Haptic Interface for Domestic Service Robot

Rhama Dwiputra and Gerhard K. Kraetzschmar
Department of Computer Science, Bonn-Rhein-Sieg University of Applied Science, Germany
Email: {rhama.dwiputra, gerhard.kraetzschmar}@h-brs.de

***Abstract*—Domestic service robots are designed for human environment. Therefore, the robot should be controlled by means of natural interactions rather than the common controllers used in laboratory setting (e.g. keyboard or joypad). In this paper, a feature for controlling a domestic service robot through physical interaction is presented. The feature showcases the utilization of the robot's manipulator as its haptic interface. The feature uses low pass filter and proportional-integral- derivative (PID) controller which remove the rapid fluctuation in the force input and stabilize the velocity output. The robot's omnidirectional capability is accommodated through different interaction modes which can be selected based on the user preference. Through the proposed approach, the physical interaction will be translated into base motion commands which adjust itself autonomously. The result is a more natural way of controlling the robot. The feature has been proven to be intuitive and safe through the user trial which was performed in a domestic environment.**

***Index Terms*—domestic service robot, haptic interface, human-robot interaction, light-weight manipulator.**

I. Introduction

Domestic service robots are designed to assist people by performing household chores in a human environment. For robots in such setting, interaction ability is one of the basic autonomous capabilities [1]. Intuitive communication with the user is one of the main skills require for performing domestic service tasks from autonomous robot [2]. One of the communication mediums between human and robot is through physical interaction. In this paper, a feature for controlling the robot through physical interaction is presented. The feature showcases the utilization of a robot manipulator as its haptic interface. Haptic interface is a device that enables person-machine communication through touch, and most commonly, in response to user movements [3]. A computer mouse and the touch screen of a smart phone are examples of haptic interfaces which are commonly used on a daily basis.

The feature translates physical interaction between a user and the robot's manipulator into base motion commands. The interaction in the feature was designed to be intuitive so that it can be used without any additional training. The proposed approach uses a low pass filter and a proportional- integral-derivative (PID) controller to achieve a stable motion of the robot base which adjusts itself based on the user interaction with the robot's manipulator. The result is a human-like interaction when controlling the robot as shown in Fig. 1. The usability of the feature was evaluated through user trial in a domestic environment. The user was asked to use the interface in two different settings. In the first setting, the user was requested to maneuver the robot as close as possible along a predefined path. The second setting required the user to control the robot, from start point to goal point, using any desired path. In the user trial, the path deviation was analyzed and user feedback was collected for the feature update.

Figure 1. Motion control through physical interaction.

This paper is organized as follows. After this introductory section, the related work will be presented in section II. Section III presents the implementation of the feature. Section IV describes the user trial and its analysis. Finally, section V summarizes the work and proposes possible topics for future work.

II. Related Work

The core of haptic technology is communication through physical interaction. Most research on robot-related haptic technology focuses on object recognition and grasping [4], [5]. In such research, the information gained through tactile sensing (pressure and texture) improve the grasping process and object handling. The work presented here focuses on haptic interface in human robot interaction.

Haptic features use physical interaction as its input. Classification of physical interaction has been investigated in [6], [7] and [8]. [6] classifies *slap*, *pat* and *scratch* by the contact area and the absolute pressure value. For classifying sensitive touching behaviors, *stroke* and *tickle*, [6] uses the temporal difference in the pressure pattern and

Manuscript received September 10, 2013; revised November 8, 2013.

doi: 10.12720/joace.2.3.282-288

contact area. [7] develops automatic categorization of physical interaction through an agglomerative hierarchical clustering method. The results from [7] show that the interactions were classified first (upper part of hierarchy) based on the region of touching and secondly (lower part of hierarchy) based on the manner of touching. [8] develops a touch recognition system through multi-windowing feature extraction and temporal decision tree classifiers. In [8], touch patterns *hit*, *beat*, *rub*, and *push* were distinguished based on force, contact time, repetition, and the change in contact area.

Recognizing different types of physical interaction allows the robot to behave appropriately in response to each specific interaction. [9] and [10] develop features where a robot can senses a human's touch and move its head towards the human direction. [9] uses a robot suit which consists of 192 sensing regions and the robot's gaze direction was determined through a direct mapping of the sensed region. [10] uses the tactile sensor elements embedded in a soft skin that covers the robot's entire body. To determine the robot's gaze direction, [10] estimates the human's position and posture using probabilistic distribution of the physical interaction. [11] and [12] develop features where the robot movement is being controlled through haptic interaction. [11] develops a robot dance partner which adjusts its movement based on the step information of the dance and the force applied by the human partner. [12] uses the contact trajectory from a haptic armor to command different movement such as *stop*, *forward* and *turn*.

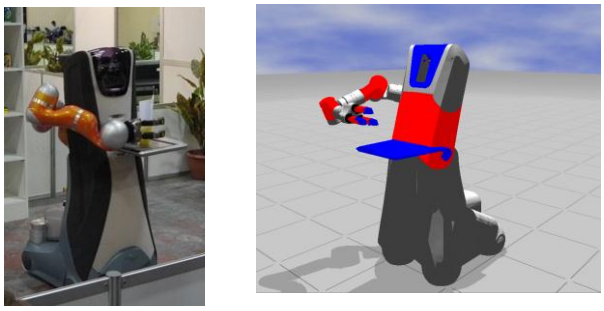
The feature presented in this paper translates physical interaction into base movement similar to those presented in [11] and [12]. Assigning different base motion command to different class of interaction such as presented in [6], [7], and [8] is not suitable since it requires the user to learn the interaction characteristic (pressure level, contact area, contact duration, orientation). Therefore, the mapping from interaction to base command for the feature was designed to be straightforward and intuitive (section III). Rather than using an additional device, the robot's manipulator is being utilized as the haptic interface to detect the physical interaction as input for controlling the robot. The result is human-like physical interaction, similar to the work presented in [9] and [10].

III. Haptic Interface for Domestic Service Robot

This section describes the robot platform, the interaction design and the approach for the feature.

A. The Robot Platform

The feature was implemented on the domestic service robot Care-O-bot 3 [13] (Fig. 2). Care-O-bot 3 is the third generation of robot assistants developed by the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA). The robot dimensions are 75 cm x 55 cm x 145 cm and it weights 180 kg. Due to its weight, the robot can only be moved using an additional controller. The robot components include a base, a tray, a torso, a head, a manipulator and a gripper.



(a) Actual (b) Simulation

Figure 2. Care-O-bot 3

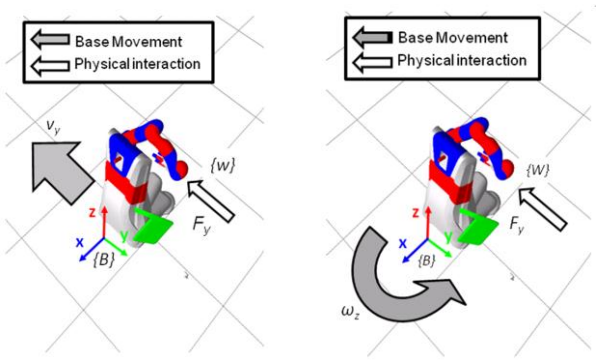
The base has four wheels with two controllable DOF for each wheel and it has the capability of omnidirectional movement. The torso contains processing units and loudspeakers. The manipulator and the tray are fixed to the torso while the head is connected to the torso with two stacked pan-tilt units. The tray can be used for placing objects and it is equipped with a touch screen which is used as a user interface. The head carries the vision sensors and microphone. The manipulator is a 7 DOF Lightweight manipulator (also known as *Leichtbau-Roboter* or LBR) with torque sensor in each joint. The stiffness and damping for each joint of the manipulator can be configured independently. The manipulator has a payload of 7 kg and a three-finger gripper as its end-effector. Each finger of the gripper has two joints and tactile sensors. The robot operates using the open source software framework ROS [14].

B. Interaction Design

The interaction design influences the intuitive aspect of the feature. Defining interactions which are too specific is not suitable because it is difficult for a user to precisely set the pressure, direction and orientation of his/her physical interaction. Therefore, a straightforward interaction is more easy to use. With this consideration, the interaction was designed as follows:

- The point of interaction is around the manipulator's end-effector {W}. It can be at the last joint of the manipulator or the gripper's finger.
- The interaction is measured in the base frame of reference $\{B\}$.
- Interaction in the x-axis (F_x) is mapped to translation velocity in x-axis (v_x).
- Interaction in the y-axis (F_y) is mapped to either angular velocity in z-axis (ω_z) or translational velocity in y axis (v_y).

Point 4 results in two different interaction modes, *shift* mode and *drive* mode. In shift mode, F_y will be mapped to v_y which produces a shifting motion. Drive mode maps F_y to ω_z resulting in the behavior of a differential drive platform. Fig. 3 shows the difference between shift mode and drive mode in mapping F_y into base command using the robot visualization in simulation.



(a) F_y in shift mode (b) F_y in drive mode

Figure 3. Feature interaction design

Both modes accommodate the omnidirectional capabilities of the robot platform. The user is able to switch between the modes while using the feature.

C. Method

Shift mode and drive mode use the same method with minor adjustments. Therefore, only the method for shift mode is described in this section. The input for the feature is the physical interaction with the manipulator. The physical interaction is detected as force vector (${}^{W}F_{ext}$) applied to the manipulator's end-effector. ${}^{W}F_{ext}$ can be calculated from the measured torque as follows:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) - J(q)^{T\,W}F_{ext} \quad (1)$$

where $M(q)$ is the positive inertia matrix, $V(q,\dot{q})$ are the Coriolis and centrifugal torque, $G(q)$ is the gravity torque, τ is the torque applied by motors at the joints and ${}^{W}F_{ext}$ is the external force applied on the end-effector of the manipulator. The calculation of ${}^{W}F_{ext}$ is provided by the manipulator's controller. The next step is applying a frame transformation to the force vector from the end-effector reference frame to the base reference frame as follows:

$$\begin{matrix}B\\W\end{matrix}T\,{}^{W}F_{ext} = {}^{B}F \quad (2)$$

where $\begin{matrix}B\\W\end{matrix}T$ is the transformation matrix from the wrist reference frame to the base reference frame and ${}^{B}F$ is the force vector applied to the end-effector measure in base reference frame. The transformation allows the feature to be used in any manipulator pose. Afterwards, the components of ${}^{B}F$ are mapped to velocity components as follows:

$$f : F_x, F_y \xrightarrow{N_x, N_y} v_x, v_y \quad (3)$$

where F_x is the component of ${}^{B}F$ in x-axis, F_y is the component of ${}^{B}F$ in y-axis, N_x is the linear constant for mapping F_x to v_x and N_y is the linear constant for mapping F_y to v_y. The velocities (v_x, v_y) are filtered using a low pass filter to remove the rapid fluctuation in the signal. Moving average algorithm was chosen for this filter due its aggressiveness in smoothing time series data. The calculation of low pass filter for each velocity is as follows:

$$v_{ma} = \frac{v_t + v_{t-\Delta t} + v_{t-2\Delta t} + \cdots + v_{t-((n+1)\Delta t)}}{n} \quad (4)$$

where v_t is the velocity input (v_x, v_y) at time t and n is the filter width. The filtered output (v_{ma}) is forwarded to the PID controller. The PID controller receives v_{ma} as the set velocity and calculates its difference from the actual velocity as the velocity error (e). The PID controller calculates the controlled velocity ($v_{control}$) as follows:

$$v_{control} = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \quad (5)$$

where K_p, K_i and K_d are the parameter gain for the proportional, integral and derivative component of the PID controller. The following if-else routine is included in the PID controller to check the value of $v_{control}$.

$$v_{output} \begin{cases} v_{max} & if \ |v_{control}| > v_{max} \\ 0 & if \ |v_{control}| < v_{min} \\ v_{control} & else \end{cases} \quad (6)$$

where v_{output} is the PID controller output to the base command velocity, v_{max} is the maximum velocity limit and v_{min} is the minimum velocity threshold. The parameters v_{min} and v_{max} prevent unwanted behaviors in the following conditions:

- Near-zero input values. It is possible that the set velocity oscillates around zero velocity when slowing down to a stop. Instead of stopping, the base will move forward and backward regularly resulting in a shaking motion. The minimum velocity threshold provides a tolerance level where input values which are lower than v_{min} are considered as zero value input.
- High input values. High input values (e.g. through accident) produce a fast motion which can be dangerous for the user and the environment. For safety reasons, the output velocity will stay at v_{max} even when the input force increases.

Fig. 4 shows the overview of the approach.

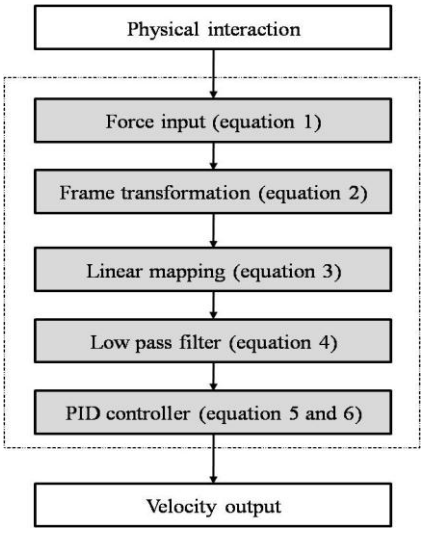


Figure 4. Haptic interface for domestic service robot

With this approach, the resulting behavior depends on the following parameters:

- Linear constant (N_x, N_y),
- The width of the filter (n),
- PID controller gain (K_p, K_i, K_d).

The value and influence of each parameter will be analyzed in the next section.

IV. Results

The feature was evaluated by two tests. The first test is the lab test which investigates the parameter values for the feature and observes the process of translating the physical interaction to the base command. The second test is the user trial where the feature usability is evaluated.

A. Lab Test

The following aspects were found through the initial implementation test:

- Feature's sensitivity depends on linear constant (N_x, N_y)
- Feature's responsiveness depends on the low pass filter width (n) and the PID gain parameter (K_p, K_i, K_d)
- The feature does not work well when the manipulator is in singularity.
- Different pose and joint stiffness level require different parameter values. Poses where one of the joint motor's axis is perpendicular to x-axis has more fluctuation in the input value due to the motor feedback on the applied force F_x. Lower joint stiffness levels also produce more fluctuation in the input value.
- The manipulator can provide a suggestive way on how to use the feature. This characteristic is similar to the way a person positions his/her arm when interacting with others (e.g. when handing object or shaking hands).

(a) Back pose

(b) Side pose

Figure 5. Feature poses

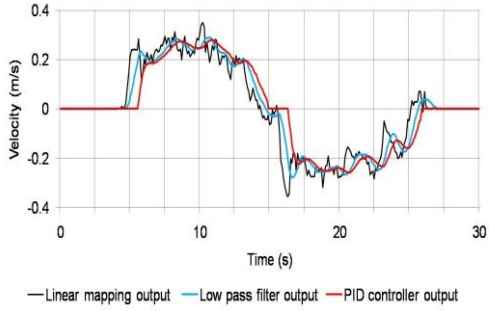
Point 3, 4 and 5 show the importance of the manipulator pose in the feature. Tuning the parameter values for each possible pose and stiffness level would be time consuming. Therefore, a user interface for configuring the parameter values is planned for future work. In addition, several poses was proposed to be used for the feature. Fig. 5 shows two possible poses for the feature.

Each manipulator pose has it strengths and weaknesses. The back pose shown in Fig. 5a is easier for new user to learn the feature because the base reference frame is directly in front of the user. However, the user's front view is blocked by the robot. On the other hand, the side pose shown in Fig. 5b allows the user to monitor the front view easily but requires more learning time for new users. It is important to note that the preferred pose also depends on the type of movement (forward, backward, rotating). The side pose is used for further analysis in this paper and user trial. The parameter values shown in Table I was found to be suitable for side pose.

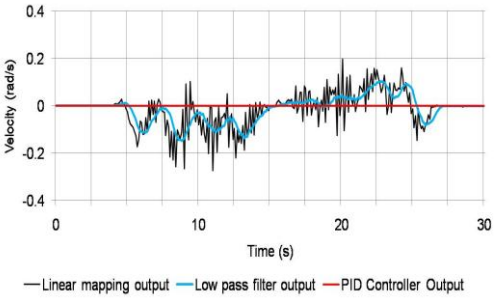
TABLE I. Parameter Values

Parameter	x axis	y axis	
		shift	drive
N_x, N_y	0.0167 m·s^{-1}/ N	0.0125 m·s^{-1}/ N	0.0167 rad·s^{-1}/ N
v_{max}	0.3 m·s^{-1}	0.3 m·s^{-1}	0.0 rad·s^{-1}
v_{min}	0.01 m·s^{-1}	0.01 m·s^{-1}	0.01 rad·s^{-1}
n	50	50	50
K_p, K_d, K_i	0.15, 0.2, 0.0	0.15, 0.2, 0.0	0.15, 0.2, 0.0

The integral element of the PID controller was intentionally removed to avoid overshoot. The investigation of the manipulator pose and the parameter values was followed by a test of performing the motion of moving forward and backward. The purpose of this test is to observe the signal value in the low pass filter and the PID controller. Fig. 6 shows the test results.



(a) Signal in x-axis



(b) Signal in y-axis

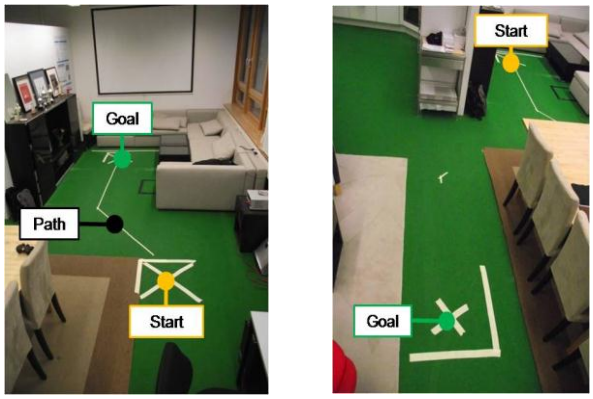
Figure 6. Lab test results

As observed in Fig. 6a, the low pass filter removes the rapid fluctuation in the input signal and the PID controller stabilizes the change of velocity. For example, the

overshoot in signal when reaching the desired velocity (5s < t <7.5s and 15s < t <17.5s) was reduced by the PID controller. Fig. 6b shows the effect of the minimum velocity threshold v_{min} (Equation 6) within the PID controller which results in zero value output.

B. User Trial

The user trial consists of two tests: *follow path* test and *free path* test. Both tests were executed in a domestic environment setting. Fig. 7 shows the setting for both tests.



(a) Follow path test (b) Free path test

Figure 7. User trial environment setting

In the follow path test, the user was requested to maneuver the robot as close as possible along a predefined path. The purpose of the test is to observe the path deviation from different user. In free path test, the user was allowed to choose any path according to their preference when moving the robot. Unlike the follow path test which focuses on having a quantitative measurement, the purpose of this test is to receive the user feedback. In every test, the position of the robot is recorded along with the required time for completing the test. Localization error is neglected and the navigation system is assumed to be accurate. In total, 20 people participated in the user trial. All users have never used the feature before and they were allowed to experiment with the feature for five minutes before performing both test.

1) Follow path test

The path deviation in follow path test is calculated as follows:

$$PD = \frac{1}{n} \sum_{a=1}^{n} D_a \quad (7)$$

where PD is the path deviation of the test, n is the number of robot positions recorded in the tests and D is the shortest distance between the robot position and the path. D is calculated by Algorithm 1.

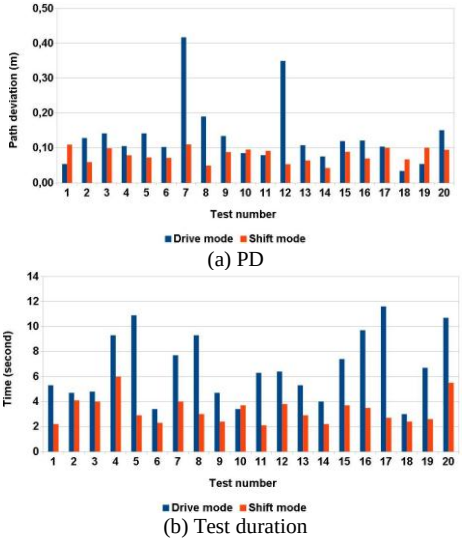
ALGORITHM I: POINT TO PATH DISTANCE CALCULATION

```
CR  = coordinate positions of the robot
NP = number of points which construct the path
CP = coordinate positions of the points which construct the path
D = point to path distance
FOR a = 1 to NP
    E = Euclidian distance of CR  and CP_a
    IF a = 1
        D = E
    ELSE
        IF E < D
            D = E
        END IF
    END IF
END FOR
```

Fig. 8 shows the PD and test duration for each user and Table II shows the statistic of the follow path test.

TABLE II. FOLLOW PATH TEST STATISTICS

Statistic		Mode		Total
		shift	drive	
PD (cm)	Average	8.05	13.49	10.07
	Maximum	11.04	41.73	41.73
	Minimum	4.26	3.4	3.4
	Standard deviation	2.04	9.34	7.22
Test duration (s)	Average	3.3	6.73	5.02
	Maximum	6	11.6	11.6
	Minimum	2.1	3	2.1
	Standard deviation	1.08	2.72	2.68



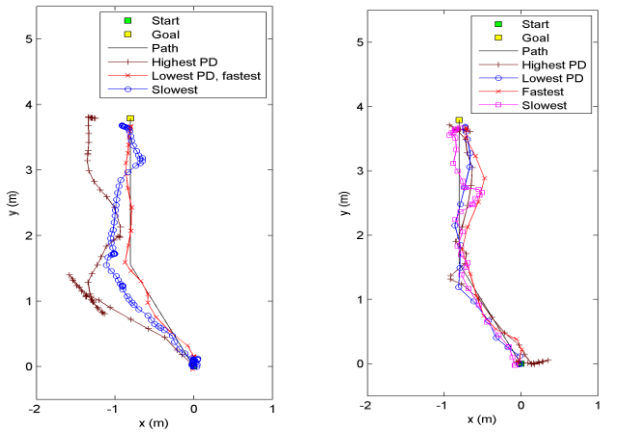
(a) PD

(b) Test duration

Figure 8. Follow path test results

The test results show that high PD rarely occurs (drive mode no. 7 and 12). In such cases, the user was able to perform better when using the shift mode. In total, 40 tests were performed (20 drive mode, 20 shift mode). These resulted in an average PD of 10.77 cm. Shift mode has a lower average of PD and test duration compared to drive mode. Fig. 9 shows the comparison of users' paths in the test with the following attributes:

- The highest PD (drive mode no. 7, shift mode no. 7).
- The lowest PD (drive mode no. 18, shift mode no. 18)
- The fastest (drive mode no. 18, shift mode no. 11)
- The slowest (drive mode no. 17, shift mode no. 4)



(a) Drive mode (b) Shift mode

Figure 9. User paths in the follow path test.

The statistics and users' paths show that shift mode is easier to use than drive mode. This can be attributed to the fact that the drive mode results in the rotational movement of the interaction point (manipulator end-effector). Therefore, the user needs to accommodate the interaction with the change of orientation. The average path deviation of 10.07 cm is low considering that the robot's base footprint is 75 cm x 55 cm. Furthermore, each user learned the feature only shortly before performing the test.

2) Free path test

Fig. 10 shows the test duration for each user and Table III shows the statistics for the free path test. Similar to the follow path test, shift mode has the lower average test duration compared to drive mode. Fig. 11 shows the comparison of the fastest (drive mode no. 18, shift mode no. 11) and the slowest (drive mode no. 3, shift mode no. 10) test results.

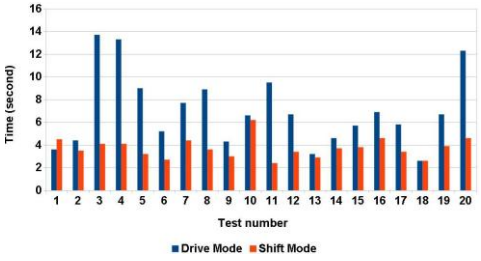
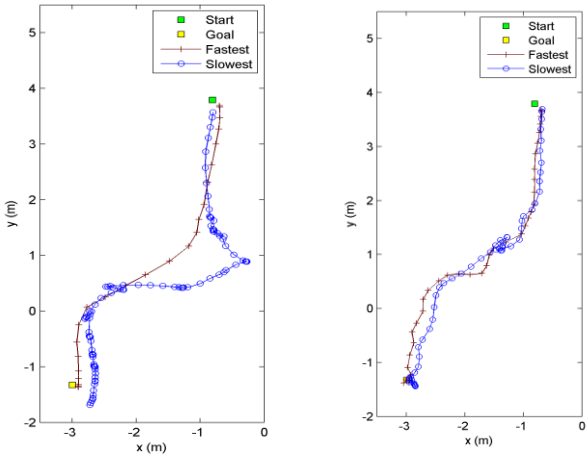


Figure 10. Free path test duration

TABLE III. FREE PATH TEST STATISTIC

Statistic		Mode		Total
		shift	drive	
Test duration(s)	Average	3.3	6.73	5.02
	Maximum	6.2	13.7	13.7
	Minimum	2.4	2.6	2.4
	Standard deviation	0.88	3.24	2.88



(a) Drive mode (b) Shift mode

Figure 11. User paths in the free path test.

Drive mode provides a variety of motion possibilities whereas shift mode is limited. All shift mode test have similar results as shown in Fig. 11b. After the test, the user was asked for their feedback on the feature. The following feedbacks are given:

- Each user had a different preference for the manipulator pose. In the tests, the user was only allowed to use the side pose shown in Fig. 5b. Afterwards, the user was allowed to try out the feature in a different manipulator pose. Some user found the side pose suitable while other preferred other poses.
- Each user had different preference on the feature sensitivity. Similar to the feedback regarding the manipulator poses, there are users who found the feature's sensitivity was to low or too high.
- The feature should accommodate controlling the motion in each component independently (only rotational or translational).

It can be concluded from point 1 and 2 that each user has difference preference in the feature setting (a common fact for any human-robot interaction). Point 3 results in an update for the feature (the ability to disable input in one of the axes). Although the feature was not configured based on the user's preference, all user manage to complete the test without difficulties.

V. CONCLUSION

A feature for controlling a domestic service robot through physical interaction has been presented. The lab

test shows how the approach is effective in translating the physical interaction into base commands. The user trial shows that the feature is intuitive and easy to use. Out of the two proposed interaction modes, shift mode is preferable to drive mode. When using the feature, each user had a different preference for manipulator pose and interface sensitivity. The fact that all users managed to complete the test without using their preferred setting demonstrates that the interaction design is intuitive and the feature is easy to use.

This feature is one possible application for the utilization of the robot's manipulator as its haptic interface. An extension to enable the robot to guide a user (the inversed version of the current feature) is currently under development. Other planned features in haptic interface for domestic service robots include cooperative work and trajectory teaching.

REFERENCES

[1] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, "Robots in human environments: Basic autonomous capabilities," *Int. J. Robotic Res.*, vol. 18, pp. 684–696, 1999.

[2] J. Stueckler and S. Behnke, "Integrating indoor mobility, object manipulation, and intuitive interaction for domestic service tasks," in *Proc. 9th IEEE-RAS Int. Conf. on Humanoids*, Paris, 2009, pp. 506–513.

[3] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sens. Rev*. vol. 24, pp. 16-29, 2008.

[4] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *Proc Int. Conf. on Robotics and Autom.*, Alaska, 2010, pp. 2342–2348.

[5] L. Natale and E. Torres-Jara, "A sensitive approach to grasping," in *Proc. 6th Int. Workshop on Epigenetic Robotics.*, Paris, 2006, pp. 87–94.

[6] F. Naya, J. Yamato, and K. Shinozawa, "Recognizing human touching behaviors using a haptic interface for a pet-robot," in *Proc. IEEE Int. Conf. on Syst., Man, and Cybern*, Tokyo, vol. 2, 1999, pp. 1030-1034.

[7] T. Taichi, M. Takahiro, I. Hiroshi, and H. Norihiro, "Automatic categorization of haptic interactions: What are the typical haptic interactions between a human and a robot?" in *Proc. 6th IEEE-RAS Int. Conf. on Humanoid Robot*, Genova, 2006, pp. 490-496

[8] S. Koo, J. G. Lim, and D. Kwon, "Online touch behavior recognition of hard-cover robot using temporal decision tree classifier," in *Proc. 17th IEEE Int. Symp. on Robot and Hum. Interact. Commun.*, Germany, 2008, pp. 425-429.

[9] M. Inaba, Y. Hoshino, K. Nagasaka, T. Ninomiya, S. Kagami, and H. Inoue, "A full-body tactile sensor suit using electrically conductive fabric and strings," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* Osaka, vol. 2, 1996, pp. 450–457,.

[10] T. Miyashita, T. Tajika, H. Ishiguro, K. Kogure, and N. Hagita, "Haptic communication between humans and robots," *Springer Tracts in Adv. Robotics*, vol. 28, pp. 525-536, 2007.

[11] K. Kosuge, T. Hayashi, Y. Hirata, and R. Tobiyama, "Dance partner robot - Ms DanceR," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst*, Las Vegas, vol. 3, 2003, pp. 3459–3464.

[12] T. Tsuji and T. Ito, "Command recognition by haptic interface on human support robot," in *Proc. IEEE/RSJ Int. Conf. on Intell. robots and Syst* , St. Louis, 2009, pp. 3178–3183.

[13] C. Parlitz, M. Haegele, P. Klein, J. Seifert, and K. Dautenhahn, "Care-O-bot 3 - rationale for human-robot interaction design," in *Proc. 39th Int. Symp. on Robotics*, Seoul, 2008, pp. 275-280.

[14] M. Quigley, K. Conley, B. Gerkey, and J. Faust, "ROS: An open-source Robot Operating System," *ICRA Workshop on Open Source Soft*, 2009.

Rhama Dwiputra is a teaching staff in the Master of Autonomous Systems program at Bonn-Rhein-Sieg University in Sankt Augustin near Bonn, Germany. He received his bachelor degree in Engineering Physics from the Bandung Institute of Technology, Indonesia in 2008 and his master degree in Autonomous Systems from the Bonn-Rhein-Sieg University, Germany in 2013. His research interests are automation, control, robotics, system modeling and human-robot interaction.

He participated in the RoboCup@Home competition as a team member of the b-it-bots team since 2012 and he was involved in the "EduFill" experiment supported by the EU-funded "ECHORD" (European Clearing House for Open Robotics Development) project. He is a member of the research team in the EU-funded "RoCKIn" ("Robot Competitions Kick Innovation in Cognitive Systems and Robotics") project.

Gerhard K. Kraetzschmar is a Professor of Autonomous Systems and Computer Science at Bonn-Rhein-Sieg University in Sankt Augustin near Bonn, Germany, where he teaches in the International Master of Science program in Autonomous Systems.

He received diploma and doctoral degrees in Computer Science from the University of Erlangen in 1988 and 1996, respectively. His research interest include robotics, artificial intelligence, artificial life, and neuroinformatics, with a particular focus on multirobot teams, robot learning and planning, architectures and middleware for autonomous sensorimotor systems, and software engineering in robotics.

He took a leadership role in numerous research projects, including a DFG-sponsored Center for Collaborative Research (SFB-527), a DFG-sponsored Focal Area Programm (SPP-1125), and several projects in the area of educational robotics. He has coordinated the EU-funded project "Roberta-Goes-EU", and is a key architect and principal investigator of the EU-funded projects "BRICS" ("Best Practice in Robotics") and "RoCKIn" ("Robot Competitions Kick Innovation in Cognitive Systems and Robotics"). He is also active as reviewer and evaluator for the European Commission the German Academic Exchange Service (DAAD), and other funding agencies. He represents the RoboCup@Home, RoboCup@Work, and RoboCupJunior Leagues in the Board of Trustees of the RoboCup Federation. After serving as the first Vice President for RoboCupJunior for three years, he has been appointed as Vice President Europe in 2011. He is a member of AAAI, ACM, IEEE CS, RAS, and SMC Societies, and GI.