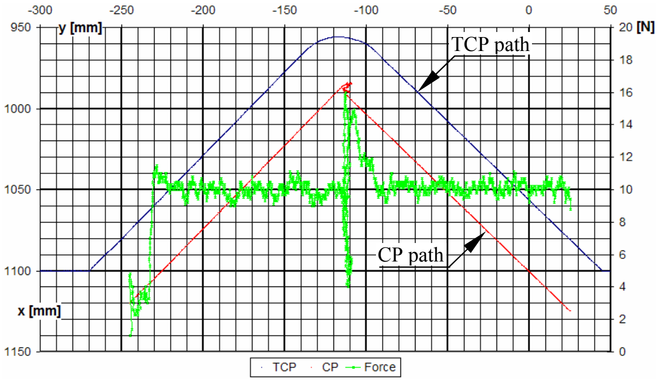


Fig. 10 Contour following using a 10 N desired contact force, 0.1 mm path step and 1 mm/s constant robot speed

Experiment 2:
$F_{contact}$ = 10 N, delta = 0.1 mm, speed = 1 mm/s

In the second experiment (Fig. 10) speed was increased to 1 mm/s. The resulting force profile and obtained object contour are quite satisfactory and not much different from the ones obtained in the previous experiment.

Experiment 3:
$F_{contact}$ = 10 N, delta = 0.1 mm, speed = 2 mm/s

The results for this experiment (Fig. 11) are similar to the ones obtained for the speed of 1mm/s, with the exception of a bigger force peak obtained when contact occurred.

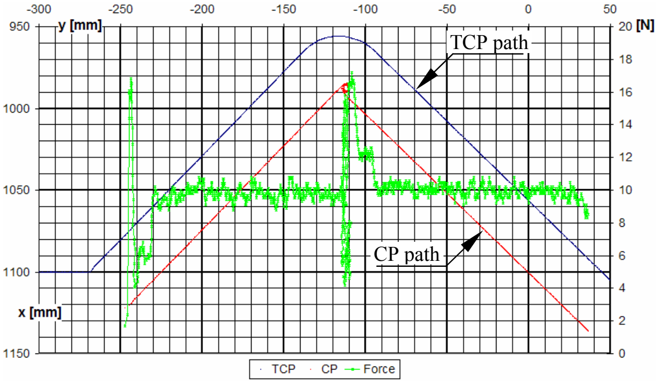


Fig. 11 Contour following using a 10 N desired contact force, 0.1 mm path step and 2 mm/s constant robot speed

Path step (delta) experiments:

In accordance with the previous results, a robot speed of 0.1 mm/s was selected for the path step (*delta*) tests. Fig. 12, 13 and 14 show the results obtained for steps equal to 0.3 mm, 0.5 mm and 0.8 mm, respectively. The results show that the desired force is obtained, although the amplitude of the oscillations changes with the increase of the step, being considerable for a step of 0.8 mm. Nevertheless, the contact force is maintained and in the first two cases the actual force values are kept between 9 and 11 N.

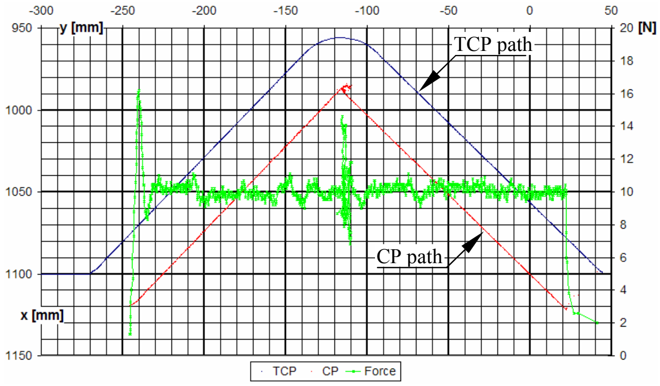


Fig. 12 Contour following using a 10 N desired contact force, 0.3 mm path step and 0.1 mm/s constant robot speed.

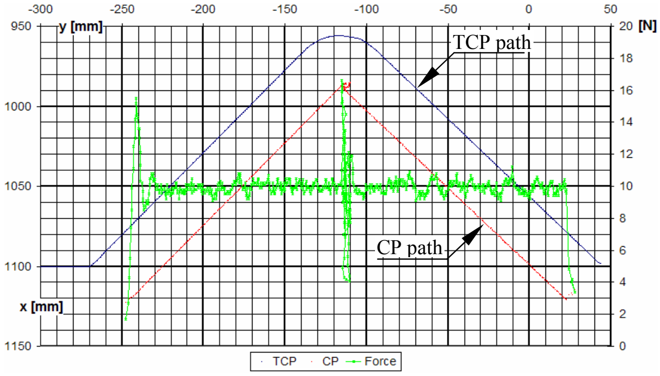


Fig. 13 Contour following using a 10 N desired contact force, 0.5 mm path step and 0.1 mm/s constant robot speed

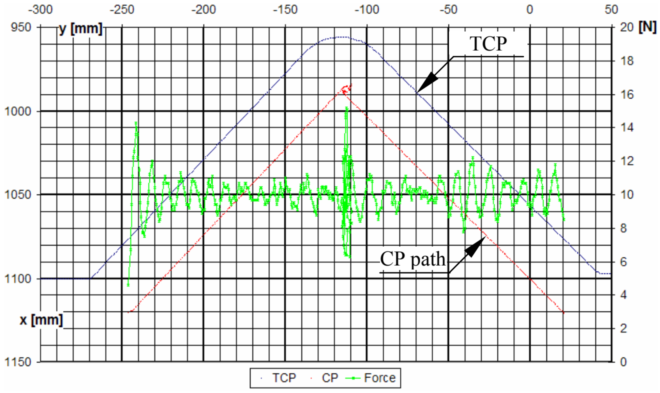


Fig. 14 Contour following using a 10 N desired contact force, 0.8 mm path step and 0.1 mm/s constant robot speed.

VI. CONCLUSION

This paper reports the development of an indirect force control strategy designed for industrial applications. With those approaches, the position of the robot is accommodated in accordance with the desired contact force and the observed actual force. Since a good model of the environment is difficult to obtain, namely on industrial applications, a simple strategy was designed to obtain the relevant parameters leading to an acceptable performance.

Experiments show a usable setup for contour following which is very useful to obtain the work-piece profile. A good selection of the path step seems to be, as expected, one of the most important variables to achieve good results: the smaller the increment over the trajectory the more regular is the resulting force profile. Low speeds also seem to lead to better results. The objective of the presented setup is to find the better compromise for a particular industrial application, achieving acceptable operational cycle times.

Sharp edges are geometric singularities difficult to overcome. According to the data collected so far, and with the observations made, possibly some mechanical improvements could be made to the sensing element in order to achieve better performance on sudden geometric variations. The diameter of the sensing element should be reduced and it should be mounted on a low friction bearing aligned with the trajectory plane's defining vector. The objective of these modifications would be to reduce friction forces and uncertainty on the direction of the force obtained from the sensor.

The strategy implemented to maintain contact with the object and keep contact force at a certain level seems to result over surfaces with a smooth and large radius continuity, although there are significant force variations on impact with objects (which is not important since impacts can be planned), especially at the higher speeds, and even more significant near object edges.

The desired contact force is also a parameter that should be tested. In the presented experiments, a contact force of 10 N was selected and oscillations of 1 N were observed around this value. In an industrial environment, more exposed to noise and vibrations, a higher contact force may be required. On the other hand the increase of the contact force also increases the flexion of the sensing tool what brings more uncertainty to the calculated contact point. Large force oscillations imply more uncertainty of the obtained work-piece contour. Like in any industrial process selected parameters are the ones that show acceptable results at higher execution speeds.

The obtained results are encouraging and the ability to perform contour recognition under a specified contact force can be very useful with the automatic deburring system being developed. In fact, this feature enables the system to acquire the exact contour of the working piece in the exact same conditions that will be used for the subsequent deburring task. This will contribute to minimize error and increase the process speed.

REFERENCES

[1] Siciliano B., Villani L., *Robot Force Control*, Kluwer Academic Publishers International Series in Engineering and Computer Science, Boston, MA, 1999

[2] De Schutter, J., and Van Brussel H., "Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks", The International Journal of Robotics Research, August de 1988.

[3] De Schutter, J. and Van Brussel, H., "Compliant Robot Motion II. A Control Approach Based on External Control Loops", The International Journal of Robotics Research, August, 1988.

[4] Nilsson K., "Industrial Robot Programming", Ph.D. Thesis, Department of Automatic Control, Lund Institute of Technology, May of 1996.

[5] Siciliano B., "Parallel Force/Position Control of Robot Manipulators", Proceedings of the 7th International Symposium of Robotics Research, Springer-Verlag, London, UK, 1996:79-89.
[6] Chiaverini, S., "Force/Position Regulation of Compliant Robot Manipulators", IEEE Transactions on Automatic Control, Março de 1994.
[7] Pires, JN "Force control in industrial robotics", PhD Thesis, Mechanical Engineering Department, University of Coimbra, June, 1999.
[8] Pires, JN, Sá da Costa, JMG, "Object-Oriented and Distributed Approach for Programming Robotic Manufacturing Cells", IFAC Journal Robotics and Computer Integrated Manufacturing, Volume 16, Number 1, pp. 29-42, March 2000.
[9] Pires, JN, "Semi-autonomous manufacturing systems: the role of the HMI software and of the manufacturing tracking software", Elsevier and IFAC Journal Mechatronics, Vol.15, N.10, 2005.
[10] Nasri, H, "Modeling and Requirements of the Automated Deburring Process", PhD Thesis, Division of Robotics, Department of Mechanical Engineering, Lund University, Sweden, May, 1999.
[11] JR3 Force/Torque Sensor Users Manual, JR3 Inc. Woodland, California, 2001.
[12] JR3 PCI Web Site, http://robotics.dem.uc.pt/norberto/jr3pci/, 2001.

Acknowledgment

This work has been partially funded by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot.