Trajectory Following Controller with Vibrational Motion
Developed Based on ORiN Middleware Interface

Fusaomi Nagataa,*, Shohei Hayashi*a, Tomoya Nagatomi*a, Akimasa Otsuka*a, Keigo Watanabeb
aTokyo University of Science, Yamaguchi, 1-1-1 Daigaku-Dori, Sanyo-Onoda 756-0884, Japan
bOkayama University, 3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan

*Corresponding Author: nagata@rs.tus.ac.jp

Abstract

Industrial robots with an open architecture have been applied to several tasks such as machining, sanding, polishing and so on. The open architecture allows system engineers to realize desirable teachingless operation. However, porting of an application software to other makers’ industrial robots is not easy even though having an open architecture, because the specifications of interface between a PC and robots differ respectively. In this paper, trajectory following control with vibrational motion is developed on ORiN (Open Robot/Resource interface for the Network) middleware interface for a small educational robot. The design with ORiN interface and its experimental results are shown.

Keywords: Industrial robot, VE026A, ORiN (Open Robot/Resource interface for the Network), Standardization of robotic interface, Trajectory following control, Vibrational motion control, CL data.

1. Introduction

Activities for the standardization in industrial automation are reported in order to enhance reusability and compatibility of elaborate robotic applications[1-3]. Up to now, industrial robots with an open architecture have been applied to several high-skilled tasks such as machining, sanding, polishing and so on. The open architecture allows system engineers to realize desirable teachingless operation. However, transplantation, i.e., porting of an application software to other makers’ industrial robots is not easy even though having an open architecture, because the specifications of interface between a PC and robots are different respectively. That is the reason why the standardization of industrial robots’ interfaces is now being expected and desired from users. Recently, ORiN (Open Robot/Resource interface for the Network) is gathering attention as a promising middleware for the standardization of robotic interfaces[4-6].

In this paper, trajectory following control using vibrational motion is developed on ORiN interface for a small–size educational robot VE026A with six–DOFs. The ORiN middleware is available on the robot. Trajectory following control of an end–effector attached to the flange of the robot is a basic but important strategy for a robotic application without using any robot languages. Here, a vibrational motion control is further considered to efficiently remove undesirable cusp marks which come out in machining process using a ball–end mill. The design with ORiN middleware interface and its experimental results are shown.

2. Robotic Applications Developed in the Past

In this section, our past developed robotic systems using open architecture controllers are introduced[7]. The open architecture concept allows us to efficiently develop a robotic application software without using a conventional device called a teaching pendant and a robot language.

Fig. 1. Polishing robot for PET bottle blow molds developed based on JS10 provided by Kawasaki Heavy Industries, LTD. in 1997.

Figure 1 shows the polishing robot for PET bottle blow molds developed based on JS10 with six–DOFs provided by
Kawasaki Heavy Industries (KHI), LTD. in 1997. The JS10 equipped with a PC-based controller having a concept of open architecture. The tip of the robot arm could be controlled by giving a relative homogeneous transformation matrix $\Delta T$ to the robotic servo system every sampling period of 4 ms. The matrix is written by

$$
\Delta T = \begin{bmatrix}
\Delta n_x & \Delta n_y & \Delta n_z & \Delta p_x \\
\Delta o_x & \Delta o_y & \Delta o_z & \Delta p_y \\
\Delta a_x & \Delta a_y & \Delta a_z & \Delta a_z \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

(1)

API functions for forward kinematics ($\theta \rightarrow T$) and inverse one ($T \rightarrow \theta$) were also provided by KHI. $\theta \in \mathbb{R}^{n\theta}$ is the joint angle vector [rad]. In order to be able to set the sampling period shorter than 4 ms, Windows multimedia timer was used. The minimum resolution is 1 ms. In this system, the orientation of a ball–end abrasive tool, i.e., normalized direction vector, was always fixed to $[0 \ 0 \ -1]^T$ as shown in Fig. 1 to efficiently abrade the contour. The radius of the ball–end tip was 5 mm. Cutter location data called CL data forming a zigzag path or a spiral one were generated by CAD/CAM Unigraphics, and which were used for desired trajectory of the tool to perform teachingless operation.

Figure 2 shows the robot sander for wooden workpieces with curved surface(8). The system was developed based on FS20 with six–DOFs provided by KHI, LTD. in 2000, whose controller has almost the same software specification as JS10’s one. In this system, the orientation of an air–driven sanding tool was controlled to be normal to surface by referring the direction components in CL data. In addition, the orientation could be manually controlled via a force feedback joystick.

Fig. 2. Robot sander for wooden workpieces developed based on FS20 provided by Kawasaki Heavy Industries, LTD. in 2000.

Also, Fig. 3 shows another mold polishing robot for PET bottle blow molds developed based on Motoman UP6 provided by Yaskawa Electric Corp. in 2002. Fortunately, the robotic controller had a similar concept as FS20’s one, so that the porting of the robotic application was not difficult and complicated. The tip of the robot arm could be controlled by giving $\Delta T$ to the technically opened servo system every sampling period of 10 ms.

Fig. 3. Mold polishing robot for PET bottle blow molds developed based on Motoman UP6 provided by Yaskawa Electric Corp. in 2002.

Further, Fig. 4 shows an industrial robot PA10 with seven–DOFs provided by Mitsubishi Heavy Industries, Ltd. When controlling this robot, the position and orientation matrix of the tool tip had to be given to the servo system, which was written by the following recursive equation.

$$
T[10k] = T[10(k-1)] + \Delta T(k) \quad (k = 1, 2, 3 \cdots)
$$

(2)

where $T[10k]$ is the homogeneous transformation matrix absolutely representing the position and orientation of the arm tip in robot coordinate system in case that the sampling period is set to 10 ms. As can be seen, PA10 is controlled by the absolute command matrix $T[10k]$ in robot coordinate system. The PA10 was applied to the painting process for artistic relief wall in Interior Design Research Institute, Fukuoka Industrial Technology Center in 2005(9).

Fig. 4. Open architectural industrial robot PA10 provided by Mitsubishi Heavy Industries, Ltd. for painting of relief wall in 2005.

Next, a machining robot for foamed polystyrene
materials is introduced. Figure 5 shows the machining robot being developed based on an industrial robot RV1A which is provided by Mitsubishi Electric Corporation. RV1A is usually used in engineering fields. The machining robot will be used for engraving foamed polystyrene materials. When the position and orientation of the arm tip are controlled, a command vector \(\mathbf{x}(k) \mathbf{o}(k)^T\) is transmitted to the robotic servo controller through Ethernet with UDP protocol. \(\mathbf{x}(k) = [x(k) y(k) z(k)]^T\) [mm] is the position vector in robot absolute coordinate system. Also, \(\mathbf{o}(k) = [\phi(k) \theta(k) \psi(k)]^T\) [rad] is the orientation vector of the arm tip. Note that the arm tip’s orientation is controlled with roll angle \(\phi(k)\), pitch angle \(\theta(k)\) and yaw angle \(\psi(k)\).

These robotic applications using articulated industrial robots have innovated a common important function for realizing the desired teachingless operation. The function is the position/orientation controller based on CL data called CAD/CAM–based trajectory following controller. The trajectory following controller can be realized by simultaneously moving position and orientation of arm tip. As an instance, Fig. 6 shows the block diagram of the trajectory following controller incorporated in the robot sander shown in Fig. 2. \(\mathbf{v}_p(k) \in \mathbb{R}^{3 \times 1}\), \(\mathbf{v}_o(k) \in \mathbb{R}^{3 \times 1}\) and \(\mathbf{v}_f(k) \in \mathbb{R}^{3 \times 1}\) are velocities to control position/orientation, contact force and pick feed, respectively. \(\mathbf{x}_p(k)\), \(\mathbf{o}_p(k)\) and \(\mathbf{f}_p(k)\) are desired position vector, orientation vector and force norm, respectively. \(\mathbf{x}(k)\) and \(\mathbf{f}(k)\) are position/orientation vector and force vector measured by respective sensors. \(\mathbf{S}_o(k) \in \mathbb{R}^{3 \times 3}\) is the switch matrix to set an active direction of position feedback control for accurate feed.

One of current problems is that transplantation of application software to other makers’ industrial robots is not easy even though having an open architecture, because the software specifications of interfaces between a PC and robots are not identical. That is the reason why standardization of industrial robots’ interfaces is now being expected and desired from users.

Fig. 6. Block diagram of CAD/CAM–based trajectory following controller and force controller used in the robot sander shown in Fig. 2.

3. About ORiN

ORiN means Open Robot/Resource interface for the Network and provides a standard network interface for factory automations, robots and other devices. After the Japan Robot Association established the ORiN in 2002, the ORiN Forum has produced unified methods for standardization. Recently, ISO20242-Part4 including applications based on ORiN was officially approved as an international standard. For example, a robotic application software developed based on ORiN is able to not only obtain sensor information easily but also transmit control input to the servo system, without considering the differences among robotic companies about the hardware type, the communication protocol between the PC and the robot, or other hardware specifications.

It is expected that if a robotic application, e.g., trajectory following controller, is developed using API functions provided by ORiN, the application can be straightforwardly ported on other industrial robots with ORiN middleware interface. Figure 7 illustrates the advantage of using ORiN, in which promising standardization can be seen.

Fig. 7. Advantage of portability of robotic application through ORiN middleware interface.
4. Trajectory Following Control with Vibrational Motion

Trajectory following controller with vibrational motion is proposed to efficiently remove undesirable cusp marks which appear in machining process using a ball–end mill. Figure 8 illustrates the image of the cusp marks. Figure 9 shows the vibrational motion vectors \( \mathbf{v}_i \) yielding in the orthogonal directions along original CL data. \( \mathbf{p}(i) \) is the position vector in CL data, \( \mathbf{v}_i(k-1), \mathbf{v}_i(k+7) \) and \( \mathbf{v}_i(k+15) \) are vibrational motion vectors online generated at the discrete time \( k-1, k+7 \) and \( k+15 \). When the vibrational motion is not applied, the ball–end mill straightly moves from \( \mathbf{p}(i-1) \) to \( \mathbf{p}(i) \) with a given velocity \( \mathbf{v}_i \). On the other hand, if the vibrational motion is generated, the mill’s tip follows the trajectory as \( \mathbf{x}_d(k-3) \), i.e., \( \mathbf{p}(i-1) \rightarrow \mathbf{x}_d(k-2) \rightarrow \mathbf{x}_d(k-1) \rightarrow \mathbf{x}_d(k) \rightarrow \cdots \rightarrow \mathbf{x}_d(k+20) \rightarrow \mathbf{x}_d(k+21) \rightarrow \mathbf{x}_d(k+22) \rightarrow \cdots \rightarrow \mathbf{p}(i) \). Note that, for example, \( \mathbf{x}_d(k) \) and \( \mathbf{x}_d(k+4) \) are located at the same point on the round–trip path via \( \mathbf{x}_d(k+2) \), i.e., \( \mathbf{v}_i(k-1) \).

![Image 8](https://example.com/image8)

Fig. 8. Image of cusp marks caused by the trajectory of a ball–end mill in removal machining process.

Here, how to generate \( \mathbf{v}_i(k) \) is explained. First of all, \( \mathbf{v}_i(k) \) is perpendicular to normal direction vector \( \mathbf{o}_x(k) \) in CL data, so that the following relation is obtained.

\[
v_{xz}(k) o_{xz}(k) + v_{yz}(k) o_{yz}(k) + v_{zz}(k) o_{zz}(k) = 0 \tag{3}
\]

Also, \( \mathbf{v}_i(k) \) and direction vector \( \mathbf{t}(k) \) are orthogonal each other, so that

\[
v_{xz}(k) t_x(k) + v_{yz}(k) t_y(k) + v_{zz}(k) t_z(k) = 0 \tag{4}
\]

Because both \( \mathbf{o}_x(k) \) and \( \mathbf{t}(k) \) are known in advance, \( \mathbf{v}_i(k) \) can be determined by simultaneously solving Eqs. (3) and (4). Further by using a sign function \( \text{sign}(k) \), a scalar \( K_v \) and a random number \( N_v(k) \) \( (0 \leq N_v(k) \leq 1) \), the vibrational motion vector is finally generated as

\[
\mathbf{f}(k) = \text{sign}(k) K_v N_v(k) \frac{\mathbf{v}_i(k)}{\|\mathbf{v}_i(k)\|} \tag{5}
\]

where

\[
\text{sign}(k) = \begin{cases} 
1 & \text{if } k \text{ = odd number} \\
-1 & \text{otherwise}
\end{cases} \tag{6}
\]

As can be seen, the length of the vibration vector depends on \( K_v \) and \( N_v \). \( \mathbf{f}(k) \in \mathbb{R}^{3 \times 1} \) is a velocity vector to yield small and ununiform vibrational motion, and which is given to a ball–end mill’s tip alternately changing the direction with a given frequency.

![Image 9](https://example.com/image9)

Fig. 9. Vibrational motion vector generated along CL data.

Figure 10 shows machining results of a formed polystyrene material, in which zigzag paths without and with vibrational motion were given to the trajectory following controller in the machining robot. It is observed that undesirable cusp marks could be efficiently suppressed by using the proposed vibrational motion control.

In the next section, the trajectory following controller with vibrational motion is tried to be ported to an articulated robot with ORiN middleware interface.

![Image 10](https://example.com/image10)

Fig. 10. Machining results of foamed polystyrene material without and with the proposed vibrational motion control.)
5. Transplantation of Trajectory Following Controller to VE026A with ORiN Interface

The robotic applications introduced up to here are developed based on each robot maker’s interface, so that the transplantation among different makers’ robots is complicated and time-consuming. Recently, ORiN is gathering attention as a promising middleware for standardization of robotic interface.

In this section, a transplantation, i.e. porting, of the CAD/CAM–based trajectory following controller with vibrational motion is tried to evaluate ORiN2 SDK. Figure 11 shows the small-sized educational articulated robot VE026A incorporated with ORiN2 SDK provided by DENSO WAVE Incorporated (12,13). Standard programming interface called CAO (Controller Access Object) works as common interface and functions between the application and the robot.

![Six-DOFs articulated robot arm VE026A incorporated with ORiN interface provided by DENSO WAVE Incorporated.]

Among methods provided by CAO interface, CaoGetPose( ) was used to get the current position and orientation \([x^T(k)\ 0^T(k)]^T\) of the arm tip. Also, CaoMove( ) was used in order to give the desired position and orientation \([x^T_d(k)\ 0^T_d(k)]^T\) to the VE026A every sampling period. In other words, it seems that we have only to use CaoGetPose( ) and CaoMove( ) in order to transplant the trajectory following controller with vibrational motion into the VE026A.

![Example of control results without and with vibrational motion along a spiral path.](image)

After the implementation on Windows Visual C++, it was confirmed that the tip of the robot arm could be successfully controlled with a command vector similar to the case of the machining robot based on RV1A. It should be noted that the orientation was dealt with \(\sigma(k)=[\varphi(k)\ \theta(k)\ \psi(k)]^T\ \text{[deg]},\ i.e.,\ not\ radian\ but\ degree\ in\ case\ of\ ORiN2\ SDK.\ Figure\ 12\ shows\ examples\ of\ the\ control\ results\ without\ and\ with\ vibrational\ motion\ control.\ It\ was\ further\ confirmed\ that\ the\ ORiN2\ SDK\ enabled\ the\ application\ software,\ i.e.,\ trajectory\ following\ controller\ with\ vibrational\ motion,\ to\ be\ easily\ transplanted\ into\ the\ VE026A.

6. Conclusions

Recently, ORiN is gathering attention as a promising middleware for the standardization of robotic interface. In this paper, CAD/CAM–based trajectory following controller using vibrational motion was introduced for a machining robot. Then, the controller was successfully ported into an educational small–sized robot VE026A by using ORiN middleware interface in a short period of time. Higher reusability and portability due to using ORiN2 SDK could be confirmed through the design and experiments. The proposed CAD/CAM–based trajectory following controller with vibrational motion will be able to be straightforwardly transplanted to other makers’ robots incorporated with the ORiN middleware interface.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 25420232.

References

(4) M. Mizukawa, T. Koyama, T. Inukai, A. Noda, N. Kanamaru, Y. Noguchi and N. Otera : “Proposal of


(11) ORiN FORUM, http://www.orin.jp/e/
