2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Adaptive Collision-Limitation Behavior for an Assistive Manipulator

Martin F. Stoelen, Virginia F. Tejada, Juan G. Victores, Alberto Jardón Huete,
Fabio Bonsignorio and Carlos Balaguer

***Abstract*— An approach for adaptive shared control of an assistive manipulator is presented. A set of distributed collision and proximity sensors is used to aid in limiting collisions during direct control by the disabled user. Artificial neural networks adapt the use of the proximity sensors online, which limits movements in the direction of an obstacle before a collision occurs. The system learns by associating the different proximity sensors to the collision sensors where collisions are detected. This enables the user and the robot to adapt simultaneously and in real-time, with the objective of converging on a usage of the proximity sensors that increases performance for a given user, robot implementation and task-set. The system was tested in a controlled setting with a simulated 5 DOF assistive manipulator and showed promising reductions in the mean time on simplified manipulation tasks. It extends earlier work by showing that the approach can be applied to full multi-link manipulators.**

I. INTRODUCTION

Assistive manipulators aim to increase the level of independence of its users through aiding in physical Activities of Daily Living (ADL). One example is the commercial Exact Dynamics iArm, see Fig. 1(a). Another is ASIBOT, a 5 Degree Of Freedom (DOF) manipulator developed at Universidad Carlos III de Madrid (UC3M) [1]. See Fig. 1(b). Assistive manipulators typically require operation in close proximity to the disabled or elderly user and safety is therefore critical. They also aim to perform tasks in real-world partially structured environments such as a user's home. For a robot, autonomous or teleoperated by a potentially disabled user, reliably performing such tasks remains a challenge. One solution may be to enable both the user and the robot to use their own sensing, control and planning capabilities in a cooperative way. This is also known as shared control.

Vanacker et al. [2] presented a strategy for filtering the commands coming from the disabled user of a wheelchair using contextual information from sensor readings as well as previous data from able-bodied users. A more recent approach used plan recognition to obtain the probability of a set of user plans, given a set of observed user commands [3]. Another approach for predicting the intent of a wheelchair user was shown in [4], where specific local models for actions, for example moving towards a door, is used. The shared control helped increase safety and reduce the user's cognitive workload. There is also considerable work on mobile robots that have an adjustable degree of autonomy. Here shared control is on the lower end of the scale of potential autonomous modes. See for example [5].

(a) The iArm.

(b) ASIBOT in the UC3M kitchen test bed.

Fig. 1. Examples of assistive manipulators, the goal application.

For higher-DOF platforms like manipulators there is less work available, although collaborative selection among known objects in the environment shows promise [6]. Other work has focused on visual object selection by the user followed by visual servoing by the manipulator [7]. The sensor and action spaces of such manipulators can be large, and exact models of the environment and their relation to every part of the robot can be hard to obtain and maintain. Distributed proximity sensing can help simplify the problem (and reduce the algorithmic complexity), see for example previous work by the authors on an adaptive proximity-based collision-limitation behavior [8]. This is here developed further and extended to a complete assistive manipulator.

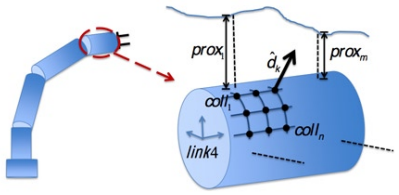
II. SYSTEM DESCRIPTION

A. Overview

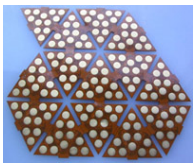
The adaptive shared control described here uses a set of distributed collision and proximity sensors to limit collisions during direct control by the user. The main purpose is to incrementally assist the disabled user in preventing the types of collisions he/she has had in the past, enabling a faster execution. The addition of a collision-limitation behavior can potentially affect negatively both the performance and satisfaction of the user. It would therefore be beneficial for the system to apply the minimum required amount of assistance for a user's abilities and disabilities, while maximizing the overall performance. This is attempted through a Hebbian learning that during collisions associates the pattern of activation of the proximity sensors to behaviors that slow the robot down in the direction of obstacles. There are two principal contributions. First, it can be applied on high-

Martin F. Stoelen, Virginia F. Tejada, Juan G. Victores, Alberto Jardón Huete, and Carlos Balaguer are members of the RoboticsLab research group within the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (mstoelen, vtejada, jcgvicto, ajardon, balaguer)@ing.uc3m.es

Fabio Bonsignorio is a member of the RoboticsLab as Santander Chair of Excellence in Robotics at Universidad Carlos III de Madrid and CEO/founder of Heron Robots of Genova, Italy bonsign@ing.uc3m.es



(a) For each robot link; n collision sensors (grid), m proximity sensors (dotted lines).



(b) Example of existing tactile sensors, from [9].

Fig. 2. Collision and proximity sensing assumed for approach.

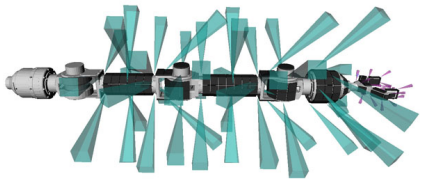


Fig. 3. Collision sensors (black squares) and proximity sensors implemented on the virtual ASIBOT manipulator. Simulated field of view shown for each proximity sensor: Medium-range Sharp GP2D120 and short-range Vishay TCND5000 as green and purple square pyramids, respectively.

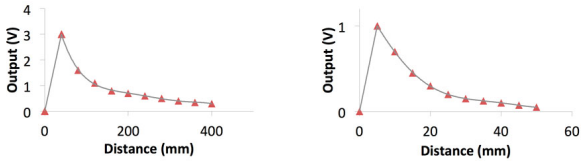
DOF assistive manipulators sharing control with a user in dynamic and only partially structured environments. Second, the system learns online, and in real-time with the user. This allows the robot and user to mutually adapt to each other, giving the user continuous feedback on, and the ability to respond to, changes in the system.

B. Proximity Sensing

In general m proximity sensors and n discrete collision sensors are here assumed for each link of the robot manipulator, see Fig. 2(a). Infrared proximity sensors were chosen, which have previously been used on full-body manipulator sensing [10] and for grasping [11]. Any type of proximity sensor can be used however, even a mix of different types for redundancy. The final implementation had 68 proximity sensors in total. See Fig. 3. 18 were simulated as Vishay TCND5000 (max. dist. 50 mm). These were all distributed over the end-effector. The remaining sensors were simulated as Sharp GP2D120 (max. dist. 400 mm). All proximity sensors had a simulated 10 ° field of view, represented in the simulation by a square 6 by 6 array of point distance measurements. The voltage output of each proximity sensor was simulated based on the minimum distance measured, $prox_j$, and the calibration specifications seen in Fig. 4. This voltage was directly fed as input to the neural networks (p_j).

C. Collision Sensing

There is currently a large research effort focused on developing tactile sensing for robots. See for example Fig. 2(b). For the simple collision sensing used here, the assumed minimum spatial resolution of the tactile sensing was 20 mm, which is well within the capability of the current state of the art [12]. See Fig. 3. A discrete value was used to represent the existence of a collision for each collision sensor. The



(a) Sharp GP2D120. (b) Vishay TCND5000.

Fig. 4. Plots of simulated voltage output for proximity sensors used (p_j). Gray line is output assumed, red triangles are calibration data points.

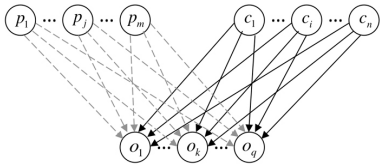


Fig. 5. Link-specific neural network with discounted Hebbian learning for m proximity sensors (dashed grey arrows are synapses), p_j, and fixed weights for n collision sensors (solid black arrows are synapses), c_i.

total number of individually distinguishable collision sensors simulated for the manipulator was 229.

D. Adaptation of Proximity Sensor Usage

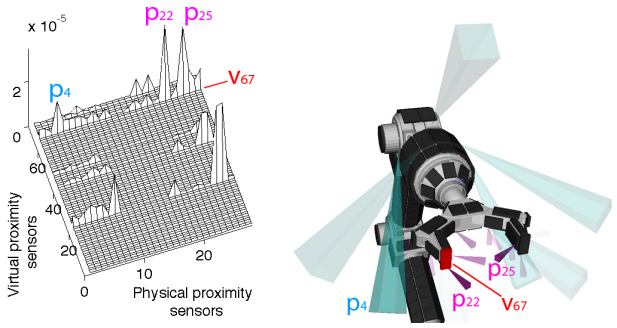
A Hebbian learning approach was chosen, inspired by the Distributed Adaptive Control (DAC) paradigm for autonomous robots [13]. See Fig. 5 for a visualization of the neural network assumed for each link. Virtual proximity sensors with a known location and pose are defined a priori ($\widehat{d}_k$ in Fig. 2(a)). The activation of each of the neurons representing the virtual sensors, o_k, varies linearly with the input it receives from collision (c_i) and proximity (p_j) sensors, according to Equation (1).

$$o_k = \sum_{i=1}^{n} w_{k,i} c_i + \sum_{j=1}^{m} w_{k,j} p_j. \tag{1}$$

Each collision sensor is hardwired (weights initialized offline) to a set of the virtual sensors to generate activation during collisions. For this paper a one-to-one mapping was used (with weight 1), and each pair was therefore assumed to be collocated. The proximity sensors have full connectivity to the layer representing the virtual sensors, and the weights ($w_{k,j}$) are updated using the discounted Hebbian learning rule in Equation (2).

$$\Delta w_{k,j} = \frac{\gamma}{m}(\eta o_k p_j - \epsilon w_{k,j}). \tag{2}$$

Learning therefore occurs whenever there is input from a given collision and proximity sensor, while "forgetting" (the discounting over time) occurs at all times. These processes are controlled with the learning rate η and discount rate ϵ, respectively. The parameter γ can be used to control the rate of change of the weights in general. The parameters η and ϵ are tuned to avoid learning with only proximity input. Here $\gamma = 1$, $\eta = 1.25 * 10^{-5}$ and $\epsilon = 1.5 * 10^{-4}$ was used. See Fig. 6 for an example of the learned neural network weights for one link.



(a) Neural network weights. (b) Corresponding visualization for v_{67}.

Fig. 6. Example neural network weights for final link (participant 5 in experiment). Visualization of weights for one virtual proximity sensor shown (v_{67}). Transparency of square pyramid representing the field of view of a given physical proximity sensor is made to vary with the corresponding weight connecting it to v_{67}. High transparency indicates low weight.

E. The Proximity Ratio

Algorithm 1 shows the scheme used for calculating the maximum proximity ratio for each link. Each virtual sensor "reading" $\vec{d_k}$ is given the magnitude of the inverse of the respective output of the neural network o_k. Then the "reading" that has the largest ratio of the projection of the commanded velocity and its own magnitude is used at each instant. This ratio is here named the proximity ratio r_k. The constant ξ_{vel} is chosen to be small to avoid computational issues if o_k is zero. The proximity ratio can then be used to limit velocities based on the virtual sensor "reading" where a collision will likely occur, and where collisions have occurred in the past (learned by the neural network). A non-zero α_{proj} means velocities in other directions are also slowed down. The translational velocity of each virtual sensor, $\vec{v}_k$, is calculated from the commanded end-effector velocity and the current kinematic pose of the robot. This is further described in Section II-F.

Algorithm 1 Maximum proximity ratio for a link, based on the translational velocities of the virtual sensors, $\vec{v}_k$, the outputs of the link-specific neural network, o_k, and the direction of the respective virtual sensors, $\widehat{d}_k$.

for $k = 1$ to q **do**
$\vec{d_k} = \frac{1}{o_k + \xi_{vel}} \widehat{d}_k$
$proj_k = \vec{v}_k \cdot \widehat{d}_k$
$r_k = \frac{\alpha_{proj} + \beta_{proj} proj_k}{\|\vec{d_k}\|}$
end for
$r_{max} = \max_k(r_k)$

F. Full-Body Collision-Limitation

Fig. 7 shows the schema for the collision-limitation behavior for a complete multi-link manipulator. The commanded velocities of the end-effector, $\vec{v}_{ee}$, are here represented in the robot base frame (b superscript). Using an iterative solver for the inverse Jacobian, the corresponding joint velocities for all joints are calculated. The translational velocities of each sensor for each link are then estimated, and used to calculate the proximity ratios (Section II-E). Finally, the original commanded end-effector velocities are limited based on the maximum proximity ratio for the complete manipulator. The output velocity, $\vec{v}_{ee,out}$, is the user-commanded velocity $\vec{v}_{ee}$ divided by this ratio. The behavior will only activate if the ratio exceeds one. For the behavior to work consistently for sensors on any link of a given manipulator, it is therefore assumed that $\vec{v}_{s_j}$ scales proportionally with $\vec{v}_{ee}$ for a given instantaneous pose. Audio feedback was used to help the user assess when the collision-limitation behavior was acting. This consisted of simple tones being played with breaks in between. The frequency of the alternation was proportional to the current maximum proximity ratio, see Fig. 7. The frequency of the tones was used to identify the link, from low frequency at the base to high frequency at the end-effector.

III. EXPERIMENT METHOD

A. Participants

8 able-bodied participants were used, all graduate and under-graduate students at UC3M. There were 3 female and 5 male, all right-handed. 4 had previous experience with 3D input devices and 3 had previous experience controlling robots. The mean age was 23.7, with a range from 19 to 40. Each participant was paid €10 for participation, and all gave their informed to participate. The experiment was performed in accordance with UC3M regulations.

B. Simulated Disability

As in previous work by the same authors [8], a noise was added to the user input, according to Equation (3). This was Gaussian noise, low-pass filtered at 2 Hz and generated independently for each Cartesian component of the noise vector ($\vec{z} = [z_x, z_z, z_{pitch}, z_{yaw}]^T$). The magnitude of the translational velocities caused by the noise increased proportionally to the magnitude of the translational velocities commanded by the user, with some noise existing also when the user did not indicate movement (non-zero α_{noise}). Similarly for the rotational velocities. See Fig. 8(b) for example trajectories.

$$
\begin{aligned}
&\vec{v}_{ee} = \vec{v}_{input} + \vec{v}_{noise},\\
&where:\\
&\vec{v} = [\vec{v}_{trans}, \vec{v}_{rot}]^T = [v_x, v_z, v_{pitch}, v_{yaw}]^T,\\
&and:\\
&\vec{v}_{noise,trans} = \vec{z}_{trans}(\alpha_{noise} + \beta_{noise}||\vec{v}_{input,trans}||),\\
&\vec{v}_{noise,rot} = \vec{z}_{rot}(\alpha_{noise} + \beta_{noise}||\vec{v}_{input,rot}||),\\
&z_x, z_z, z_{pitch}, z_{yaw} \sim \mathcal{N}(0, \sigma^2).
\end{aligned}
\tag{3}
$$

This served as a crude estimation of the loss of control caused by a physical impairment, and allowed for a homogeneous set of able-bodied participants. While real end-users are needed to validate the clinical credibility of any assistive technology [14], the use of simulated disabilities can help drive the early development. For example a random component being added to an able-bodied user's computer mouse

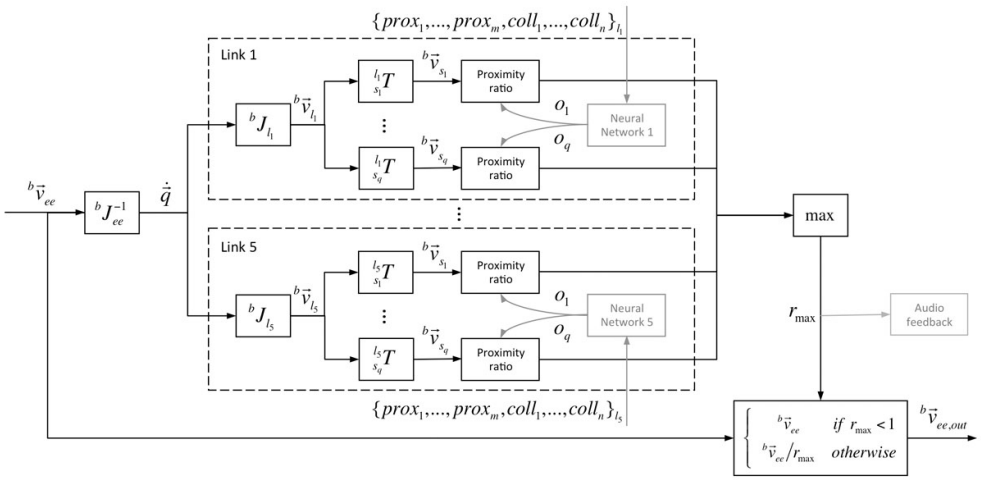
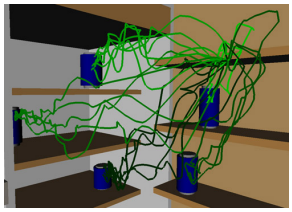


Fig. 7. The full-body collision limitation schema. The current joint angles are used in calculating the Jacobian (J), but are here omitted for clarity. The b superscript is used to denote the robot base frame, while T denotes transformation. Grey color indicates external modules.

movements [15]. Exploratory works in shared control have also used simulated disabilities [3]. A velocity-dependent noise was here used, as it amplifies an effect already seen in the speed-accuracy trade-off of many human movements. That is, faster movements require greater forces in the muscles, which again may introduce more nervous system noise [16]. An increase in the signal-dependent neuromotor noise has been related to stroke-related motor deficiencies [17], and children with dystonia [18].



(a) Example translational trajectories. Partic. 6, shared control.

(b) Participant performing experiment (with 3D effects activated).

Fig. 8. The experiment setup.

C. Simulated Environment and Tasks

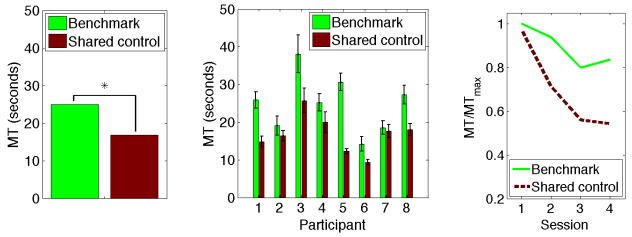
The simulated environment used in the experiment can be seen in Fig. 8 and Fig. 12. The experiment was performed in the OpenRAVE simulator [19], running at approx. 50 Hz. The ASIBOT robot was simulated to be attached to the right-hand side of the user's wheelchair. Implementing the system first in simulation provided a flexible and easily controllable environment for including human trials actively in the development process [8]. The tasks performed involved moving the end-effector of the robot from an initial resting position to a pre-grasp position around one of 5 simulated cans in the virtual environment. See Fig. 8(a). For a given trial the target can was red, while the remaining were blue. A trial was automatically judged as completed when the two fingers of the robot end-effector was positioned around the thickest part of the can, with a velocity below a given threshold. The participants controlled the Cartesian x, z, $pitch$ and yaw velocities of the robot end-effector, in the end-effector local frame. The arm was reset to the initial position if any part of the robot collided with the environment, the physical model of the user, or any of the cans. The participant would then loose the time spent up until the collision, as the timer kept running. For all trials the participants were instructed to attempt to achieve the lowest mean times possible, while keeping in mind that collisions were costly in terms of time. See [8] for details on this performance metric.

D. Physical Setup

The physical experiment setup can be seen in Fig. 8(b). The input device was a SpaceNavigator 6 DOF joystick. The simulation of the robot in the environment was displayed in 3D on a 40 inch (approx. 102 cm) display (Samsung 3D TV, UE40D8000). The participants used active 3D glasses for depth perception. A colored timer was also shown. The execution rate of the shared control was 40-70 Hz, while the 4 neural networks operated at 80-100 Hz, nominally. The experiment was run on an 8-core Dell i7-2600 @ 3.4 GHz.

E. Procedure

The testing was performed over 2 days for each participant, with multiple sessions each day. The total time committed each day was about one hour per participant. Each session consisted of 3 repetitions of each of the 5 tasks (5 target locations), for 15 trials in total. The first day the participants were introduced to the experimental setup and was given 3 sessions for training. This was followed by 2 sessions for establishing a benchmark. The shared control was not used. The second day the participants were introduced to the shared control, and were first given a maximum of 2 training sessions with the adaptive shared control activated. That is, each participant was told to attempt to achieve a comfortable level of assistance, and could decide when the training should be ended. Then the adaptation was



(a) Average MT over participants. (b) Individual MT, error bars indicate standard error. (c) Average MT learning curves.

Fig. 9. The Mean Time (MT) with and without (benchmark condition) the shared control. Based on the two non-training sessions for each condition.

disabled (both learning and forgetting), and each participant was given 2 sessions to establish the performance with the shared control, using the learned neural network weights.

IV. RESULTS AND DISCUSSION

The overall results, expressed in Mean Time (MT), can be seen in Fig. 9. MT is here the mean time over all attempts for all tasks for one participant. There was a statistically significant improvement in average MT over participants of 32.5% with the shared control. A paired t-test was used, with $t(7) = 3.96$, $p = 0.005$. This is comparable to previous results [8]. Fig. 9(b) shows the equivalent comparison for each participant. While all participants had a reduction in the MT metric, there were large individual differences in the amount of reduction, ranging from 5.3% for participant 7 to 59.9% for participant 5. That is, in under 20 minutes each participant was able to "negotiate" a level of assistance that at least did not inhibit, and for most cases seems to have improved the performance. One participant noted that *"[the system] helped out when close to the target, could move faster"*, another that *"it helped a lot on the more complex tasks, and on the simpler especially in the last second before finishing, where before attempted to go step-wise"*.

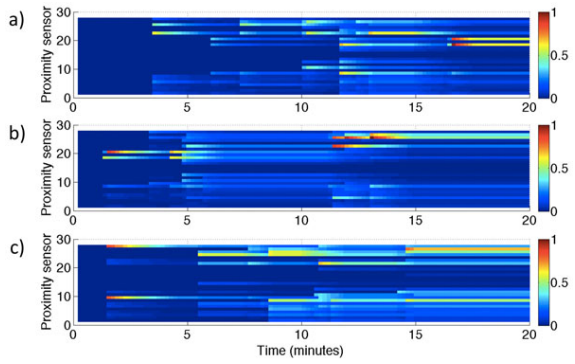


Fig. 10. Examples of the development of the neural network weights for the final link. Mean weights for each proximity sensor of final link, normalized with maximum over 3 participants: a) 3, b) 5 and c) 6.

See Fig. 10 for examples of the development of the neural network weights for the final link. A corresponding visualization of the final usage of the proximity sensors is given in Fig. 11. While participant 3 primarily received

(a) Participant 3. (b) Participant 5. (c) Participant 6.

Fig. 11. Visualizations of the learned proximity sensor usage for the final link. Transparency of square pyramid representing the field of view of a given sensor is made to vary with the mean neural network weights for sensor. High transparency indicates low usage, and vice versa.

assistance when close to obstacles on the left, participants 5 and 6 had a more symmetric usage of the proximity sensors. The latter two participants also used the in-hand sensors, which are useful for slowing the robot down in the last moments of the tasks. An example of the effect of the shared control on the end-effector velocities can be seen in Fig. 12. Significant assistance is provided as the participant is approaching the target can, by the limitation of the commanded x and z velocities from 12 seconds onwards. It can also be seen that the system allows the user to freely perform gross movements when there is sufficient space (first 4 seconds here). In a "real life" implementation the user should likely be given control over when to adapt the proximity sensor usage. Different sets of weights could also be stored for different contexts, for example indoor vs outdoor environments.

A within-subject experiment design was used, with the same order for each participant. Extensive practice was therefore given to attempt to reduce the effect of learning across conditions. Fig. 9(c) shows that the MT improved considerably during the first 2 sessions for each condition and stabilized reasonably well for the last 2 sessions (in which performance was measured). Furthermore, the experiment was spread over 2 days to avoid excessive fatigue in the participants. There is extensive evidence of the enhancing effect of sleep on for example sequential finger tapping, although there is less evidence of a significant effect on for example pursuit tracking [20]. In any case, future experiments should be counterbalanced, with separate days for training, benchmarking and shared control.

V. CONCLUSIONS AND FUTURE WORK

The adaptive collision-limitation behavior developed can be applied to high-DOF manipulator platforms operating in environments where a set of sufficiently accurate environment models and sensor to end-effector mappings are difficult to obtain. This might make it interesting also in other applications, such as in teleoperation. The online nature of the adaptation seems important for maintaining the system predictable from the user's perspective. A controlled experiment with realistic simulations of the tasks, sensors and the 5 DOF ASIBOT manipulator showed promising results for 8 able-bodied participants with simulated disabilities. Future work will refine the experimental paradigm used, and explore the application on a larger set of tasks (including

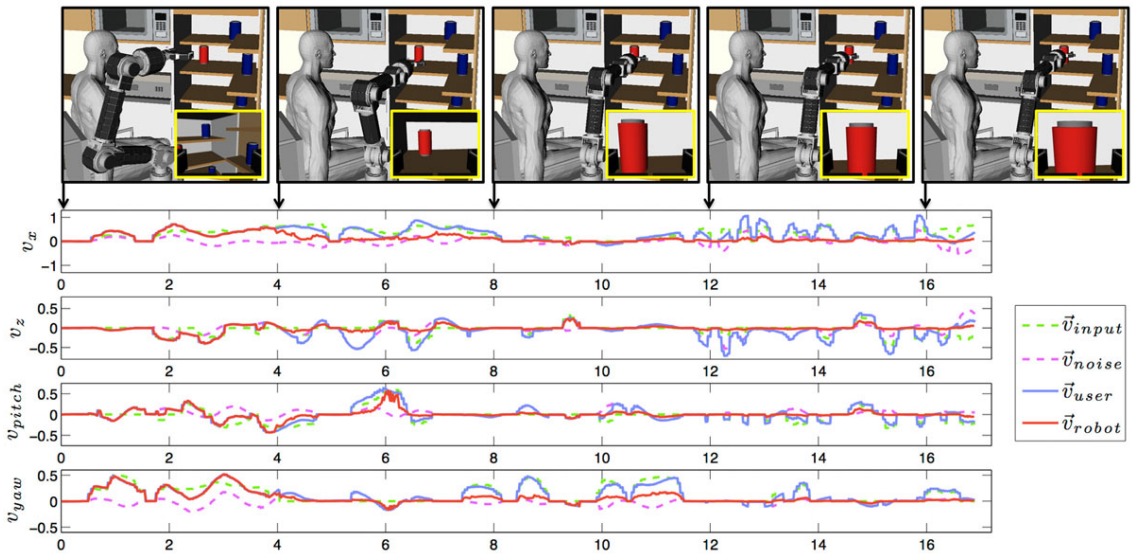


Fig. 12. One attempt by participant 1 on one task with shared control. Cartesian x, z, $pitch$ and yaw components of velocities (in end-effector frame) shown, with time in seconds on the x-axis. A discrepancy between the input ($\vec{v}_{user}$) and the output ($\vec{v}_{robot}$) velocities of the shared control means assistance is provided. Actual robot poses along trajectory shown, but the camera angle is altered for visualization. End-effector camera view shown in inserts. Description of phases: 0-4 seconds: highly coordinated gross movement, 4-8 seconds: adjustment of pitch during forward movement, 8-12 seconds: mainly yaw adjustments, 12-16 seconds: the final approach to the target.

generalization to unseen tasks) and to usage over a longer time-frame. The workload and overall satisfaction of disabled users controlling virtual and physical implementations of the system will have to be investigated.

VI. ACKNOWLEDGMENTS

Funding has been received from the ARCADIA project DPI2010-21047-C02-0, thanks to the MINECO ministry of Spain, and the ROBOCITY2030 II project S2009/DPI-1559, thanks to Comunidad de Madrid and EU structural funds.

REFERENCES

[1] A. Jardón, A. Giménez, R. Correal, R. Cabas, S. Martínez and C. Balaguer, A portable light-weight climbing robot for personal assistance applications, *Industrial Robot: An International Journal*, vol. 33. no. 4, pp. 303-307, 2006.

[2] G. Vanacker, D. Vanhooydonck, E. Demeester, A. Huntemann, A. Degeest, and H. Brussel, "Adaptive filtering approach to improve wheelchair driving performance", *in ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, United Kingdom, 2006, pp. 527-532.

[3] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, H. Brussel and M. Nuttin, User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving, *Journal of Autonomous Robots*, vol. 24, pp. 193-211, 2008.

[4] T. Carlson and Y. Demiris, "Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 876-888, 2012.

[5] M.A. Goodrich, D.R. Olsen, J.W. Crandall and T.J. Palmer, "Experiments in adjustable autonomy", *in Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001, pp. 1624-1629.

[6] B. Pitzer, M. Styer, C. Bersch, C. DuHadway and J. Becker, "Towards Perceptual Shared Autonomy for Robotic Mobile Manipulation", *in Proceedings of IEEE International Conference on Robotic and Automation*, Shanghai, China, 2011, pp. 6245-6251.

[7] K. Tsui, H. Yanco, D. Kontak and L. Beliveau, "Development and evaluation of a flexible interface for a wheelchair mounted robotic arm," *in Proc. of the 3rd ACM/IEEE Internat. Conf. on Human Robot Interaction*, Amsterdam, The Netherlands, 2008, pp. 105-112.

[8] M.F. Stoelen, V.F. Tejada, A. Jardón, F. Bonsignorio and C. Balaguer, "Benchmarking Shared Control for Assistive Manipulators: From Controllability to the Speed-Accuracy Trade-Off", *in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 4386-4391.

[9] G. Cannata, M. Maggiali, G. Metta and G. Sandini, "An Embedded Artificial Skin for Humanoid Robots", *in Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, 2008.

[10] E. Cheung and V.J. Lumelsky, Proximity sensing in robot manipulator motion planning: system and implementation issues, *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, 1989, pp. 740-751.

[11] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena and A.Y. Ng, "Reactive grasping using optical proximity sensors", *in IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, 2009, pp. 2098-2105.

[12] H. Yousef, M. Boukallel and K. Althoefer, Tactile sensing for dexterous in-hand manipulation in robotics - A review, *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 171-187, 2011.

[13] R. Pfeifer and C. Scheier, *Understanding Intelligence*, MIT Press, Cambridge, MA; 1999.

[14] K.M. Tsui and H.A. Yanco, "Towards Establishing Clinical Credibility for Rehabilitation and Assistive Robots Through Experimental Design", *in Workshop on Good experimental methodology in robotics, Robotics Science and Systems*, Seattle, WA, USA, 2009.

[15] J. Mankoff, H. Fait and R. Juang, Evaluating accessibility by simulating the experiences of users with vision or motor impairments, *IBM Systems Journal*, vol. 44, no. 3, pp. 505-517, 2005.

[16] A.A. Faisal, L.P.J. Selen and D.M. Wolpert, Noise in the nervous system, *Nat. Rev. Neurosci.*, vol. 9, no. 4, pp. 292-303, 2008.

[17] P.H. McCrea and J.J. Eng, Consequences of increased neuromotor noise for reaching movements in persons with stroke, *Exp. Brain Res.*, vol 162, pp. 70-77, 2005.

[18] T.D. Sanger, J. Kaiser and B. Placek, Reaching Movements in Childhood Dystonia Contain Signal-Dependent Noise, *Journal of Child Neurology*, vol. 20, no. 6, pp. 489-496, 2005.

[19] R. Diankov, Automated Construction of Robotics Manipulation Programs, *PhD thesis*, Robotics Institute, Carnegie Mellon University, Aug. 2010.

[20] K. Blischke, D. Erlacher, H. Kresin, S. Brueckner and A. Malangré, "Benefits of Sleep in Motor Learning - Prospects and Limitations", *Journal of Human Kinetics*, vol. 20, pp. 23-35, 2008.