Trajectory Generation for Assembly Tasks Via Bilateral Teleoperation

Ghazaei, Mahdi; Cho, Jang Ho; Johansson, Rolf; Robertsson, Anders

Published in:
[Host publication title missing]

DOI:
10.3182/20140824-6-ZA-1003.02559

2014

Citation for published version (APA):
Trajectory Generation for Assembly Tasks via Bilateral Teleoperation

M. Mahdi Ghazaei Ardakani* Jang Ho Cho**
Rolf Johansson*** Anders Robertsson****

* Faculty of Engineering (LTH), Lund University, SE 221 00 Lund, Sweden (e-mail: mahdi.ghazaei@control.lth.se)
** Korea Institute of Machinery & Materials (KIMM), South Korea (e-mail: jangho@kimm.re.kr)
*** Faculty of Engineering (LTH), Lund University, SE 221 00 Lund, Sweden (e-mail: rolf.johansson@control.lth.se)
**** Faculty of Engineering (LTH), Lund University, SE 221 00 Lund, Sweden (e-mail: anders.robertsson.lth.se)

Abstract: For assembly tasks, the knowledge of both trajectory and forces are usually required. Consequently, we may use kinesthetics or teleoperation for recording human demonstrations. In order to have a more natural interaction, the operator has to be provided with a sense of touch. We propose a bilateral teleoperation system which is customized for this purpose. We introduce different coordinate frames to make the design of a 6-DOF teleoperation straightforward. Moreover, we suggest using tele-admittance, which simplifies instructing the robot. The compliance due to the slave controller allows the robot to react quickly and reduces the risk of damaging the workpiece.

Keywords: Teleoperation, Robotics, Force control, Assembly task, Trajectory generation.

1. INTRODUCTION

Robots have been used for decades to conduct repetitive tasks, e.g., assembly and pick-and-place, to replace human beings. Numerous industrial environments have benefited by employing robotic systems in terms of qualitative outcomes and efficient processing. In turn, the demand of robotic systems has been increasing in various application areas in order to deal with more complex and more flexible tasks. However, generating robotic commands for complex and flexible tasks has been a challenging problem. There are numerous suggested solutions in the literature and many of them rely on trajectory programming prior to the operation. In general, those methods that require state-event modeling is not easily implementable or compatible for different tasks.

Generating robotic motions is usually based on direct teaching via some human robot interface. The manipulation of the robotic systems is possible not only with the use of a teach pendant but also with direct manipulation (so called lead-trough programming). The direct manipulation of robots is way more intuitive than the conventional programming since trajectories are generated via human demonstration. Hence, it has been widely used in human skill acquisitions, Argall et al. (2009); Lee et al. (2012). However, the interface between human and robots is not convenient and difficult to transfer accurate motions due to mechanical properties of robots such as inertia and friction. Hence, it is mainly used for small and light robots. Although compliant motion control could be employed to reduce inertial/frictional force, it is still very difficult to get instantaneous motion with industrial robots.

An alternative approach to the kinesthetic one for generating trajectories is teleoperation of a robot. In the teleoperation setup, a human operator manipulates a “master device” to control a “slave robot” in a remote environment. The master device is to decode the commands issued by an operator and forward them to the slave robot. For precise remote control of the slave robot, it is required to provide sensory feedback such as visual, haptic, or aural feedback to the operator. Haptic feedback provides the operator with the sense of touch from what the robot may perceive. Hence, it is essential when there is a contact between the slave robot and the environment. The haptic feedback establishes a bilateral communication channel between the operator and the robot besides the visual feedback.
The control of bilateral teleoperation systems has been extensively studied for decades, see (Hokayem and Spong, 2006) and references therein. A schematic diagram of a bilateral teleoperation system is illustrated in Fig. 1. The control architecture represents which information has to be exchanged between two sides (Aliaga et al., 2004). The ultimate goal of bilateral teleoperation is to achieve transparency by means of two-directional position and force tracking (Lawrence, 1993; Yokokohji and Yoshikawa, 1994). Despite recent advances in bilateral teleoperation in the task space (Liu and Chopra, 2012; Wang, 2013), experiments are still mainly limited to the robots with a few degrees-of-freedom. In general, the design of bilateral controller for industrial robots is challenging due to practical problems including inertia, friction, control bandwidth, and time delay.

In this paper, we propose a convenient framework to generate robotic trajectories within the teleoperation setup. Since stiff contacts are very difficult to handle in bilateral teleoperation, a compliant motion control have been employed. Moreover, the targeted compliance of the slave robot can be adjusted by the operator in real-time. This tele-admittance idea increases flexibility for various tasks. We implemented the proposed approach in a 6 DOF setting and verified it with a couple of assembly tasks.

The main contributions of this paper can be formulated as follows:

- Introduction of slave-assisted teleoperation by means of compliance
- A framework for defining a mapping between a master and a slave robot
- A 6 DOF setup to collect trajectory and interaction forces for teaching an industrial robot

The remainder of this paper is organized as follows. In Section 2, a mapping between a master device and a slave robot is defined. The control architecture of the system is described in terms of tele-admittance and haptic feedback to the operator in Section 3. In Section 4, the experimental results are illustrated with the interaction with a solid block, peg-in-hole and snap-fit tasks. Several issues raised in the paper are discussed in Section 5. Finally, a summary and some concluding remarks are provided in Section 6.

2. KINEMATIC CHAIN

A mapping between the DOF of a master device and a slave robot must be defined. To have a natural and generic approach we make use of the following feature frames. Each feature frame defines a coordinate frame and is attached to an object according to De Schutter et al. (2007).

For the master device, we make use of:

1. Master Base frame: This is the basic coordinate system provided by the haptic device. The position and orientations values are relative to this frame.
2. Master Task frame: This defines a natural frame for manipulation tasks, which might be for example rotated with respect to the orientation of the haptic device.

Similarly, for the slave we define:

1. Slave Base frame: This is the default coordinate frame for a robot, which has usually its origin at the base of a robot.
2. Slave Task frame: This frame defines a coordinate system which is more convenient for specifying a task.
3. Flange frame: This is a frame firmly attached to the flange of the robot.
4. Slave frame (Tool frame): The origin of this frame is located at the tool center point (TCP). The frame defines the tool coordinate system.
5. Sensor frame: This frame is attached to the force/torque sensor and aligned with its coordinate system.
6. Wrist frame: This is a frame for active compliance of the robot. The origin of this frame acts as a virtual joint with respect to external forces. The designated principal moments of inertia are aligned with this frame.

Various slave coordinate frames are illustrated in Fig. 2. Fig. 3 represents the relationships between these frames. Each transformation corresponds to a translation and a rotation. The solid arrows represent the given transformations (either fixed or measurable). The dashed arrows correspond to the inferred transformations.

We define the mapping between the haptic interface and the robot as the translational and the rotational velocity coupling between the master frame with respect to the master task frame and the slave frame with respect to the slave task frame. In case of no scaling of motion and no interaction forces, this is equivalent to the matching of the slave frame and the master frame relative to their corresponding task frames.
3. CONTROL ARCHITECTURE

We employ a modified force-velocity control architecture, which requires position information of the master device and force information of the slave robot. Let us denote \( \mathbf{v}_m \) as the translation velocity of the master frame with respect to the master task frame. Then, \( \mathbf{v}_m \) is transmitted to the slave side with a motion scaling factor \( \gamma \) and used as a reference velocity for the slave robot. To enable micro motion at the slave side, \( \gamma \) is generally set to less than one.

3.1 Tele-admittance

Often due to limitations such as bandwidth and delay, it is not possible to have a fully transparent haptic teleoperation. Therefore, it is desirable to give the robot some extent of autonomy. Specially, when dealing with contact forces, the delay in the feedback might impede the operator to react in time. Consequently, a stiff industrial robot may cause workpiece or tool damages. A human-like strategy to mitigate this problem is adjusting the arm impedance. In case of high uncertainties, by reducing the impedance it is possible to reduce the interaction forces. Ajoudani et al. (2012) showed that transferring the desired impedance is an effective strategy for teleoperation. Thus, we allow the operator to adjust it remotely.

A 6 DOF admittance controller was implemented to achieve active compliance at the wrist point, Siciliano and Villani (1999); Caccavale et al. (1999); Hogan (1985). The controller provides an isotropic translational and rotational admittance behavior with tunable parameters. The robot acts as a mass-spring-damper system. The rotational part allows for different moment of inertia along each axis.

Similar to Villani and De Schutter (2008), we use the notation \( m, k_i, D \in \mathbb{R} \) for the translation part and \( M \in \mathbb{R}^{3 \times 3} \), and \( k_o, D_o \in \mathbb{R} \) for the rotational part. Using the admittance controller, we enforce the following dynamics for the wrist frame due to the vector of the external force \( \mathbf{F}_i \) and the vector of torque \( \tau_i \).

\[
\begin{align*}
\dot{\mathbf{p}} &= \mathbf{v} \\
\dot{m}\mathbf{v} &= -k_i\mathbf{p} - D\mathbf{v} + \mathbf{F}_i \\
\dot{\mathbf{q}} &= \frac{1}{2}\dot{\mathbf{\omega}} \otimes \mathbf{q} \\
M\dot{\mathbf{\omega}} &= -2k_o\mathbf{n} \times \mathbf{\omega} - \mathbf{\omega} \times M\mathbf{\omega} + \tau_i
\end{align*}
\]

where \( \mathbf{p} \) is the displacement vector of the wrist from the commanded position, \( \mathbf{v} \) is the velocity vector, \( \mathbf{\omega} = (0, \mathbf{\omega}) \), “\( \otimes \)” and “\( \times \)” indicate quaternion multiplication and cross product, respectively.

We chose the wrist frame for numerical integration. Since we have assumed that the principal moments of inertia are aligned with this frame, matrix \( M \) becomes diagonal in (10). To insure the unity constraint on the quaternion, it is normalized after the numerical integration.

Assuming we have an ideal reference tracking, we equate the reference signals with the position of the calculated compliant frame above.

3.2 Haptic feedback

On the master side, the haptic feedback to the operator given in the master task frame, \( \mathbf{F}_{\text{fb}}^{MTF} \), can be designed as follows. We denote the vector of forces at the TCP point given in the slave task frame by \( \mathbf{F}_i^{STF} \) and the translational...
velocity of the slave frame with respect to the slave task frame by \( v_s \). Note that these variables appear delayed.

\[
F_{fb}^{MTF} = \gamma(D_2 - D_1)\dot{v}_s - D_2v_m + \alpha F_i^{STF},
\]

where \( v_m \) is the translation velocity of the master frame with respect to the master task frame; \( \alpha \) represents a force scaling factor to adjust direct haptic feedback to the operator; \( D_1 \) and \( D_2 \) correspond to the free-motion damping and in-contact damping, respectively. When there is no contact between the slave and the environment and the delay is not significant, \( F_i^{STF} \) and \( v_m - \gamma v_s \) are almost equal to zero so that the operator perceives only some damping force corresponding to \( D_1 \). On the other hand in contact, \( v_s \approx 0 \) and the operator perceives a scaled force with the other damping coefficient, \( D_2 \). Two damping factors were deployed to enhance the contact detection and stability.

Let \( R_{11} \) be the rotation of \( T_{11} \) transformation. The force feedback in the master base frame can be calculated as below.

\[
F_{fb} = R_{11}F^{MTF}.
\]

4. EXPERIMENTAL SETUP

Our setup consisted of a Force Dimension Omega-7 device which is a 7-DOF haptic interface with closed loop stiffness of 14.5 (N/mm), see Force Dimension (2013). We used the extra DOF of the device (the extent of the openness of the gripper) to adjust admittance remotely. Moreover, an industrial robot, ABB IRB 140, a wrist mounted six axis force and torque transducers with internal electronics, JR3 100M40A were used. The controller was implemented in Simulink® and using the Real-time workshop the code was compiled for ExtCtrl, Blomdell et al. (2010). The final code runs at 250 Hz on a Fedora Xenomai machine connected via the ExtCtrl interface to the robot.

See Fig. 4 for the setup. Table 1 represents the nominal parameters of the slave controller.

<table>
<thead>
<tr>
<th>( m )</th>
<th>( k_i )</th>
<th>( D_i )</th>
<th>( M_{xx} )</th>
<th>( M_{yy} )</th>
<th>( M_{zz} )</th>
<th>( k_a )</th>
<th>( D_a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>10</td>
<td>0.4</td>
<td>0.2669</td>
<td>0.2669</td>
<td>0.0538</td>
<td>18</td>
<td>5</td>
</tr>
</tbody>
</table>

30

Fig. 4. Peg-in-hole setup: (Front) Master side with Omega-7 haptic interface. (Back) IRB140 robot with metal pipe held in the gripper.

Fig. 5. Pushing a solid block. The trajectory is in blue, force vectors are in red, and torque vectors are in green.

Fig. 6. Pushing a solid block. Comparison of master (blue) and slave (red) trajectories.

Fig. 7. Pushing a solid block. Comparison of master (blue) and slave (red) orientations.

30

Fig. 8. Pushing a solid block. Interaction force and torque.

5. RESULTS

Three different scenarios were tested. The operator sequentially touched three sides of a solid block in the first experiment. The second experiment was a peg-in-hole operation. The third one involved a more challenging task of snap-fit.

The pushing task was meant to demonstrate the compliant behavior of the slave robot. The operator starts from the right hand side of the block and pushes the block. Then
6. DISCUSSION

6.1 Stability

Since bilateral teleoperation system is to couple two dynamical systems (master and slave), it leads potentially to instability. The analysis and synthesis is made difficult by several factors including: transmission time delays between master and slave, uncertain dynamics of operator and environment (Hokayem and Spong, 2006). The use of a industrial robot as a slave device may increase these factors in terms of control bandwidth and nonlinear dynamics.

Although the given experimental setup provides relatively high update rate for an industrial robot, it is less than the recommended update rate of 1 kHz for haptic interfaces (Burdea and Brooks, 1996). Therefore, it may cause an additional delay in tracking reference trajectories on the slave. There is also a time delay over the Ethernet network between master and slave. There are some solutions to deal with the network delays, see (Kristalny and Cho, 2012) and references therein. However, they are not practical for the internal delays.

As a remedy, we employed force scaling factor $\alpha$ on the haptic feedback to the operator. The scaling factor contributes to the closed-loop stability by decreasing the overall loop gain (motion of master $\rightarrow$ motion of slave $\rightarrow$ environment’s reaction $\rightarrow$ force feedback). Although
a lower $\alpha$ may disturb the perception of the environment, it provides closed-loop stability under several uncertainties discussed above. To achieve higher $\alpha$, the control bandwidth and delays have to be addressed in a robust control framework.

6.2 Autonomy in operation

As discussed earlier, it is not always possible to build a transparent teleoperation system. This problem is more prominent if we do not have a 4-channel teleoperation system. The master device used in our experiment provides only the measurement of the position and not the forces. Additionally, it is not possible to feedback torques to the operator. To mitigate these problems, we have implemented an admittance control. By choosing a suitable wrist frame and adjusting the admittance parameters, $\Delta T$ can in practice fill the gaps. For example, in the snap-fit task the indirect force control strategy helped the switch glide into its final position without demanding an accurate operation from the operator. Hence, the admittance control provides a loose coupling between the master and the slave. From this perspective, we have equipped the robot with a certain degree of autonomy.

This could be viewed as cooperation between the operator and the slave robot. Another possible cooperative scenario is enforcing pure kinematic constraints on the motion of the slave robot, which can relax the operator from taking them directly into account.

For comparison, we performed the same tasks while either turning off the force controller in the robot or the force feedback to the operator. Through a series of the experiments, we realized that tasks could be performed in shorter time and with lower values of interaction forces in the cooperative scenario compared to the others.

7. CONCLUSION

To generate trajectories for assembly tasks, the detection of contacts by the operator plays an important role. The task can be simplified by providing force feedback to the operator. In this work, we have introduced the notion of slave-assisted teleoperation. The active compliance on the slave side together with tele-admittance strategy gives the operator more freedom to manipulate objects with lower risk of damaging a workpiece.

Moreover, we have introduced a structured way to define required coordinate frames for teleoperation. This facilitates a quick setup of multi-degrees of freedom teleoperation and customizing it according to the preference of an operator.

Thanks to these features and the quaternion implementation, we were able to build a singularity-free 6-DOF teleoperation system for an industrial robot. The developed setup aims for collecting both position and force data for assembly tasks. The data can be processed for learning purposes. The force/torque provides valuable information for triggering segmentation and defining tolerances.

REFERENCES


