Advanced Usability Through Constrained Multi Modal Interactive Strategies: The CookieBot

Gabriele Bolano[1], Pascal Becker[1], Jacques Kaiser[1], Arne Roennau[1], Ruediger Dillmann[1]

***Abstract*— Service robots are becoming able to perform a variety of tasks and they are currently used for many different applications. For this reason people with different backgrounds and also without robotic experience need to interact with them. Enabling the user to control the movement of the robot end-effector, it is important to provide an easy and intuitive interface. In this work we propose an intuitive method for the control of the robot TCP position and orientation. This is done taking into account the robot kinematics in order to avoid dangerous configuration and defining rotational constraints. The user is enabled to interact with the robot and control its end-effector using a set of objects tracked by a camera system. The autonomy level of the robot changes depending on the different phases of the interaction for a better efficiency. An intuitive GUI has been developed to ease interaction and help the user to achieve a better precision in the control. This is possible also through the scaling of the tracked motion, which is represented as visual feedback. We tested the system through multiple experiments that took into account how people with no experience interact with the robot and the precision of the method.**

Fig. 1: The user controls the robot TCP through the use of a tracked object. A graphical user interface guides the user in the interaction, enabling also a better precision in the control.

I. Introduction

In the last years, robots took over the repetitive tasks in factories and are now entering the consumer market. Service robots are nowadays used in industrial scenarios, domestic environments and as first contact point to interact with organizations customers [1], [2]. Because of this spread, people with very different backgrounds and even without any robotics experience need to be able to interact with these robots. Many systems are completely autonomous and the user does not need to interact with them. For example cleaning robots just need to perform their task autonomously and they have just to avoid collisions with humans and obstacles changing the direction of their motion based on sensor data.

For other tasks the user needs to teleoperate the robot in order to control its TCP while using a specific tool [3], [4]. Teleoperation is relevant for tasks that require flexibility in the definition of the robot path. Importantly, an intuitive user interface to make this control interaction flexible and precise should be provided [5], [6]. Such systems are used for robotic-assisted surgery, in order to scale bigger human movements to more precise robot positions on the patient [7]. Hybrid approaches consist of alternating between autonomous and constrained teleoperation phases. Some applications might require the selection of different levels of autonomy [8], [3], [9]. For example the robot might need to take a tool from a set and the user interaction would be more efficient if limited to the task selection instead of the actual control of the robot to grasp the desired tool. Once the task is selected and the robot autonomously gets the needed item, the autonomy level can be set again to teleoperation control to enable the user to directly control the TCP to arbitrary positions.

Since service robots are mainly used by people with no robotic experience, it is important to define positional and rotational constraints. This is crucial to avoid risks and undesired collisions when the user is not fully confident with the system. However, this limitations should still allow the user to have a flexible control on the position of the robot tool including rotations. To develop an intuitive control method it is also important to provide useful information to guide the user during the different phases of the interaction and give him a visual feedback about the robot behavior [10].

In this paper we propose a system to control a robot in an intuitive way, with the use of different autonomy levels. The user starts by taking a tool from a set of control tools tracked by cameras. This triggers the robot to autonomously get the corresponding tool and move to the control area. At this point, the user can teleoperate the 3D position of the robot TCP by moving the tracked tool. The user also controls the rotation of the TCP through the orientation of the tracked tool in a safe manner. In our approach, the inclination angle of the tracked tool is mapped into the rotation of the last joint of the robotic manipulator. In this way, we avoid dangerous reconfiguration of the robot kinematic due

[1]The authors are with FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany {bolano, pbecker, jkaiser, roennau, dillmann}@fzi.de

to arbitrary rotations, allowing a flexible and safe control of the TCP. To ease the interaction and make the unexperienced user able to quickly understand how to control the robot and its behavior, a GUI has been developed. This helps the user in the different phases of the interaction, enabling him also to achieve a better precision in the control and have a feedback on the robot motion. We implemented the components needed and the communication between them using the open-source framework ROS [11]. In order to test the system with unexperienced users, our approach was developed for an application in which a service robot had to decorate cookies through the interaction with the user. Anyway, our approach is modular and generic and can be deployed for general applications and with arbitrary robotic arms.

The structure of this paper is as follows. In Section II, we present the related work. In Section III, we describe our interaction approach to control a robot TCP position and orientation. In Section IV, we describe an application scenario developed and an additional experiment used to evaluate the system. Finally, we provide conclusions and perspectives in Section V.

II. Related Work

Service robots are becoming able to execute complex tasks and are being used for many different applications [12], [1]. They are usually programmed to execute a given task autonomously or to help the user providing information and guiding him. For example cleaning robots are used in domestic environments to clean the floor and their interaction with the user is limited in avoiding possible collisions. Other service robots are deployed for tasks in which the interaction with the human is the most important aspect. Tribel et al. propose a service robot for guiding passengers and help them in busy airports [1]. The work focuses on making the robot aware of the human social behavior, recognizing the people activities and relations.

For most of service robots, the interaction with the human is limited in the selection of the desired task, which is then performed autonomously by the robot. Many works focus on the interaction with robot in order to communicate the desired behavior through the use of GUIs, speech or gestures [13], [14]. Li et al. propose a system to communicate intentions with a non-verbal communication based on the user's eye gaze [15]. Their approach is used to infer the intention of the user in order to trigger an assistive robot to provide proper service. For other tasks the user might need to adapt the robot's motion or teleoperate its TCP in order to reach custom positions. This problem has been addressed in the work from Muszynski et al. [3]. The authors studied how to control personal service robots through the use of an hand-held computer interface. The user is enabled to switch between different autonomy levels for the robot: skill control, body control and task control. The results of the experiments showed that the autonomy levels configuration increased the efficiency, while the situational awareness given by the camera view could be improved. The selection of different autonomy levels is also studied in the work of Baker et al. [16]. The authors suggest a system that allows a robot to range from fully autonomous to teleoperated. The method is designed for an urban search and rescue scenario.

Teleoperation has the advantage to make the user able to move the robot at a safe distance with precision. The research has investigated how to make this control as intuitive and easy as possible for unexperienced users [17]. Since humans are using gestures to communicate between them and show their intentions, this is an interesting communication channel that has been studied in many works [18], [19]. Some approaches consist of detecting the user's gestures in order to trigger predefined robot motions, as for example moving in one direction [20]. Wolf et al. propose a system in which a wearable device is used to capture electromyography and inertial signals [21]. These are used to send commands to the robot in order to trigger autonomous capabilities. In this way the system gains in safety, not relying entirely on human's input. If the robot task involves the use of a tool, the tracking of an hand-held device can help the user to achieve a better precision in the control. Fischer et al. proposes a study, which consists in a comparison between the use of a control object and a data glove [22]. The results of the experiments conducted by the authors show that the control object led to fewer errors. The scaling of the teleoperated motion is a method that allows the user to have a better precision in the control, for example for surgical applications [7]. In this way, bigger user movements can be mapped into smaller robot motions, enabling the robot to have a more steady position and allowing small errors in the control performed by the human.

In a previous work we studied how to control the 3D position of a robotic arm TCP in an intuitive and flexible way, without the need of wearable devices or a set of predefined gestures [5]. The user's hand position relate to a reference point was used in order to send velocity commands to the robot. A camera was used to track the hand position and a GUI displayed on a screen was used to provide him a visual feedback about the control and the commands sent to the robotic arm. The drawback of these methods is that they usually provide just the control of the position of the robot end-effector, without any degree of freedom on the orientation. This is usually because the kinematics of the robot can be a problem for arbitrary positions and rotations. This can led the robot to being not able to reach the desired position from its current configuration or forcing him to reconfigure its joints. This could be dangerous and cause undesired collisions.

GUIs are helpful tools in order to make the user understand how to interact properly with the robot and understand its behavior through the use of visual feedback [10]. They provide him support to achieve the intended behavior without the need of previous experience with the system.

In this work we propose an intuitive method to interact and control a service robot with different levels of autonomy. In contrast to other methods which allow only the control of the position, in the system proposed the user is able to

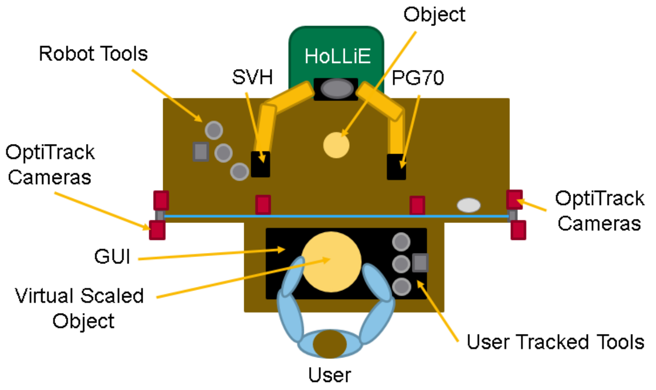

Fig. 2: The overall hardware setup used for the interaction between HoLLiE and the user.

intuitively teleoperate the robot TCP position and orientation using a set of control tools, which are tracked by a camera system. The interaction with these tools, enables the robot to grasp or place the corresponding one in order to safely go towards the control area. At this point the user is enabled to teleoperate the TCP motion with a scaled control, that enables him to achieve a better precision. Our focus is also on the flexibility and safety of the control, since in literature there is a lack of methods that allows the user to control the position and rotation of the robot TCP avoiding kinematics reconfigurations. In order to enable the unexperienced user to learn immediately how to interact with the robot without any teaching phase, we developed a GUI that supports him in the different phases of the interaction. The feedback about the robot behavior and the tracking of the tools enable him to feel more comfortable during the control and achieve a better precision.

III. Approach

For the interaction control developed we used our service robot HoLLiE, which has two PILZ PRBT robotic arms with 6-DOF. For the interaction system developed, we focuses on the control of one arm, which is equipped with a Schunk SVH 5-finger hand. The setup included the marker based tracking system OptiTrack and a monitor to display the user interface. The overall hardware setup is depicted in Fig. 2. Fig. 3 represents the system software architecture and the communication between the components. We used the behavior framework FlexBE [23] to define the high level logic of the interaction and the handling of the events coming from the other modules. The control of the robot and hand is done by alternating between a joint-position controller and a non-compliant version of the Cartesian controller described in [24], implemented with ROS control [25]. The joint-position controller is used to autonomously reach, grasp and dispose tools with fixed trajectories defined with our Motion Pipeline. The Cartesian controller is used to teleoperate the robot in a constrained manner.

To enable the control of different tools, we provided the user with a set of control tools, which positions and orientations are tracked all the time. The use of Optitrack allows to track the 6-DOF pose of the needed objects with sub-millimeter accuracy and low latency. We used a set of six cameras to monitor the user's workspace and ensure the tracking of the tools for arbitrary positions of those within this area. Each of these objects has a set of reflective markers that are used by the tracking system. Once the user grasps or releases one of them, the robot is triggered to execute the same action on the real one, which is positioned in a fixed position in its workspace. In order to identify the different tools, we used different patterns for each one. Their home positions are marked on the GUI and once the user takes one of them, the execution of the corresponding robot's trajectory is triggered. Fig. 4 shows a set of tools, each one with a different pattern of markers. Both flat and 3D markers are used, depending on the shape of the object. This is done to ensure the tracking of the objects while they are grasped and manipulated by the user. The Tools Manager component keeps track of the state of the tools, updated accordingly to their distance from their home positions. A specific tool is activated when its distance to its home position is above a threshold value. In our system only one tool can be active at a time and the feedback on the display confirms the user which one it is. The Tools Manager also communicates with the scaling component, in order to provide the current active tool id to compute the robot target position. As a feedback to the user, the GUI highlights which object has been taken and reports the information that the robot is executing the trajectory to reach and grasp it.

After the grasping of the desired tool, the robot moves to the control area and the GUI notifies the user when he can start to control the robot TCP. In Fig. 6 is represented the situation in which the user has selected one tool and the GUI notifies him to wait for the robot to get the corresponding one. It also displays as feedback the current active tool and its placing position in order to finish the interaction or switch to another tool. The GUI visualizes a virtual representation of the control area in order to give the user a better understanding of the robot workspace limits. If the position of the tracked tool is outside of the control boundaries, the closest valid position is send as target goal. This allows the user to commit errors during the control, avoiding unwanted robot configurations and possible collisions with the surrounding environment. To control the robot with the Cartesian poses computed from the tracking system, we use

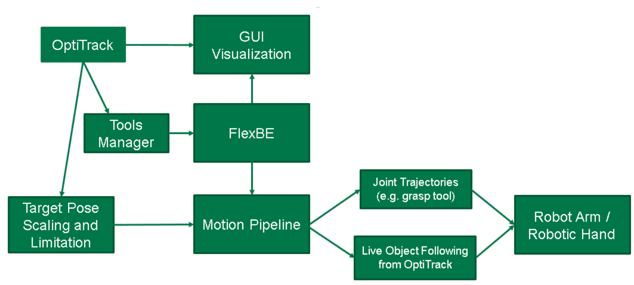

Fig. 3: Overview of the system software architecture and communication between the components.

Fig. 4: A set of control tools which can be used by the user to interact with the robot. Each one of them is identified and tracked by the Optitrack system through the use of different patterns of markers.

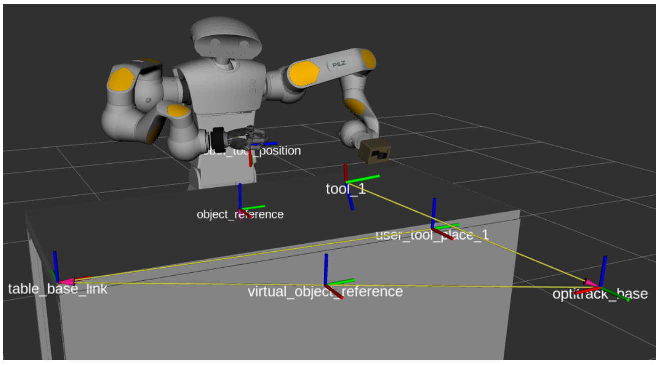

Fig. 5: The reference frames used to transform the position of the tracked tool into the desired target of the robot. The information from the tracking system is converted in the reference frame of the virtual object, which is then scaled and transformed into the reference frame of the real object in the robot workspace.

a non-compliant version of the Cartesian controller proposed in [24]. This controller based on forward dynamics trades-off precision to avoid singularity and sudden reconfigurations. The parameters of this controller allow to set a more reactive or smooth control depending on the application. The position of the tool captured by the tracking system is converted into the reference frame of the virtual object displayed on the GUI. Fig. 5 shows the reference frames that are used to transform the tracked pose into the desired robot target. In this way the user can move the tool above that and this pose is then transformed into the real object reference frame. This position is the one used to send the target goal to the robot controller. Before that, the scaling is applied in a coherent way with the virtual representation on the GUI. The scaling of the tracked movements allows the user to have a better precision in the robot control on a small volume.

In order to allow the flipping of the robot tool, we implemented an approach to control the rotation of the robot TCP based on the orientation of the tracked tool. The angle that is relevant for the flip of it is the one between its axis and the gravity axis. To avoid the robot to reconfigure completely its kinematics and to make it ables to reach all the desired positions without dangerous configurations, this angle is mapped into the rotation of the last joint of the robot arm. In this way, the kinematic configuration of the robot remains unchanged and only the rotation of the tool is controlled. In Fig. 8 is represented the approach that allows to map the desired angle to the last robot joint rotation. The use of fixed trajectories to reach the control area has the big advantage to provide a safe joint configuration at each start of the Cartesian control. This prevents sudden configuration changes.

To handle the different phases of the interaction, we deployed a state machine using FlexBE [23], which is an open source high-level behavior engine. In this way the various events from the tracking system trigger the state machine to execute the corresponding predefined joint trajectory. Once this execution is finished, the state machine is responsible for the switch to the Cartesian controller in order to enable the teleoperation mode. The placing of the tool back to its home position triggers then the switch to the joint trajectory controller in order to make the robot place the tool back.

IV. Evaluation

The system has been used on our service robot HoLLiE to decorate cookies with four different toppings. The robot had two different types of tools to use. Three canisters containing sugar decorations as sprinkles, stars and marshmallows and a sugar dispenser. The canisters needed to be flipped in order to make the toppings coming out and the sugar dispenser was enabled through the use of a valve. This latter was activated or deactivated based on a height threshold of the tool from the table surface. The user had a corresponding set of control tools and he could decide which one to use just taking one from its position marked on the GUI. This was displayed on a monitor mounted on top of a table. After waiting for the robot to get the selected tool, the user was able to control the TCP with the method described in the previous section. A virtual representation of the different toppings was displayed in order to give a feedback to the user about the activation of the tool during the control. For example, the sugar gun was enabled based on the distance of the tracked tool from the monitor displaying the GUI. In Fig. 9 is represented the simulation of the sugar flow, that was used to give the user a feedback about the activation of the tool and to help him to define the desired path with more precision. We used the right arm of HoLLie to grasp the tools and decorate the

Fig. 6: The GUI notifies the user that the robot is getting the tool that has been selected. In this way he can understand the robot behavior and detect when the teleoperation mode in enabled to directly control it.

Fig. 7: During the teleoperation mode, a scaled virtual representation of the object to be worked is displayed. This is useful to understand the control boundaries and to allow a better precision with the scaling of the motion. The GUI highlights also the current active tool and its placing position in order to end the interaction or switch to another one.

Fig. 9: The GUI shows a simulation of the sugar flow that is dispensed through the gun. This is useful to give the user a feedback about the activation of the tool and help him in the definition of the desired path.

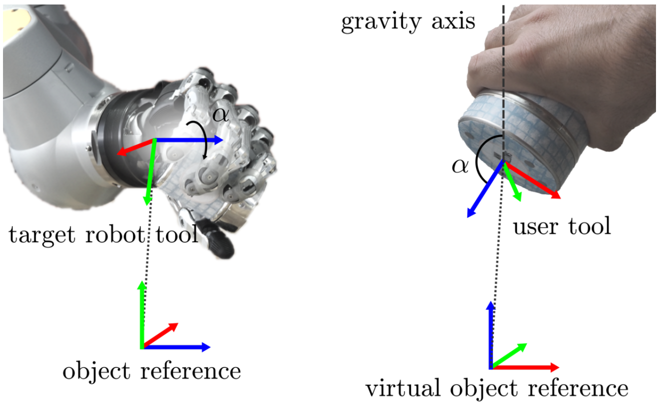

Fig. 8: Calculating the target robot tool pose with respect to the tracked user tool pose. The translation of the target robot tool is the same as user tool in their respective reference frame, as depicted with the dotted lines. However, the target robot tool only rotates with respect to a single predefined axis corresponding to the last joint of the PILZ robotic arm. This facilitates the task of the cartesian controller which only needs to control the last joint to decrease orientation errors. We set the angle of this rotation α to be the angle formed by the user tool and gravity axis.

Fig. 10: A user without previous experience with the system uses the sugar gun to decorate the cookie and the GUI to have a feedback about the tool activation.

cookies, while the left arm was equipped with a Schunk PG70 gripper in order to place the cookies in the control area and deliver them to the user through the use of a spatula. The control of the two arms was completely decoupled with the use of our Motion Pipeline and FlexBE. In this way the cookies could be handed to the users with the left arm, while the right one was placing back the current tool, with a consequent decrease of waiting time.

A. *Real-world user evaluation*

The system was used at the Stallwächter Party 2019 in Berlin, where more than two hundreds people with no experience with the system and with no technical background have tested the application developed. All the users understood quickly how to use the system without instructions. They interacted properly with the robot in order to get the desired tool. They were also able to control the position of the robot TCP to decorate the cookies using all the tools available, drawing letters and arbitrary shapes on the cookie surface. Fig. 10 shows a user decorating a cookie with the sugar gun tool and using the GUI to have a visual feedback about the tool activation. In this application, the rotation control was also important for some of the tools, in order to make the the toppings coming out. The users at the event understood easily how to get the desired rotation of the TCP acting on the tracked tool rotation.

B. *Performance evaluation*

To further evaluated the precision and the motion latency of the system developed, we added a robot to simulate the user movements. We used an UR5 mounted on a table coupled with a tracked tool attached to the end-effector. This additional robot was used to repeatedly execute a predefined motion between two points in the workspace tracked by the cameras. At the lower position we added a button that we used to record the time in which the position is reached. A similar button was placed in the HoLLiE workspace in order to record the time in which the teleoperated robot reached the same position. In this way we measured the motion latency between the two events. In Fig. 11 the setup used for the experiment is reported. The evaluation has been done using different velocity settings for the UR5 motion.

The motion latency between the movement of the tracked tool and the reaching of the goal position for the teleoperated robot is caused by many factors including the delay introduced by tracking system, the computation of transformations, the Cartesian controller and the communication between the components. The charts in Fig. 12 and 13 report the motion latency of the system expressed in milliseconds. The results are plotted for different velocities settings of the UR5 and for 50 evaluation iterations. Using the maximum speed, which is 1500 mm/s, the motion latency caused the teleoperated robot to not be able to reach the target position in order to press the button. We faced the same issue also for some attempts with the speed set to 75%. The chart in Fig. 14 represents the numbers of failed attempts for each robot speed.

The results of the experiments showed that the system presents a motion latency, which is due to the various components and their communication. The Cartesian controller adapts to this by not reaching all the positions attended by the tracked objects. However this showed to be quite negligible and not proving issue in the reach of the target position, except for high speeds.

V. Conclusions

The system developed enabled an intuitive and flexible human interaction with a service robot. The application developed showed that our control system is easy to use for people with no robotics experience. The use of different autonomy levels allowed the human to control different tools with the robot in a flexible way and without the need to worry about the motion to grasp and switch them. The users that tested the system were able to precisely control a complex robot TCP position and rotation without the need of a teaching phase. The definition of constraints enabled the robot to avoid collisions with nearby object. The mapping of the tracked tool orientation to the rotation of the last joint of the robot arm, allowed a flexible use of the tools, avoiding dangerous reconfiguration that could occur allowing general 3D rotations. The use of an intuitive GUI helped the users to understand how to interact with the robot and its behavior. In this way the interaction was more comfortable and the users had a better feeling understanding what the robot was doing. The GUI allowed also them to have a better precision in the control, representing a virtual and scaled version of the object to work on. In this way bigger movement were mapped to smaller ones in the robot workspace.

The evaluation of the system precision and motion latency conducted with an additional robot showed that our system does not introduce a big latency in order to reach the target pose. High speeds caused the robot to not reach the goal position with precision and this is because of the different components involved and their communication. For example, the reduction of latency of the tracking system could reduce

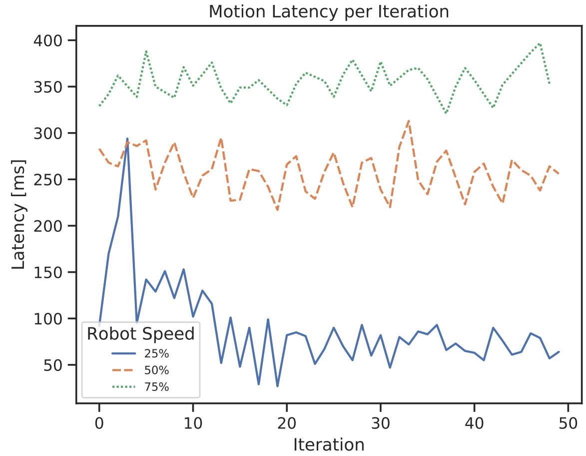

Fig. 12: Motion latency between the movement of the tracked tool and the reaching of the target position with the teleoperated robot. The values are expressed in milliseconds and are reported for each iteration.

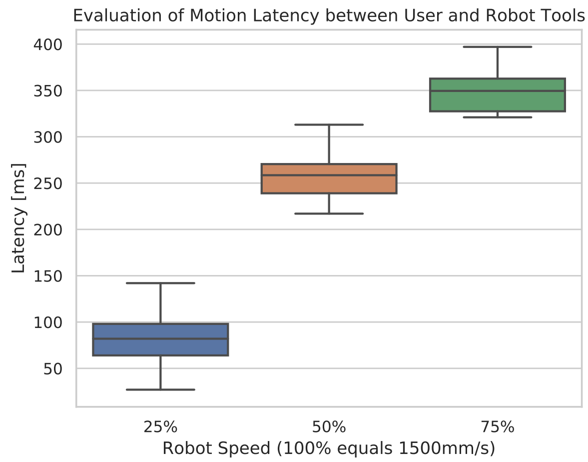

Fig. 13: Motion latency between the movement of the tracked tool and the reaching of the target position with the teleoperated robot. The values are expressed in milliseconds and are reported for different UR5 speeds.

Fig. 11: The setup used to evaluate the precision and motion latency of the system. A UR5 was used to move the tracked tool between different positions. The time needed to reach the different positions was measured with two buttons placed in the robots workspace.

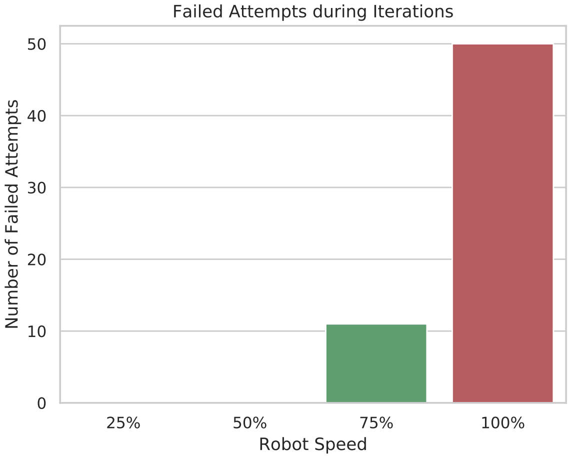

Fig. 14: Failed attempts to reach the target position tracked from the UR5 motion. The results are reported for different UR5 speeds.

this problem.

The overall setup could be improved using a marker-free tracking system in order to avoid the use of external markers to have a good tracking of the needed objects. The control of the TCP rotation could be enhanced in order to allow a safe 6 DOF control which allows the user to have a more flexible control.

REFERENCES

[1] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O. A. I. Ramírez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A. J. Lilienthal, T. Linder, M. Lohse, M. Magnusson, B. Okal, L. Palmieri, U. Rafi, M. van Rooij, and L. Zhang, *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*. Cham: Springer International Publishing, 2016, pp. 607–622.

[2] M. Mast, M. Burmester, B. Graf, F. Weisshardt, G. Arbeiter, M. Španěl, Z. Materna, P. Smrž, and G. Kronreif, *Design of the Human-Robot Interaction for a Semi-Autonomous Service Robot to Assist Elderly People*. Cham: Springer International Publishing, 2015, pp. 15–29.

[3] S. Muszynski and S. Behnke, “Adjustable Autonomy for Mobile Teleoperation of Personal Service Robots,” 2012.

[4] M. M. Coad, A. M. Okamura, S. Wren, Y. Mintz, T. S. Lendvay, A. M. Jarc, and I. Nisky, “Training in Divergent and Convergent Force Fields During 6 - DOF Teleoperat ion with a Robot - Assisted Surgical System,” *2017 IEEE World Haptics Conference (WHC)*, no. June, pp. 195–200, 2017.

[5] G. Bolano, A. Tanev, L. Steffen, A. Roennau, and R. Dillmann, “Towards a Vision-Based Concept for Gesture Control of a Robot Providing Visual Feedback,” *to appear*.

[6] H. Yamada, N. Tao, and Z. DingXuan, “Construction tele-robot system with virtual reality,” in *2008 IEEE International Conference on Robotics, Automation and Mechatronics, RAM 2008*, vol. 00, 2008, pp. 36–40.

[7] F. Richter, R. K. O. Member, and M. C. Y. Member, “Motion Scaling Solutions for Improved Performance in High Delay Surgical Teleoperation.”

[8] J. J. Laviola, U. Engelke, and D. R. Fernández, “Natural User Interfaces for Adjustable Autonomy in Robot Control,” *IEEE Computer Graphics and Applications*, vol. 35, no. June 2014, pp. 20–21, 2015.

[9] S. A. Mostafa, M. S. Ahmad, and U. T. Mara, “A Flexible Human-Agent Interaction Model for Supervised Autonomous Systems,” *2016 2nd International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR)*, no. August, pp. 106–111, 2016.

[10] G. Bolano, A. Roennau, and R. Dillmann, “Transparent Robot Behavior by Adding Intuitive Visual and Acoustic Feedback to Motion Replanning,” in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018, pp. 1075–1080.

[11] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.

[12] J. Pripfl, T. Körtner, D. Batko-klein, D. Hebesberger, M. Weninger, C. Gisinger, A. Weiss, M. Bajones, and M. Vincze, “Results of a Real World Trial with a Mobile Social Service Robot for Older Adults,” pp. 497–498, 2016.

[13] R. Stiefelhagen, C. Fugen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, “Natural Human-Robot Interaction using Speech , Head Pose and Gestures,” pp. 2422–2427, 2004.

[14] T. Watanabe, “InterRobot: A Speech Driven Embodied Interaction Robot,” pp. 322–327, 2000.

[15] S. Li and X. Zhang, “Implicit Intention Communication in Human – Robot Interaction Through Visual Behavior Studies,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 437–448, 2017.

[16] M. Baker, O. Hall, H. A. Yanco, and O. Hall, “Autonomy Mode Suggestions for Improving Human- Robot Interaction *,” pp. 2948–2953, 2004.

[17] L. Peppoloni, F. Brizzi, C. A. Avizzano, and E. Ruffaldi, “Immersive ROS-integrated Framework for Robot Teleoperation,” *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 177–178.

[18] K. Qian, J. Niu, and H. Yang, “Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect,” in *International Journal of Smart Home*, vol. 7, no. 4, 2013, pp. 203–208.

[19] V. S. Rao, C. Engineering, and D. Malsburg, “Gesture Based Robot Control,” in *2006 Fourth International Conference on Intelligent Sensing and Information Processing*, 2006, pp. 0–3.

[20] T. L. Baldi, G. Spagnoletti, M. Dragusanu, and D. Prattichizzo, “Design of a Wearable Interface for Lightweight Robotic Arm for People with Mobility Impairments,” in *2017 International Conference on Rehabilitation Robotics (ICORR)*, 2017.

[21] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, “Gesture-Based Robot Control with Variable Autonomy from the JPL BioSleeve,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1160–1165.

[22] K. Fischer and F. Kirstein, “Teleoperation for Learning by Demonstration: Data Glove versus Object Manipulation for Intuitive Robot Control,” no. 269959, 2014, pp. 346–351.

[23] P. Schillinger, S. Kohlbrecher, and O. von Stryk, “Human-Robot Collaborative High-Level Control with an Application to Rescue Robotics,” in *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016.

[24] S. Scherzinger, A. Roennau, and R. Dillmann, “Forward dynamics compliance control (fdcc): A new approach to cartesian compliance for robotic manipulators,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4568–4575.

[25] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernández Perdomo, “ros_control: A generic and simple control framework for ros,” *The Journal of Open Source Software*, 2017.