Flexible 3D Trajectory Teaching and Following for Various Robotic Applications

Theodor Borangiu*, Alexandru Dumitrache*, Anamaria Dogar*

**Centre for Research and Training in Industrial Control, Robotics and Materials Engineering, University Politehnica of Bucharest, Romania (Tel: +40/21 402 93 14; e-mail: {borangiu, alex, dogar} @cimr.pub.ro)*

Abstract: This paper addresses an actual problem regarding complex industrial robot applications. Based on the fact that no CAD model for the processed parts is available, the application presented here consists in a 3D accurate path following, applicable in various robot tasks. Using a sensor-based 3D path learning procedure available in automatic or manual mode, the 6 d.o.f industrial robot will be able to reproduce in real-time the learned trajectory. Calibration and synchronization aspects are presented, and experimental results are provided and analyzed.

Keywords: Robotic manipulators, trajectory learning, laser range finder, robot vision.

1. INTRODUCTION

This article presents an approach for solving the 3D path following problem, illustrated in Fig. 1, where the CAD model of the part is not known. The robot arm has to move a tool tip along a 3D trajectory on a given workpiece, for performing various technological operations. The second goal is to present the possibility of using a laser-based profile scanner as a 3D vision sensor for precise robot guidance, since the accuracy of these devices is usually very good, reaching tens of microns or even micrometers. The laser sensor used for validating the methods presented here is capable of achieving a 30 μm standard deviation per point, which is comparable to the repeatability of the robot arm.

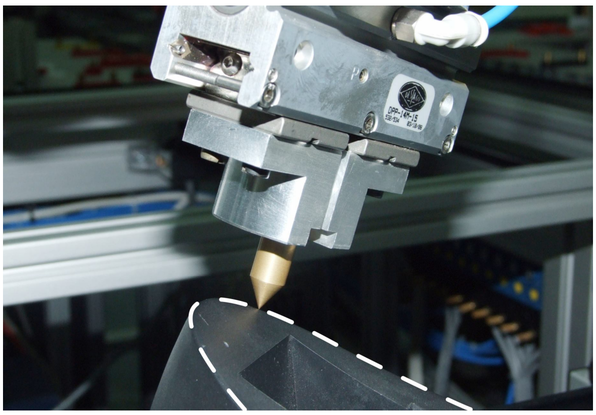

Fig. 1. Sample 3D path following problem

A typical solution for this problem is to extract the 3D path from the CAD model of the workpiece (Naveh, 2008; Norberto et al., 2004; Sallinen et.al, 2003; Weihua et.al, 2000). This approach, being the most used in industry, has a few drawbacks, which will be addressed in this paper. One of them comes from the lack of accuracy in the kinematics module of the current robotic arms when they work without additional information from sensors. This results in a mismatch between the real and ideal 3D descriptions of the path. Furthermore, the path following problem requires that the robot has to know the exact 3D position and orientation of each part. In other words, using the trajectory directly from CAD requires very good absolute accuracy of the robot arms, and also for the part fixtures and other mechanical elements.

The lack of accuracy in robot kinematics may have different sources, such as machining tolerances in manufacturing the robot joints, errors in the electromechanical calibration, elasticity of the links (Lange et.al., 2008), the payload of the robot, or link dilatation due to heating of the robot actuators or neighbourhood equipment (Poonyapak et.al, 2008). Even unexpected operations, such as hitting the emergency stop while the robot is moving, may have an impact on the absolute accuracy of the robot.

The robot manufacturers often specify the repeatability of the robot arm, which only measures the ability of the robot to return to a previously taught location, within the specified range. The robot arm used here is able to achieve a repeatability of 20 μm at constant temperature; however, the absolute accuracy is much worse (0.5 mm estimated).

There are also situations when there is no 3D CAD model available, for example, when processing free-form objects like clay or wax models; a system for painting unknown parts without any CAD data is presented by (Pichler, 2002).

2. PROPOSED SOLUTION

The proposed system integrates a two vision technologies. A short range, triangulation-based laser profile scanner, which measures 2D profiles with good accuracy, mounted on the robot wrist, provides 3D vision capabilities. The main

* This work is funded by the National Council for Scientific University Research, in the framework of the National Plan for Research, Development and Innovation, grant 69/2007.

advantages of this sensor are the high accuracy and also the good robustness in changing lighting conditions, since the laser light is significantly more intense than ambient light. Moreover, since the laser is usually monochromatic, the tolerance for ambient light changes is increased using optical filters. However, the disadvantages are the slow speed, since the robot arm has to move the sensor in order to scan a 3D area, the amount of sensor data for processing can be very high, and also the difficulties with highly reflective or semitransparent materials. The eye safety hazard for the operator should be also considered.

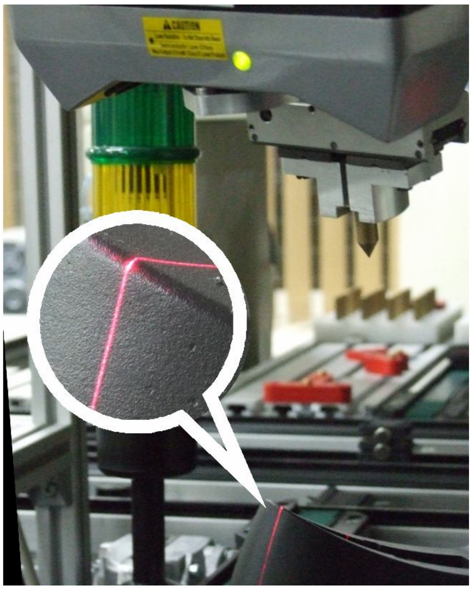

Fig. 2. The laser sensor looking at the edge of a black object.

For industrial usage, it is required to make sure the robot is able to follow the 3D path for many similar parts which come on a conveyor belt, in different positions and orientations. If the parts are rotated around Z with arbitrary angles, and also have small rotations (e.g. less than 5 degrees) around X and Y, it is possible to use the 2D camera to perform a quick localization, which detects the XY position and rotation around Z with approximation, and afterwards, the 3D sensor, which is accurate but slow, will detect the exact 3D position and orientation of the part by scanning a few key areas and matching them to the reference data recorded for the sample part used for teaching. The matching procedure is performed using an efficient variation of Iterative Closest Point algorithm (Rusinkiewicz and Levoy, 2001) applied on point cloud data from the laser sensor. In this way, the hybrid vision solution compensates from the slow speed of the laser sensor, but exploits its accuracy in precise 3D localization.

3. CALIBRATION ISSUES

In this scenario, the robot learns a complex 3D path using the laser sensor, and then follows them using a physical tool. This can be achieved using two tool transformations (Fig. 3):

- T_L from the robot wrist to the reference frame of the laser sensor;
- T_T from the robot wrist to the tip of the tool.

The two transformations are determined using a calibration procedure.

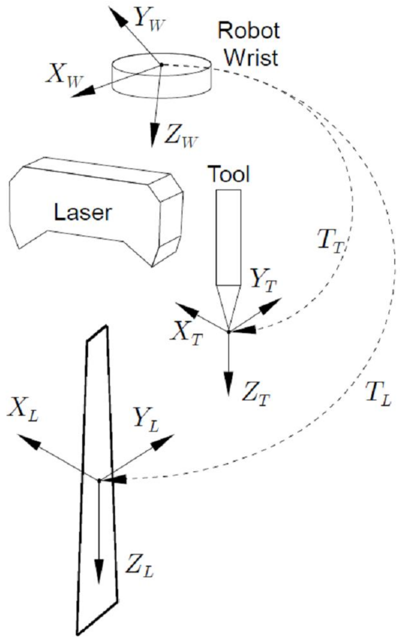

Fig. 3. The two tool transformations: from wrist to laser plane and from wrist to tool tip. All the elements are coupled using a mechanical fixture, not represented here.

The T_L transformation describes the position and orientation of the laser plane, with respect to the robot wrist. The data from the laser is a 2D set of points mapped to the laser plane. This data is extended into 3D on the YZ plane considering the X coordinate equal to 0. The coordinates from the sensor are transformed into the reference frame of the robot base by applying the T_L matrix, combined with the robot direct kinematics transformation, which describes the instantaneous location of the robot at the time of measurement.

Determining this transformation allows the robot to reach any location identified by the sensor, by changing the tool transformation. The measurements are performed using the T_L transformation, and the motions are generated switching to the T_T tool transformation.

The calibration process starts with aligning the laser reference frame $X_L Y_L Z_L$ to the wrist reference frame $X_W Y_W Z_W$. This procedure compensates for the misalignments of the fixture between the laser sensor and the robot wrist. A basic operation in the calibration process is the *ball matching*, where the laser sensor is swept along X_L over a tooling ball. The point cloud obtained is segmented, and a sphere is fitted to the data in order to determine its center (Fig. 4).

The sphere is fitted using the Riemann method, by projecting the data points on a 4D paraboloid, the result being a hyperplane. A similar method was presented in (Frühwirth et.al, 2003) for fitting a circle, and it was straightforward to extend the method for fitting the sphere. This method is advantageous because the hyperplane fitting problem is linear, and can be solved by well-known robust fitting methods based on weighted least squares (Fox, 2002). The robust fitting method is iterative and slower than the classical least squares fitting technique, but the results (center and radius) are not affected by outliers in data.

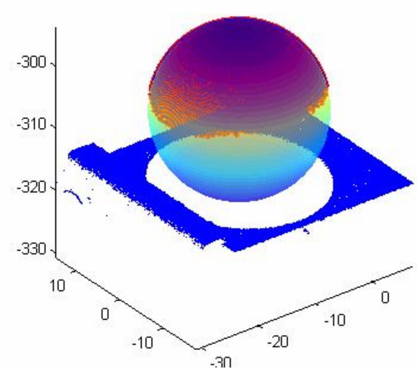

Fig. 4. The ball matching process used for calibration.

By placing the tooling ball in different position in the laser field of view, the orientation of the sensor plane can be determined with basic 3D geometry. At least 3 positions are required for determining the plane equation.

If the sphere is scanned while the sensor plane is not perpendicular to the sweeping direction, which is normal before performing the calibration, the 3D data will not represent a perfect sphere. The result will be a "stretched" sphere, which will result in large errors in the fitting procedure, and its center will not be determined exactly. For this reason, if the angle between the vectors X_L and X_T is too large, the calibration procedure has to be repeated until the orientation of the plane (yaw, pitch and roll) converges to the correct values.

After the orientation has been determined, the center point is computed by attempting to keep the tooling ball with its center on the origin. The robot will detect the ball with different orientations; from each orientation, the origin of the sensor field of view will be in the same point (center of the tooling ball). The translation component of T_L is determined by solving a linear system, either exactly or using least squares, as described in (Hallenberg, 2007).

The tool transformation for the physical tip can be computed using a similar method, either by manually positioning the tip to a fixed point, in different orientations (at least 3), or by using a calibration method specific to the particular tool type.

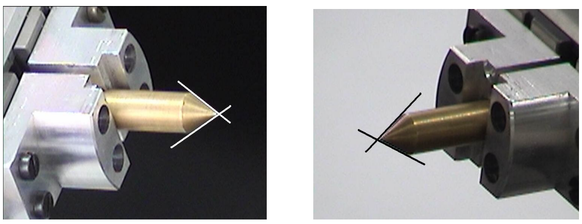

Fig. 5. Tool tip calibration using two fixed cameras.

The tool tip used in this experiemnt was calibrated using two cameras with PTZ (pan, tilt and zoom) capabilities (Fig. 5). The 2D vision system detects the tip of the tool by computing the intersection point of two edges, and the calibration software adjusts the robot so that the tool tip remains in the center of both images, while the tool is changing the orientation. This method detects the tool offset (*dz* for tool length; *dx* and *dy* for the gripper eccentricity) by solving a linear system similar to the one used for the laser sensor.

Using one fixed camera will detect the position changes in only two directions. It could be possible to estimate the third coordinate (depth) by detecting the scale factor of the tool, but this is not accurate. Two cameras offer 4 position variables, which is redundant, but they offer information about the position on all 3 degrees of freedom adjusted.

The calibration begins by positioning manually the cameras, so the tool tip is visible in both images, like in Fig. 5. Then, the system will adjust the robot so that the tool tips are close to the center of image (exact match is not possible since the cameras are positioned with very low accuracy). The current robot location, and the position of the tool tips, are saved as reference positions. The next two locations are determined by changing the orientation from the reference robot location with a small amount (e.g. 5 degrees) in two different directions (e.g. *X* and *Y*). The amount of rotation should be small enough so that the tool tip remains in the image. The robot will translate the tool tip towards the reference positions on the two images, and the final locations are saved.

The small change in orientation allows the computation of a low-precision tool transformation, with an error of the order of magnitude of 1 mm. This will allow the robot to execute rotations around the tool tip (with approximation), and now the robot is able to change the orientation with high amounts (e.g. 45 degrees) while keeping the tool tip visible in both images from the cameras. The calibration process is repeated for various orientations: at every step, the robot changes the tool orientation, and then performs a translation until it reaches the reference position in both images. The calibration software records a set of *N* robot locations, all having the tool tip located at the same point in space, but with different orientation. The new set of robot locations can be rewritten as a linear system that can be solved using least squares, and the solution determines the final value for T_T.

The reader will observe that there is not need for any calibration for the fixed cameras; the software doesn't have to know anything about their location with respect to the robot. The only requirement is that *Z* axes of the two cameras shoud not be parallel; otherwise, the second camera will provide the same information as the first one. It is reccommended that the angle between the two cameras should be close to 90 degrees if possible, in order to maximize the pixel-to-mm ratio, which is the main factor which determines the accuracy of the method. Angle values between 45 and 135 degrees will also give good results.

In this method, the tool orientation is assumed to be correct, since it is not a critical parameter for the 3D path following problem presented here. In the implementation, it was assumed that the tool had the same orientation as the laser sensor; however, this does not limit the generality of the method, and the sensor can have any orientation, different from the one used for the tool.

4. TRAJECTORY LEARNING

This section describes the process of learning a 3D path which follows a well defined feature on the workpiece. The feature can be a sharp edge, a filleted edge, or any shape that can be recognized and located in the 2D profile data from the laser sensor. In the example presented in Fig. 6, the user has taught the 2D vision engine to locate the rounded edge of the workpiece, which has to be followed by the robot.

The learning process has two stages:

a) Learning the coarse, low resolution trajectory (manually or automatically);

b) Refining the accuracy by computing a fine, high resolution trajectory (automatically).

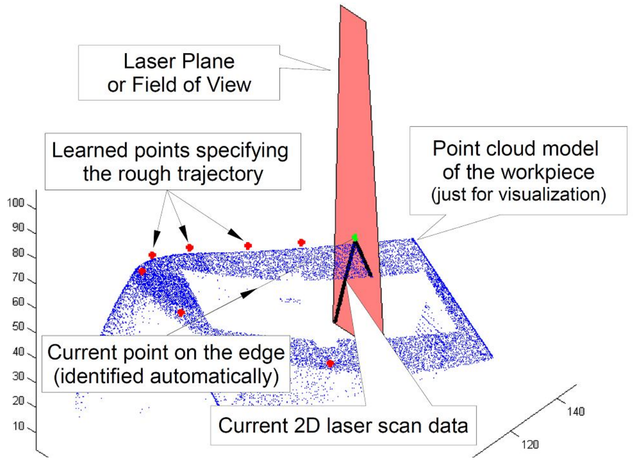

Fig. 6. Interactive trajectory learning. The user is positioning the laser sensor, until the edge to be followed appears in the field of view, and is located automatically.

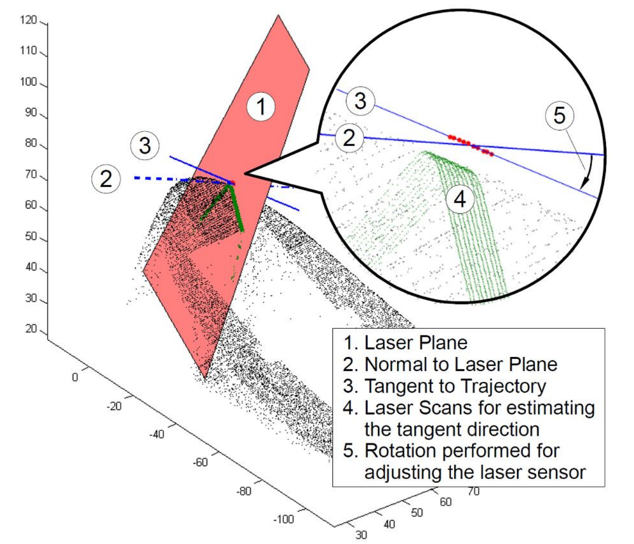

Fig. 7. Automatic orientation of the sensor. The robot moves the sensor on a small linear segment in order to estimate the tangent vector at the desired 3D path, and then rotates the sensor plane, making it perpendicular to this vector.

Coarse Learning - Manual mode

The first step can be either interactive or automatic. In the interactive mode, the user has complete control of the learning process. The user positions manually the sensor by jogging the robot, so that the desired edge becomes visible. The system helps the user by automatically detecting the edge position, and it can also adjust the orientation.

For the orientation, the system will choose, by default, the current orientation of the sensor, or the user may specify a relative rotation. The system also offers a semi-automatic option for aligning the sensor in such a way that the laser plane is perpendicular to the trajectory, by computing the rotations around Y_L and Z_L axes. The tangent to the trajectory is detected using a small scan pass (Fig. 7), by fitting a straight line to the positions of the feature located by the 2D vision engine.

The rotation around X_L is determined by the 2D vision engine when locating the tracked feature.

Coarse Learning - Automatic mode

The second method for learning a coarse 3D trajectory is automatic. The user positions the sensor at the starting point, learns the 2D edge model and specifies the search direction (which can be the positive or negative direction of X_L) and the spacing between location. A higher spacing will allow the learning process exeute faster, but in this case it may have difficulties on regions with high curvature. For the example presented, a good choice for spacing in automatic mode is 10 mm, while in manual mode, the distance between the points taught can be higher, e.g. 20 or 30 mm.

The automatic mode will stop automatically when the edge is not recognized any more. If there is any ambiguity, i.e. two or more similar edges are detected in the same snapshot, the system either uses a heuristic to choose the most plausible solution, or asks the user which one should choose.

The main advantage of the automatic mode is that it can run with very little user interaction (only at startup and in case of special situations), while the manual mode provides more flexibility and is advantageous when the task is more difficult and the user wants to have full control over the learning procedure.

Fine Trajectory Learning - Automatic mode

In this step, the coarse trajectory is followed using a continuous robot motion. During the motion, the tracked edge is always visible in the sensor field of view; however, the tracking error may be large. The sensor performs measurements while the robot is moving, and the data is collected on the PC, the sensor readings are matched to the instantaneous robot positions and sent to the 2D vision engine for edge detection and precise localization. The result of this step is a fine spaced sequence of robot locations, which represent an accurate description of the 3D path to be followed by the tool tip.

5. SYNCHRONIZATION ISSUES

The data from the vision (laser) sensor has to be matched with the instantaneous position of the robot in order to obtain consistent measurements expressed in the robot reference frame. There are three main approaches for this:

a) Stop and look. Using this method, the measurements from the sensor are read only when the robot is not moving, and has reached its programmed destination. The method is the easiest to implement, does not need any synchronization signals between the vision sensor and the robot, but it is also the slowest, being limited at around 1 or 2 sensor readings per second. It is used currently in the coarse learning stage for the 3D trajectory, in automatic mode.

b) Buffered synchronization. This method requires a trigger signal, usually sent from the vision sensor in the middle of the exposure period, to the robot. When receiving the signal, the robot latches its instantaneous position, and stores it in a buffer. The trigger signal may be reversed, so the robot activates the vision sensor. The data from the robot and the sensor is collected on the PC and processed at a later time. This method allows sensor readings to be taken while the robot is still in motion, and close-spaced measurements can be taken at much higher rates, e.g. 50 readings/second. However, there may be a significant delay from the of data acquisition until the data is processed by the PC. In the system used here, the bottleneck is the Ethernet link between the robot and the PC, and the delay is usually 0.2 – 0.3 seconds, and could reach 1 second. The buffers ensure that the data is matched properly even when high delays occur in communication.

This strategy is used in the fine learning stage for the 3D trajectory, which is automatic. If the delay is assumed to be less than a certain value, this method can be also used for the coarse learning stage, limiting the robot speed and using a predictor module. Should the delay exceed the assumed value, due to a transient perturbation in communication, the robot can pause the learning process and resume automatically, so the delays will not affect the result.

The fine learning stage can be also implemented on a system which does not support this method of synchronization, using the *stop and look* approach; the only disadvantage will be a significant reduction in speed.

c) Tight loop synchronization. This method is suitable for closed loop robot guidance using the vision sensor, and it involves small and predictable response times from both the vision engine and the control loop. Using this method will allow substantial speed up of the coarse learning stage; however, both the hardware and software requirements will be much higher. This method will also allow real-time path identification and precise tracking in only one pass.

6. EXPERIMENTAL RESULTS

The method for path teaching and following was validated using the same sensor to record the measurements while the robot was following the recorded path. Since in this application, the position of tool tip point is very important to be on the trajectory, but small changes in orientation are tolerable, the measure for the tracking error was chosen to be the distance between the ideal (recorded) path, and the actual tool center point. For the evaluation of the error, the deviation was measured with the same sensor, being used in passive mode, i.e. not having any influence on the path. The laser sensor was used in buffered synchronization mode.

The first experiment shows the tracking errors achieved for a coarse trajectory, which was learned automatically, being defined by points spaced at 10 mm. Fig. 8a) shows that the coarse trajectory is reasonably good for the low curvature regions, while exhibiting higher errors for the middle portion with high curvature. In this graph, the robot was stopping to each taught location, and without waiting for the position errors to be nulled, accelerating towards the next location.

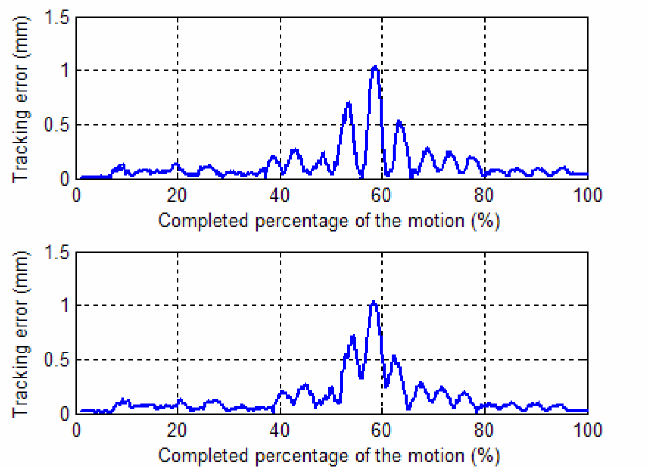

Fig. 8. Tracking error for the coarse path, with two possible interpolation methods: a) Point-to-point motion using linear segments; b) Smooth continuous motion.

Fig. 8b) represents the tracking error during a continuous path interpolation, which is obtained using a proprietary algorithm that does not actually reach the planned locations, but blends the motion segments achieving a behavior similar to B-Spline curves. The tracking error has a similar shape, with the difference that the error does not drop close to zero at every location taught, and this is normal because the robot does not actually reach these locations.

From Fig. 8 it is possible to see that the tracking error is low enough in order to allow the system to automatically compute the fine 3D path, at high resolution. The refinement process is much more robust, and it may accept a coarse tracking error about 10 times higher, the main condition being that the tracked edge remains visible in the field of view of the sensor throughout the coarse trajectory tracking.

Fig. 9 display the tracking error for the high-resolution path, followed with constant speed. The first test was executed with 10 mm/s at the tool tip, which was the same speed used for coarse trajectory tracking, and the second one was slowed down at 1 mm/s. In both cases, the tracking error is much smaller than in the coarse tracking case, the error signal is more similar to a white noise, and its main sources are the vibrations during the motion, the tracking errors from the robot servo loop, and also the estimation errors from the 2D vision engine which was used for processing the data from the laser sensor in order to locate the tracked edge.

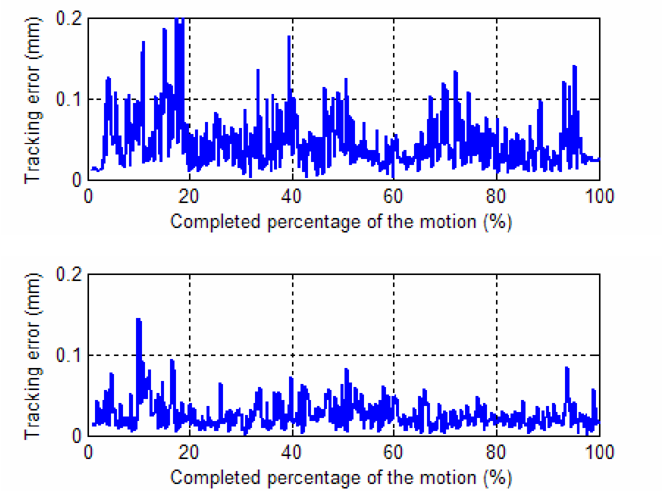

Fig. 9. Tracking error for the fine path, at different speeds for the tool tip: a) 10 mm/s; b) 1 mm/s.

CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper presented a flexible and accurate method for teaching a 3D path along a visual feature of an existing object, using a laser sensor for robot guidance. The method was validated successfully, and there are many industrial tasks which can employ it, such as sealant dispensing, painting wireframe workpieces, welding and edge deburring.

The laser sensor can be used only for teaching the path, and removed from the robot during the technological process. This is desirable when the process is likely to damage the sensor, for example, in welding applications. Since the sensor can be attached easily on the robot, on top of the existing end effector, and the calibration routine is quick and only requires a tooling ball visible its workspace, the same sensor can be used for teaching paths for many robots.

The method may also be used for assesing the repeatability of industrial robots while following a continuous path. An important conclusion that can be drawn from this experiment is that a robot arm will not be able to achieve the repeatability indicated by the manufacturer in continouous motion, while it achieves it when it has to stop at some prescribed location.

The calibration method used for the experiment provides also a framework for performing 3D vision guidance tasks using the robot, such as detecting accurate position and 3D orientation of parts for grasping and manipulation. The coordinates returned by the laser sensor can be expressed in the reference frame of the robot base, allowing it to build a 3D description of its environment. A collision engine can use this information to prevent crashes between the robot and its surrounding equipments, and a path planning module can be used in order to generate autonomous motions for performing various tasks in unstructured environments.

There are some tasks which require a precise orientation of the tool; for other tasks, the exact orientation is not critical. For the second category, a post-processing module can exploit the freedom of changing slightly the orientation of the tool from the taught values, along the path, in order to optimize certain criteria, for example, ensuring motion smoothness, or minimizing the speed of wrist joints while keeping the tool tip speed constant, thus reducing vibrations. Some tasks may allow arbitrary rotation of the tool around its *Z* axis; this becomes an extra degree of freedom which can also be used for singularity avoidance using a planning algorithm.

Another goal of the project is improving the accuracy of the method. It is expected that smoothing the orientation component will reduce vibrations and also the tracking error, and it will also allow higher tracking speeds.

REFERENCES

Fox, J. (2002). *Robust Regression.* in *An R and S-PLUS Companion to Applied Regression,* Sage Publications.

Frühwirth R, Strandlie A et.al. (2003). *A review of fast circle and helix fitting.* Nuclear Instruments and Methods in Physics Research, vol. A 502, pp. 705-707

Hallenberg, J. (2007). *Robot Tool Center Point Calibration using Computer Vision.* Master Thesis, Electrical Engineering Dept., Linköpings University, Sweden

Lange, F., Hirzinger, G. (1999). *Learning accurate path control of industrial robots with joint elasticity.* In Proceedings of IEEE International Conference on Robotics and Automation, pp. 2084-2089.

Naveh N. (2008). *How to Make a Robot Path in 10 Seconds.* Machine Design. Ed. Leslie Gordon.

Norberto P. J., Godinho T., Ferreira P. (2004). *CAD Interface for automatic robot welding programming.* Industrial Robot: An International Journal, no. 1, vol. 31, pp. 71-76

Pichler, A., Vincze, M., et. al. (2002). *A Method for Automatic Spray Painting of Unknown Parts.* Proceedings of IEEE International Conference on Robotics and Automation, 2002. ICRA '02.

Poonyapak P, Moyra J. M., Hayes J. (2008). *Experimental Error in Measuring Temperature-Induced Deformation in a Single-Link System*, Proceedings of RAAD 2008.

Rusinkiewicz, S., Levoy, M. (2001). *Efficient Variants of the ICP Algorithm.* Third International Conference on 3D Digital Imaging and Modeling.

Sallinen M. (2003). *Modelling and estimation of spatial relationships in sensor-based robot workcells.* PhD Thesis, University of Oulu, published by VTT.

Weihua S., Ning X., et. al. (2000). *Automated CAD-guided robot path planning for spray painting of compound surfaces.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol. 3, pp. 1918-1923.