Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia CIRP 3 (2012) 191 – 196

www.elsevier.com/locate/procedia

45th CIRP Conference on Manufacturing Systems 2012

Robot Path and End-Effector Orientation Planning Using Augmented Reality

H.C. Fang[a], S.K. Ong[a,]*, A.Y.C. Nee[a]

[a]*Mechanical Engineering Department, Faculty of Engineering, National University of Singapore*
Engineering Drive 1, Singapore 117576, Singapore

* Corresponding author. Tel: +65-6516-2222; fax: +65-6779-1459; *E-mail address*: mpeongsk@nus.edu.sg

Abstract

This paper presents an Augmented Reality-based approach for planning the path and the orientation of the end-effector for an industrial robot. The targeted applications are those where the end-effector is constrained to follow a visible path, the position and model of which are unknown, at suitable inclination angles with respect to the path. The proposed approach enables the users to create a list of control points interactively on a parameterized curve model, define the orientation of the end-effector associated with each control point, and generate a ruled surface representing the path to be planned.

Keywords: Robot planning; Orientation planning; Augmented Reality

1. Introduction

With the prevalence of robots supporting humans in key activity areas, there has been a progressive increase in the demand for robotic systems that are safer, easier to install and reprogram, etc., with intuitive Human-Robot Interaction (HRI) interfaces for the SMEs and users who may not be experienced in the field of robotics [1-2]. Breazeal *et al*. [2] identified the overlapping space that can be perceived by both the human user and the robot system as a key requirement for effective HRI. Recent research on HRI suggests that augmented reality (AR) interfaces can enrich the interaction process in robot manipulation [3-6]. In addition, the users are able to interact with the spatial environment in path planning and robot end-effector (EE) orientation planning [7-10]. Interaction with a virtual robot model instead of a real robot increases the safety of the operator when he/she is present within the operating range of the robot.

This paper presents an AR approach, namely, Robot Programming using Augmented Reality (RPAR-II), for robot path planning, robot EE orientation planning, and path optimization incorporating robot dynamics. The RPAR-II approach assists the users in the evaluation of the trajectory and provides cues to tune the controller parameters. The paper is organized as follows. Section 2 presents the related studies. Section 3 gives an overview of the system. Section 4 presents the proposed approach for interactive robot path and EE orientation planning. Section 5 presents the case studies to evaluate the system developed. Section 6 summarizes the paper.

2. Literature survey

Numerous research studies have been reported on HRI interfaces as more enabling technologies become available. One example is virtual reality (VR), which is aimed at increasing the intuitiveness for the operators in an immersive environment [11-12]. A major advantage of using VR is the provision of various evaluation options due to its scalable modeling capability. However, majority of the VR systems require extensive modeling of the environment and additional effort to maintain/update the mapping between the virtual environment and its real counterparts. Another example is robot programming by demonstration (PbD), which involves a user performing a task manually, leaving the

doi:10.1016/j.procir.2012.07.034

robot to observe, follow and learn the demonstrations in real-time. This approach, however, requires the robotic systems to be integrated with suitable sensors and processing techniques for knowledge acquisition, human gesture recognition, etc. [13].

AR has been applied in various robotics applications. AR can assist the users in pre-operative planning as well as real-time tasks implementation. Marin *et al*. [3] presented a mixed reality-based visualization interface where virtual cues can be displayed to the users for overlapped objects isolation and grasping operations conducted at a remote site. An AR-based cueing method was reported [4] to assist the users in robot navigation in a tele-operation task under display-control misalignment conditions. These studies show positive effects of using AR on operator performance in *ad hoc* tele-robotic tasks.

AR offers the possibility to visualize and interact with the robot paths augmented onto the real environment. Present research on robot task planning using AR mainly focuses on geometric path planning considering only robot kinematics. The AR-based simulations presented in these systems are conducted to verify the accessibility of the planned paths. Chong *et al*. [7] presented a method to plan a collision-free path through guiding a virtual robot using a probe attached with a planar marker. Zaeh and Vogl [8] introduced a laser-projection-based approach where the operators can manually edit the paths projected over a real workpiece using an interactive stylus. Reinhart *et al*. [9] adopted a similar approach [8] in robotic remote laser welding tasks. Several approaches have been reported on the planning of the EE orientation along a path using AR. Ong *et al*. [10] presented an approach which enables the user to modify the orientations to avoid possible collisions. Reinhart *et al*. [9] developed a computer-based approach for planning a smooth path along a weld seam by optimizing the EE orientations.

The generation of an optimized trajectory for a given robotic task considering motion constraints based on the de-coupled approach [14] has been well studied and reported. Convex optimization techniques, such as interior point methods [15-16], have been applied to solve de-coupled trajectory planning. These methods solve the optimization issue by reducing it to a sequence of linear equality constrained problems. In this research, a similar approach is implemented to determine the time-scale trajectory taking into account joint velocity and joint torque constraints.

3. System overview

The setup of the RPAR-II system is shown in Fig 1. It includes a Scorbot-ER VII robot arm mounted with an EE, a computer, a monitor-based display, a camera, and an interaction device attached with a marker-cube.

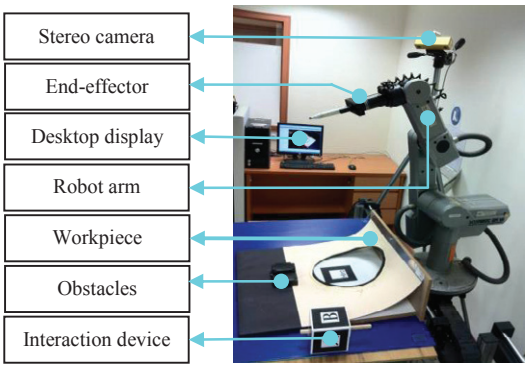

Fig 1. System setup

The AR environment consists of the physical entities in the robot operation space and a virtual robot model. The target applications for the proposed system are pick-and-place operations and path following operations. The main modules of the system are as follows.

- Interaction Module: An ARToolKit-based tracking method is used for virtual robot registration and interaction device tracking. The interaction device is used to guide the virtual robot in task planning. The robot has five degree-of-freedom (DOF). Therefore, a pose tracked using this interaction device should be mapped to a valid robot configuration through inverse kinematics techniques.
- Path and EE Orientation Planning Module: Generation of a suitable solution space and a smooth path within the solution space, and the determination of the EE orientations along the path.
- Path Optimization Module: Techniques to transform a geometric path into an optimized trajectory subject to robot dynamics.
- Simulation Module: Interactive simulation of the planned motion, allowing the user to evaluate visually the quality of the trajectory prior to the translation of the trajectories into robot programs..

4. Methodology

This section presents a methodology for robot path and EE orientation planning using AR. The role of the users in HRI are (1) generating the Collision-Free Volume (CFV); (2) selecting and creating the control points; (3) modifying the control points where necessary, and (4) defining and adjusting the EE orientation at each control points.

4.1. Path planning

Given a pick-and-place task, instead of searching directly for a collision-free path, the proposed approach first creates a series of control points within the CFV to form a path. Next, the path is optimized using convex

optimization techniques. A bounding cylinder enclosing both the EE and the virtual object is used as a swept model to facilitate the CFV check, which determines whether the swept model is within the CFV.

4.1.1. Control point creation

The control points created must be accessible by the EE, and the swept model at each control point should be within the CFV. The bounding cylinders corresponding to the control points can be defined by Equation (1), where r and h are the radius and height; $\mathbf{o}_i$ and $\mathbf{z}_i$ are the origin and axis of the cylinder corresponding to the ith control point; N_p is the number of control points; and t_i is the index of the control points.

$$BC = \left\{ BC_i\left(\mathbf{o}_i, \mathbf{z}_i, r, h, t_i\right) \in \Re^3; i = 1, 2, ..., N_p \right\} \quad (1)$$

4.1.2. Control point modification

A path formed by a list of control points may not be collision-free even if these points are within the CFV. Therefore, there may be a need to modify these points to generate a new path. A Euclidean distance-based method is proposed to assist the user in selecting a control point. This method computes the distance between the probe and each control point, and associates the value with the corresponding point. The distances are updated automatically as the probe moves, and the control point that has the minimum distance to the probe is highlighted as a candidate point to be selected. Equation (2) gives the definition of a control point to be selected, where $\mathrm{S}(\mathbf{o}_0, \mathbf{v}_i)$ is the Euclidean distance between $\mathbf{o}_0$ (the tip of the probe) and $\mathbf{v}_i$ (the ith control point).

$$\boldsymbol{v}_{poi} : \mathrm{S}\left(\mathbf{o}_0, \boldsymbol{v}_{poi}\right) = \min\left\{ \mathrm{S}\left(\mathbf{o}_0, \mathbf{v}_i\right); i = 0,1,2,...,N_p \right\} \quad (2)$$

4.1.3. Path generation

If the path generated through interpolation is within the CFV, it will be collision-free in the joint space. Two issues need to be addressed in path interpolation, namely, the number of control points, and the data spacing between the control points.

In the control point creation stage, a small number of control points may generate a path where one or more segments are likely to be outside the CFV. On the other hand, an excessive number of control points may produce an overly fitted path. As the number of control points is task-dependent, a user can first select a small number of control points (e.g., five) to perform the interpolation, and insert new points progressively where necessary.

Considering the Euclidean distance between two consecutive control points, a normalized measurement can be assigned to each point as a time stamp for path interpolation. It is proportional to the cumulative distance from the start point to this point. The time stamp for the jth control point is given by Equation (3), where d_i is the Euclidean distance between the jth and (j-*1*)th points. The two end-points are $t_0 = 0$ and $t_{N_p+1} = 1$.

$$t_j = \frac{\sum_{i=1}^{j} d_i}{\sum_{i=1}^{N_p+1} d_i}, \; j = 1,2,...,N_p \quad (3)$$

4.2. EE orientation planning

A curve model can be represented by a sequence of parameterized points with even interpolation intervals. This representation is useful for trajectory optimization where the curve has to be redefined in a scalar path coordinate by evenly discretized points. The curve representation is given by Equation (4), where N_s is the number of the parameterized points; $\mathbf{p}_k$ is the kth point, and the auxiliary index for this point is $a_k = k/N_s$.

$$P = \left\{ P_k\left(k, a_k, \mathbf{p}_k\right) \in \Re^3; k = 0,1,.....,N_s \right\} \quad (4)$$

Before proceeding to plan the EE orientation, the Euclidean distance-based method as described in Section 4.1, is used to select the list of control points among the parameterized points of the curve. In the modification mode, a point from the control point list can be deleted (Fig 2(a)), or a point be inserted into the existing list (Fig 2(b)), where $N_p = N_s$ in Equation (2).

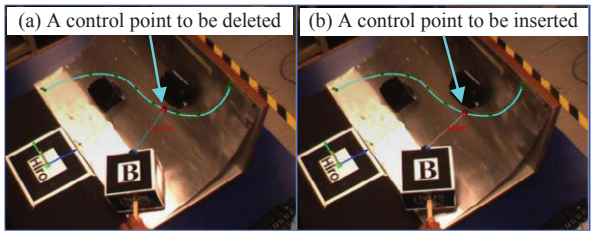

Fig 2. Control point selection

4.2.1. End-effector orientation at a control point

A data structure associated with each control point is illustrated in Fig 3. The parameters defined in the data structure are listed in Table 1. α is task-dependent and predefined with respect to the Z-axis of the coordinate system defined at each control point; k, a, p and α can be determined for each point during the control point selection stage. The rest of the parameters can be initialized as zero and specified later during the EE orientation planning stage.

A coordinate frame is first defined at the start of the curve. The coordinate frames of other control points can be defined by applying the transformations reflecting the

changes in the curve direction, which define ***R***. The EE orientations at these points are used to define *e* and *β*.

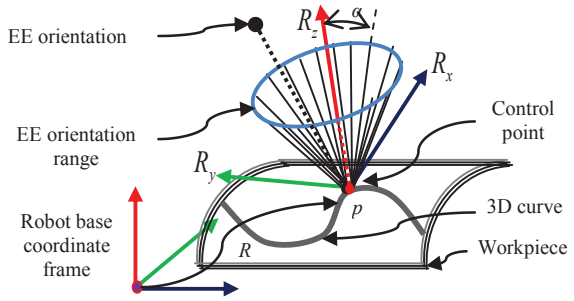

Fig 3. Parameters associated with a control point

Table 1. Data structure of the control points

k	Index of control point
a	Auxiliary index of control point
p	Positional vector of coordinate frame defined on control point
R	Rotation matrix of coordinate frame defined on control point
α	EE inclination range
e	Vector defines the orientation of the EE
β	Angle between EE and *z*-axis of robot base frame

4.2.2. End-effector orientation interpolation

At any parameterized point on the curve, an EE orientation can be represented with respect to the robot base frame by a unit vector. The angle between the unit vector and the *Z*-axis of the robot base frame can be interpolated. The interpolation parameters are the same as the parameters in Equation (4). The EE orientations along the curve are given by Equation (5). Hence, a ruled surface [18] can be given by Equation (6), where P_m is the curve model and it defines the ruled surface *directrix*; W_m defines the direction of the *ruling*; v is a the length of the *ruling*.

$$W = \left\{ W_m(l, a_m, \beta_m) \in \Re^3; m = 0,1,.....,N_s \right\} \quad (5)$$

$$G_{rs} = \left\{ P_m + v \cdot W_m; m = 0,1,...,N_s \right\} \quad (6)$$

4.3. Trajectory optimization and simulation

In this research, a trajectory minimizing the path duration subject to joint torque and velocity constraints is determined through solving a convex optimization problem using the log-barrier method [15]. The objective function for time-optimal trajectory is represented using the parameters associated with a scalar path coordinate, and the penalty function for the constraints is given by an averaging sum-log function [16]. The implementation of trajectory optimization considering joint torque constraints has been described in Ref. [19]. The recursive Newton-Euler algorithm [17] is used to model the dynamic behavior of the robot. In this research, only the dynamics of the positional DOF of the Scorbot-ER VII robot are considered. An estimation of these parameters is adopted from [20].

The trajectory can be simulated using a virtual robot under a discrete Proportional-Derivative (PD) control scheme. A normalized measurement associated with the simulated torque and velocity, as defined in Equation (7), is compared along the trajectory. $\tau_{sim}(i)$ and $\dot{q}_{sim}(i)$ are the simulated torque and velocity of joint *i*; $\bar{\tau}(i)$ and $\bar{\dot{q}}(i)$ are their upper bounds.

$$m_u(i) = \max\left\{ \left| \frac{\tau_{sim}(i)}{\bar{\tau}(i)} \right|, \left| \frac{\dot{q}_{sim}(i)}{\bar{\dot{q}}(i)} \right| \right\} \quad (7)$$

From Equation (7), if $m_u(i) \in (0,1)$, both the computed torque and velocity for each joint/link are within their limits. In this case, the link that has the largest torque and velocity among all the links is highlighted to indicate that it is the link most likely to deviate from the planned path. If $m_u(i) > 1$, the computed torque or velocity for joint *i* violates their constraints, the simulation will stop with link *i* highlighted. The user can adjust the control gain relevant to this joint, and execute the simulation again. Some rules can be applied for rough tuning of the control gains based on the cues observed during simulation, i.e., the derivative gains usually have the largest effect on the output of the control system; the proportional gains will need to be tuned if the simulated trajectory is found to have shifted away from the planned trajectory.

5. Case studies and discussions

5.1. Case study I

This case study demonstrates a pick-and-place task for transferring an object from a start point to a goal point in a full-scale environment. Fig 4 shows the process of planning a collision-free path for this task. Fig 5 illustrates the procedure to modify the control points.

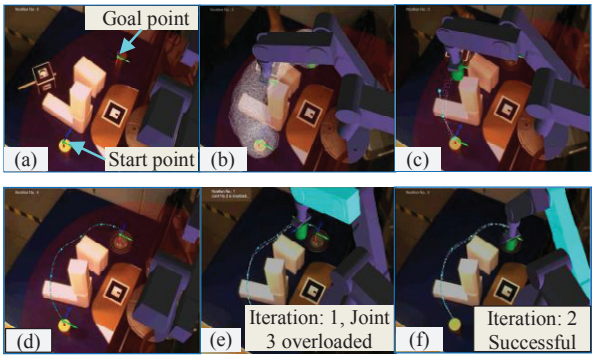

Fig 4. Case study I. (a) Task setup; (b) CFV generation; (c) Creation of control points; (d) Geometric path generation; (e) Trajectory simulation: unsuccessful trial; (f) Successful trajectory simulation

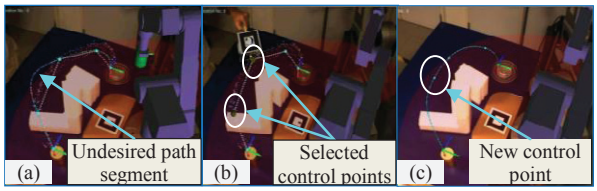

Fig 5. Control points modification. (a) Initial path formed; (b) Control points selection; (c) Path re-generation

5.2. Case study II

Fig 6 demonstrates the proposed method for EE orientation planning along a spatial circular curve, where the orientation of the EE needs to avoid the obstacles and the edge along the curve. This case study is designed to emulate robotic arc welding, gluing, etc.

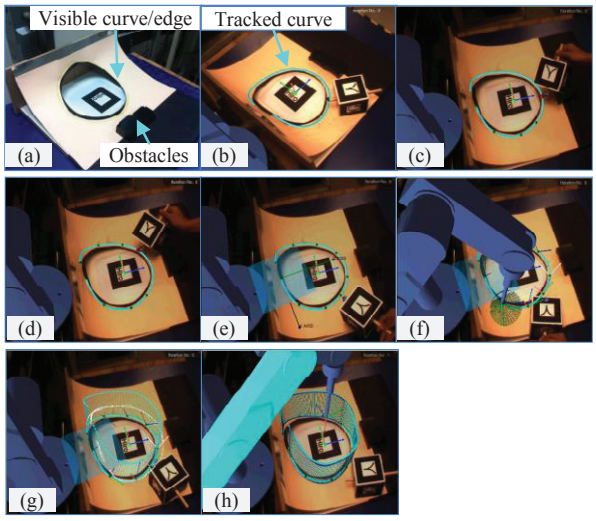

Fig 6. Case study II. (a) Workpiece; (b) circular curve model; (c) control points selection; (d) modification of control points; (e) definition of coordinate frame at curve model; (f) definition of EE orientation at a control point; (g) generation of ruled surface; (h) successful trajectory simulation.

5.3. Case study III

Fig 7 demonstrates the method for EE orientation planning along a spatial S-shaped curve which lies on a curved surface. The planning results show that the robot EE is able to travel along the path at orientations within an acceptable range with respect to the path.

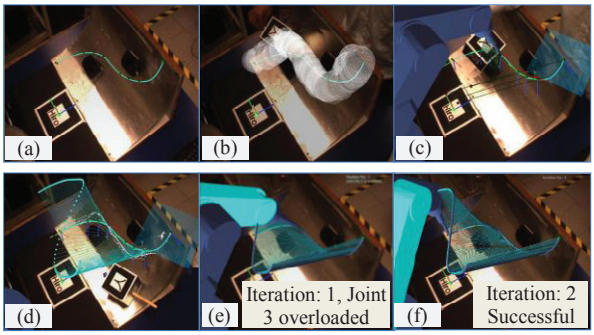

Fig 7. Case study III. (a) Output curve and selected control points; (b) CFV generation; (c) Definition of EE orientation at each control point; (d) Ruled surface generation; (e) Trajectory simulation: unsuccessful trial; (f) Successful trajectory simulation

5.3.1. Accuracy evaluation

There are two main sources of errors affecting the overall accuracy in this case study, namely, curve tracking error due to the tracking method adopted, and robot modeling error, i.e., the kinematics and dynamics modeling error. Fig 8 depicts the tracking errors in the X-, Y- and Z-directions in this case study. The average tracking error is nearly 11.0mm, with the camera installed at 1.5m away from the workpiece. The error is mainly caused by the ARToolKit tracking method adopted in the process of acquiring 3D data points. However, it should be noted that the EE orientation planning process would not introduce additional errors as the control points are selected from the curve model. To this end, the tracking errors do not severely affect the proposed method for planning the EE orientation along a visible curve.

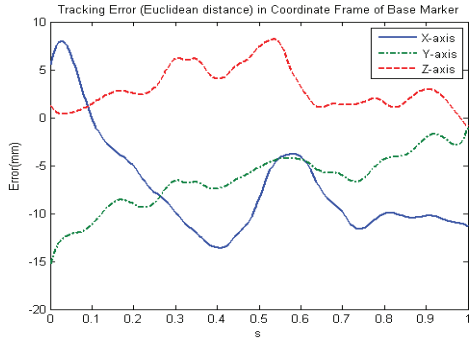

Fig 8. Curve tracking error in case study III

The trajectories determined in this case study have been tested on the Scorbot-ER VII robot. The plots in Fig 9 are the actual trajectories of the EE in the Cartesian coordinates, as compared with the referenced curve. The discrepancies shown in Fig 9 are mainly caused by the robot dynamics modeling errors where only a set of estimated robot dynamics parameters are implemented in the trajectory optimization process.

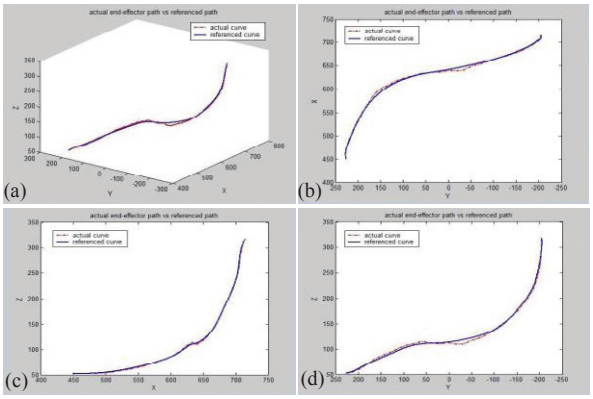

Fig 9. Trajectory implemented on real robot

6. Conclusions and future work

In this research, methodologies integrating AR technology have been proposed to assist users in robot path and EE orientation planning incorporating robot dynamics. A Euclidean distance-based method has been developed to facilitate control points selection and modification. A convex optimization method has been implemented to obtain an approximated time-optimal trajectory. Prior to translation into robot controller codes, the planned trajectory is simulated with a virtual robot and controller parameters can be adjusted gradually during simulation. The three case studies demonstrate the RPAR-II approach in planning pick-and-place tasks and path following tasks.

A number of areas can be further explored to improve the proposed system. A more accurate and robust tracking method can be developed to enhance the accuracy of this system. Stereo camera with unparalleled optical axes will improve the accuracy of the disparity map, yielding better performance in stereo depth estimation. Improvement can be made to develop a more intuitive and non-distracting interface facilitating robot EE orientation definition and modification. The system can be extended to suit other types of tasks, such as robotic painting, assembly, etc., where both pick-and-place operations and path following operations may be required for a single task.

References

[1] Thrun, S., 2004. Toward a Framework for Human-Robot Interaction, Human-Computer Interaction, 19/1-2:9-24.
[2] Breazeal, C., Edsinger, A., Fitzpatrick, P., Scassellati, B., 2001. Active Vision for Sociable Robots, IEEE Transactions on System, Man, and Cybernetics, Part A: Systems and Humans, 31/5:443-53.
[3] Marin, R., Sanz, P.J., Nebot, P., Wirz, R., 2005. A Multimodal Interface to Control a Robot Arm via the Web: A Case Study on Remote Programming, IEEE Transactions on Industrial Electronics, 52/6:1506-20.
[4] Chintamani, K., Cao, A., Ellis, R.D., Pandya, A.K., 2010. Improved Tele-manipulator Navigation during Display-Control Misalignments using Augmented Reality Cues, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 40/1:29-39.
[5] Jara, C.A., Candelas, F.A., Gil, P., Fernández, M., Torres, F., 2009. "An Augmented Reality Interface for Training Robotics through the Web," 40th International Symposium on Robotics. Barcelona, Spain: 10-13 March 2009, p.189-194.
[6] Bischoff, R., Kazi A., 2004. "Perspectives on Augmented Reality based Human-robot Interaction with Industrial Robots," International Conference on Intelligent Robots and Systems. Sendai, Japan: 28 September - 2 October 2004, p.3226-31.
[7] Chong, J.W.S., Ong, S.K., Nee, A.Y.C., Youcef-Youmi, K., 2009. Robot Programming using Augmented Reality: An Interactive Method for Planning Collision-Free Paths, Robotics and Computer-Integrated Manufacturing, 25/3:689-70.
[8] Zaeh, M.F., Vogl, W., 2006. "Interactive Laser-Projection for Programming Industrial Robots," 5th International Symposium on Mixed and Augmented Reality, Santa Barbara, CA: 22-25 Oct 2006, p.125-8.
[9] Reinhart, G., Munzert, U., Vogl, W., 2008. A Programming System for Robot-based Remote-Laser-Welding with Conventional Optics, CIRP Annals–Manufacturing Technology, 57/1:37-40.
[10] Ong, S.K., Chong, J.W.S., Nee, A.Y.C., 2010. A Novel AR-based Robot Programming and Path Planning Methodology, Robotics and Computer-Integrated Manufacturing, 26/3:240-9.
[11] He, X., Chen, Y., 2009. Haptic-aided robot path planning based on virtual tele-operation, Robotics and Computer-Integrated Manufacturing, 25/4-5:792-803.
[12] Papakostas, N., Alexopoulos, K., Kopanakis, A., 2011. Integrating Digital Manufacturing and Simulation Tools in the Assembly Design Process: A Cooperating Robots Cell Case, CIRP Journal of Manufacturing Science and Technology, 4/1:96-100.
[13] Aleotti, J., Caselli, S., Reggiani, M., 2004. Leveraging on a virtual environment for robot programming by demonstration, Robotics and Autonomous Systems, 47/2-3:153-61.
[14] Craig, J.J., 2005. Introduction to Robotics, Mechanics and Control, Pearson Education Inc., NY.
[15] Boyd, S., Vandenberghe, L., 2004. Convex Optimization, University Press, Cambridge.
[16] Verscheure, D., Diehl, M., De Schutter, J., Swevers, J., 2009. "On-line time-optimal path tracking for robots," International Conference of Robotics and Automation. Kobe, Japan: 12-17 May 2009, p.599-605.
[17] Roboop: http://www.cours.polymtl.ca/roboop, last accessed on 5 Nov 2011.
[18] Tsai, M.J., Stone, D.J., 2009. Inverse velocity analysis for line guidance five-axis robots, Robotics and Computer-Integrated Manufacturing, 25/4-5:736-45.
[19] Fang, H.C., Ong, S.K., Nee, A.Y.C., 2012. Interactive trajectory planning and simulation using Augmented Reality, Robotics and Computer-Integrated Manufacturing, 28/2:227-37.
[20] Constantinescu, D., Croft, E.A., 2000. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths, Journal of Robotic Systems, 17/5:233-49.