

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/275960113>

# Adaptive Aid on Targeted Robot Manipulator Movements in Tele-Assistance

Article in *Paladyn* · August 2016

DOI: 10.1515/pjbr-2016-0002

CITATIONS

2

READS

342

5 authors, including:



**Martin F. Stoelen**

Høgskulen på Vestlandet

50 PUBLICATIONS 129 CITATIONS

[SEE PROFILE](#)



**Virginia F. Tejada**

University of Plymouth

13 PUBLICATIONS 29 CITATIONS

[SEE PROFILE](#)



**Alberto JARDÓN HUETE**

University Carlos III de Madrid

124 PUBLICATIONS 862 CITATIONS

[SEE PROFILE](#)



**Fabio Bonsignorio**

Scuola Superiore Sant'Anna

65 PUBLICATIONS 168 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



GummiArm - a compliant and printable robot arm [View project](#)



Self-adaptable robot assisted manufacturing in intelligent sustainable work cells [View project](#)

## Research Article

## Open Access

Martin F. Stoelen\*, Virginia F. Tejada, Alberto Jardón, Fabio Bonsignorio, and Carlos Balaguer

# Adaptive Aid on Targeted Robot Manipulator Movements in Tele-Assistance

DOI 10.1515/pjbr-2016-0002

Received October 21, 2014; accepted April 7, 2015

**Abstract:** The teleoperation of robot manipulators over the internet suffers from variable delays in the communications. Here we address a tele-assistance scenario, where a remote operator assists a disabled or elderly user on daily life tasks. Our behavioral approach uses local environment information from robot sensing to help enable faster execution for a given movement tolerance. This is achieved through a controller that automatically slows the operator down before having collisions, using a set of distributed proximity sensors. The controller is made to gradually increase the assistance in situations similar to those where collisions have occurred in the past, thus adapting to the given operator, robot and task-set. Two controlled virtual experiments for tele-assistance with a 5 DOF manipulator were performed, with 300 ms and 600 ms mean variable round-trip delays. The results showed significant improvements in the median times of 12.6% and 16.5%, respectively. Improvements in the subjective workload were also seen with the controller. A first implementation on a physical robot manipulator is described.

**Keywords:** Teleoperation, assistive robotics, time-delays, haptics, shared control, neural network, proximity sensing, Distributed Adaptive Control (DAC)

## 1 Introduction

Robot teleoperation over the internet is subject to time-variable delays typically ranging from hundreds of milliseconds to seconds. Such delays are especially limiting for robot manipulators, which typically require coordinated, fast and precise movements. See Fig. 1a for an abstraction of the problem. One potential application is the teleoperation of assistive robot manipulators mounted on wheelchairs. These robots give users who are constrained to a wheelchair, and who have a low amount of mobility in the upper-limbs, independence on some daily life tasks. Commercial examples include the Exact Dynamics iArm and the Kinova Jaco. At Universidad Carlos III de Madrid (UC3M) there is ongoing research with the two platforms ASIBOT [1] and AMOR, see Fig. 2. The latter was developed by Exact Dynamics in Holland. Note that such robot manipulators are also known as Wheelchair-Mounted Robot Arms (WMRA).

All current commercial platforms of this type are controlled directly by the user, with little environment sensing and autonomy in the robot. The users have access to a range of personalised user interfaces, from chin joysticks to simple push buttons. However, the task execution times can be very high [2], and so can the mental and physical demand. Research is currently underway to explore what assistance can be provided to the user to help amend these issues [3–6]. Shared autonomy on mobile manipulators is also being explored [7]. However, many tasks may simply be too difficult to achieve for the user, or too tiring. We believe that enabling a remote operator to aid the user on more complex tasks could significantly increase the usefulness of such devices. The remote operator could for example be a family member or a care provider, controlling the robot over an internet connection. This type of teleoperation of a personal assistive robot manipulator, here denoted tele-assistance, is the scenario application aimed for in this paper. As tele-assistance would typically be performed over an internet connection, considerable and variable time-delays are likely.

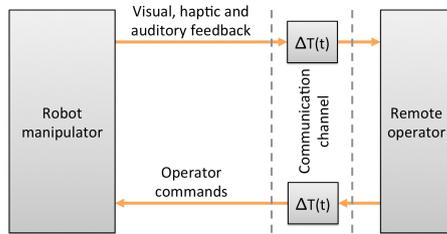
A large body of work on teleoperation systems that address time-delays make two important assumptions: 1)

**\*Corresponding Author: Martin F. Stoelen:** Centre for Robotics and Neural Systems (CRNS), Plymouth University, UK, E-mail: martin.stoelen@plymouth.ac.uk

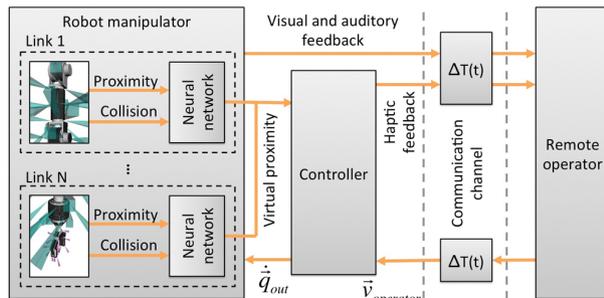
**Virginia F. Tejada, Alberto Jardón, Carlos Balaguer:** RoboticsLab, Department of Systems Engineering and Automation, Universidad Carlos III of Madrid, E-mail: (vtejada, ajardon, balaguer)@ing.uc3m.es

**Fabio Bonsignorio:** The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa and CEO/founder of Heron Robots, Genoa, Italy, E-mail: fabio.bonsignorio@sssupsup.it, fabio.bonsignorio@heronrobots.com





(a) Teleoperation under variable time-delays.



(b) Tele-assistance with the adaptive aid proposed here.

**Figure 1:** An overview of the problem (a) and the approach followed for improving tele-assistance (b). A remote operator is helping control the assistive robot manipulator of an elderly or disabled user on a difficult task, but is hindered by variable time-delays in the communications ( $\Delta T(t)$  in figure) and limited camera views. The approach followed here provides an adaptive aid for the teleoperator, where a controller learns to make use of distributed proximity sensors on the manipulator to limit collisions with the environment. The main idea is that this aid helps the teleoperator enforce the accuracy requirements of the task, enabling a faster execution. The system learns by directly associating proximity sensor readings with collisions when they occur, through distributed and real-time neural networks. The controller operates on the Cartesian velocities provided by the user,  $\vec{v}_{operator}$ , to produce suitable robot joint velocities  $\dot{\vec{q}}_{out}$ .



(a) ASIBOT.



(b) Exact Dynamics AMOR.

**Figure 2:** Assistive robot manipulators used for research at UC3M.

that exact models of the environment are available (from sensor data), and 2) that the required human movements can be explicitly modeled [8]. Neither seem realistic for assistive robot manipulators, which may be used in any number of environments, such as in the user's home or in a grocery store, and on a great variety of tasks. However, we

can assume that the operator will be performing targeted movements to objects in the environment. Such targeted movements are characterized by the trade-off between the speed of execution and the accuracy required to avoid errors (for example collisions). This is exemplified by Fitts' law [9], which has been extensively used for evaluating human-machine interfaces.

We believe that a controller that can help enforce the accuracy requirements of a teleoperation task can indirectly allow the operator to move with greater speed, and reduce the effect of the time-delays, much like obstacle-based force-feedback can improve performance on simple corridor following tasks [10]. We also believe that sampling the environment directly through distributed proximity sensing can be an effective way to estimate the robot's effective state. This paper presents a controller for teleoperating assistive manipulators based on these ideas, where the aid provided is gradually adapted to each operator's needs. Thus an attempt is made to cater for differences in the skills of different operators, in the time-delays for different user-operator pairs, and in the typical tasks on which the users require assistance.

This is achieved through: i) a controller that limits the velocity of the robot (and provides haptic feedback) based on distributed collision and proximity sensors, and ii) neural networks that adapt in real-time the use of the proximity sensors, based on past collisions.

## 2 Related Work

One way to mitigate the effects of time-delays in teleoperation is predictive displays. These displays use virtual models of the robot in its environment to provide immediate feedback to the operator on the outcome of commanded actions. One potential application of predictive displays is space teleoperation [11]. Predictive displays have been shown to aid under conditions with variable time-delays, for example in remote vehicle operation [12]. A limitation of predictive displays is the requirement for accurate models of the robot, and the environment in which it is used.

Haptic force feedback allows human operators to perform complex tasks with physical contact. This includes applications in the medical field, handling of toxic materials, mechanical design and outer space exploration [13]. However, the variable time-delays, packet losses, and disconnections that can occur over an internet connection can induce unstable force, degrade the performance and can be harmful to the teleoperators [14]. Different time-delay compensation techniques have been developed to

overcome this problem, such as wave-scattering theory (and the wave-variable approach) [15] and Smith predictors [16].

Approaches that take into account information gained online about the remote Environment, the human Operator or the desired Task to be performed have been labelled EOT-adapted controllers [8]. This is a similar concept to "shared control", where the user and the robot use their own sensing, control and planning capabilities in a cooperative way. Examples of assistive technologies include wheelchairs that attempt to aid the disabled or elderly user in performing navigation tasks [17, 18]. Shared control has also been proposed for teleoperation, for example to reduce the velocity before impact (and the impact force) [19]. Haptic shared control has been shown to lead to performance improvements, but sometimes at the cost of the operator feeling that he/she is fighting the system [20]. The work presented here complements the typical approaches given that: a) it assists the operator on targeted movements in unknown environments, b) it does not require extensive up-to-date models of the robot, it's environment, and it's user, and c) the aid provided is adapted to each operator's needs in real-time.

Our approach was inspired by the Distributed Adaptive Control (DAC) paradigm [21–23]. In particular, to use distributed sensing, and to adapt the usage of the sensors through a simple associative learning that has some of the main constituents of Classical Conditioning [24]: i) a predefined value system, expressed in combinations of Unconditioned Stimuli (US) and Unconditioned Reflexes (UR), and ii) a mechanism for associating Conditioned Stimuli (CS) representations to US representations. An example application of DAC is a mobile robot with distributed collision and proximity sensing. The collision sensors are hardwired to predefined motor actions (the UR) that turn the robot away from the obstacles. A collision avoidance behavior, representing a Conditioned Reflex (CR), is then gradually learned by associating proximity sensor readings (the CS) with the collisions sensors (the US) during collisions. That is, proximity sensors that detect objects close-by during collisions will gradually begin activating the obstacle avoidance by themselves.

The system thus creates a sense-associate-act coupling, where the environment is used as a communications channel and where the adaptation of the behavior will bias the future sensory information it receives. This can be taken advantage of to stabilize the system [22]. In the above example, once the robot no longer collides with the obstacles, it will stop learning new connections. See also previous work by the current authors on adaptive proximity-based collision-limitation for disabled users controlling

assistive manipulators directly [5, 6]. The work presented in the current paper extends this approach to teleoperation with time-varying delays.

### 3 System Description

The overall system architecture can be seen in Fig. 1b. The remote operator has access to visual, force and audio feedback from the robotic manipulator, and uses this information to command the robot with Cartesian velocities  $\vec{v}_{operator}$ . The input device and control mode used here are further described in section 3.1. Both robot commands and operator feedback is passed over a communication channel with variable time-delays, such as a typical internet connection. The manipulator is covered with a set of collision and proximity sensors (see section 3.2). The usage of these sensors in each link of the manipulator is regulated by a dedicated link-local Neural Network (NN), which learns by associating proximity and collision sensors during collisions with the environment. See section 3.3. The output of each neural network is a set of virtual proximity readings, which are used in the controller. Here the robot is slowed down, and force/audio feedback is provided to the operator. This is further described in section 3.4.

#### 3.1 Haptic Control Mode



**(a)** Stylus local frame. **(b)** Haptic forces.

**Figure 3:** The PHANTOM Omni haptic input device, with a visualisation of the control mode assumed.

A hybrid control mode based on both position and velocity input was used here, see Fig. 3. A Sensable PHANTOM Omni haptic device was used, which has 6 DOF position sensing and 3 DOF ( $x$ ,  $y$ , and  $z$ ) force feedback. The  $x$ ,  $y$  and  $z$  displacements of the stylus, in the stylus-local frame, was used to control the corresponding velocities of the robot end-effector. The *pitch* angle of the robot end-

effector matched the *pitch* angle of the stylus at all times. The total force provided,  $\vec{F}_{tot}$  in Equation (1), was composed of two components. First, a spring-like component that always returned the stylus to the same position in space (the origin defined) if let go of. This is denoted as  $\vec{F}_{spring}$  in Equation (2). Second, a component that provided feedback during aid, denoted as  $\vec{F}_{prox}$ . More details on the calculation of the latter can be found in Equation (6) in Section 3.3.

$$\vec{F}_{tot} = [F_x, F_y, F_z]^T = \vec{F}_{spring} + \vec{F}_{prox}. \quad (1)$$

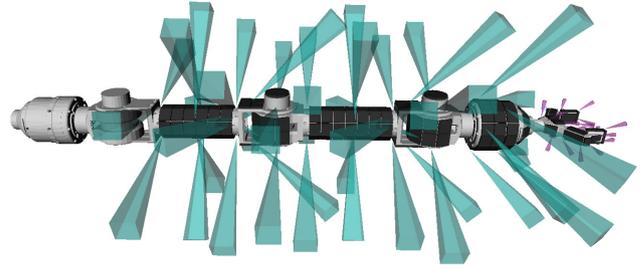
$$\begin{aligned} \vec{F}_{spring} &= k_{spring} \Delta \vec{P}, \\ \text{where:} & \\ \Delta \vec{P} &= [\Delta x, \Delta y, \Delta z]^T. \end{aligned} \quad (2)$$

### 3.2 Collision and Proximity Sensing

The total number of individually distinguishable collision sensors simulated for the manipulator was 229. Each sensor returned a simulated force of contact based on the depth of penetration with an obstacle. See Fig. 4. Thus 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 [25]. Infrared proximity sensors have previously been used in full-body proximity sensing on robot manipulators [26], and for grasping [27]. For the former, over 1000 infrared proximity sensors were used to perform online movement planning and execution in unknown and dynamic environments, over 20 years ago. This remains a challenging task today, even with the excellent sensors technology (e.g. 3D time-of-flight sensors) and high-power computers available. See Section 4.2 and Section 5.2, respectively, for details on the specific implementations used for the two experiments performed here.

### 3.3 Neural Networks

There are potentially many ways in which a distributed set of proximity and collisions sensors could be utilized. It may be desirable to attempt to automatically adapt the usage to both operator abilities and scenario of usage, given the large set of parameters that requires tuning for correct operation. This is the general approach followed here, see Fig. 5. The approach assumes  $n$  collision sensors and  $m$  proximity sensors for each link, each of which is represented by a neuron in a respective input layer of a neural network.



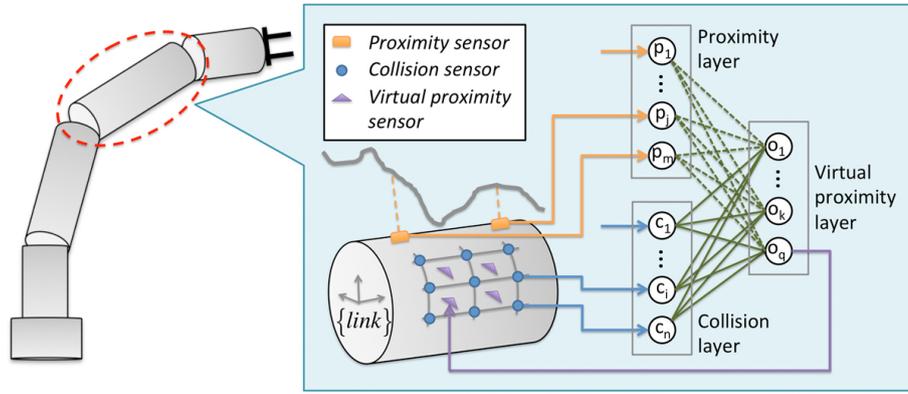
**Figure 4:** Collision sensors (black squares) and proximity sensors implemented on the virtual ASIBOT manipulator. Simulated field of view shown for each proximity sensor: Long-range Sharp GP2D120 and short-range Vishay TCND5000 as green and purple square pyramids, respectively. This was the proximity sensing configuration used for the first experiment presented here.

An output layer with  $q$  neurons is used to represent a set of virtual proximity sensors. The activation of the neurons in this layer ( $o_k$ ) is linear, as seen in Equation (3). The collision sensor neurons are hardwired to the virtual proximity sensor neurons (solid green lines in Fig. 5). The distribution of these weights depends on the proximity of the virtual proximity sensor to a given collision sensor. In the simplest case (the one used here), each collision sensor has a unity weight connecting it to the closest virtual proximity sensor. Whenever a collision sensor activates, it thus also activates a virtual proximity neuron. The discounted Hebbian learning rule in Equation (4) is then used to associate this activation with the simultaneous activation of any proximity sensors on the same link. The learning thus modifies the synapse weights between real and virtual proximity sensors, the dashed green lines in Fig. 5. In Equation (4)  $\gamma$  is the learning rate,  $\epsilon$  the discount rate. These must be tuned to get the desired learning behaviour, that is, to make sure the robot learns from the collisions, but also gradually "forgets" and lowers the aid when there have been none for a given period.

$$o_k = \sum_{i=1}^n w_{k,i} c_i + \sum_{j=1}^m w_{k,j} p_j. \quad (3)$$

$$\Delta w_{k,j} = \frac{\gamma}{m} (\eta o_k p_j - \epsilon w_{k,j}). \quad (4)$$

The unit normal vector representing the direction of each virtual sensor  $k$  is denominated  $\hat{d}_k$ . To represent the "distance" from each virtual proximity sensor to an obstacle (based on the NN output) another variable was defined,  $e_k$ . This was made to vary proportionally with the inverse of the activation of the output neuron for the same sensor ( $o_k$ ), as seen in Equation (5). Each virtual proximity sensor can be associated with multiple proximity sensors and multiple collision sensors, thus being capable of



**Figure 5:** Visualization of the adaptive component of the aid on a generic manipulator. Each link has a set of distributed proximity sensors and collision sensors. The sensor signals are fed into two layers of a link-local neural network. This neural network adapts the usage of the proximity sensors during collisions with the environment, through a simple Hebbian associative learning mechanism. The output of the neural network is used to represent "virtual" proximity sensors, which depend on the actual sensory input, but also previous associations between sensed collisions and sensed proximity. The neural network weights are shown in green: dashed lines indicate Hebbian learning (proximity layer), while solid lines indicate fixed weights (collision layer).

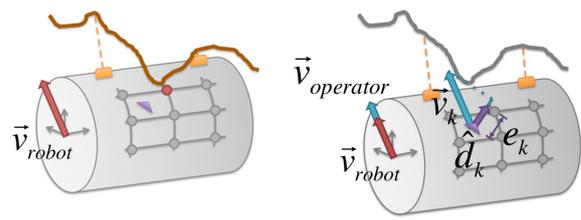
representing quite complex sensorial "fingerprints". The number of virtual proximity sensors can also be scaled to fit the computational resources available.

$$e_k = \frac{1}{\epsilon_k + o_k}. \tag{5}$$

### 3.4 Controller

The collision-limitation behavior then uses the output of the NNs to reduce the magnitude of the commanded velocity at each instant. That is, to limit  $\vec{v}_{operator}$  to  $\vec{v}_{robot}$ , as shown in Fig. 6. To do this the system calculates a proximity ratio  $r$  for all virtual proximity sensors, shown in Algorithm 1. The proximity ratio for a given virtual proximity sensor  $k$  increases with proximity to an object (low  $e_k$ ) and with a high translational velocity in the direction of the virtual sensor vector (projection of  $\vec{v}_k$  on  $\hat{d}_k$ ). The ratio is therefore high when there is a low time to collision. The maximum proximity ratio for each link is used at any time.  $\alpha_{proj}$  is typically set to zero, to only limit velocities in the direction of obstacles.

Fig. 7 shows the schema used for limiting the velocity for a multi-link manipulator. The received velocities of the end-effector,  $\vec{v}_{operator}$ , 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 first calculated, then each link is treated separately. Using the known kinematic structure of the robot and the current joint angles, the translational velocities of each sensor for each link is calculated. These are then used together with the output of the link-specific neural

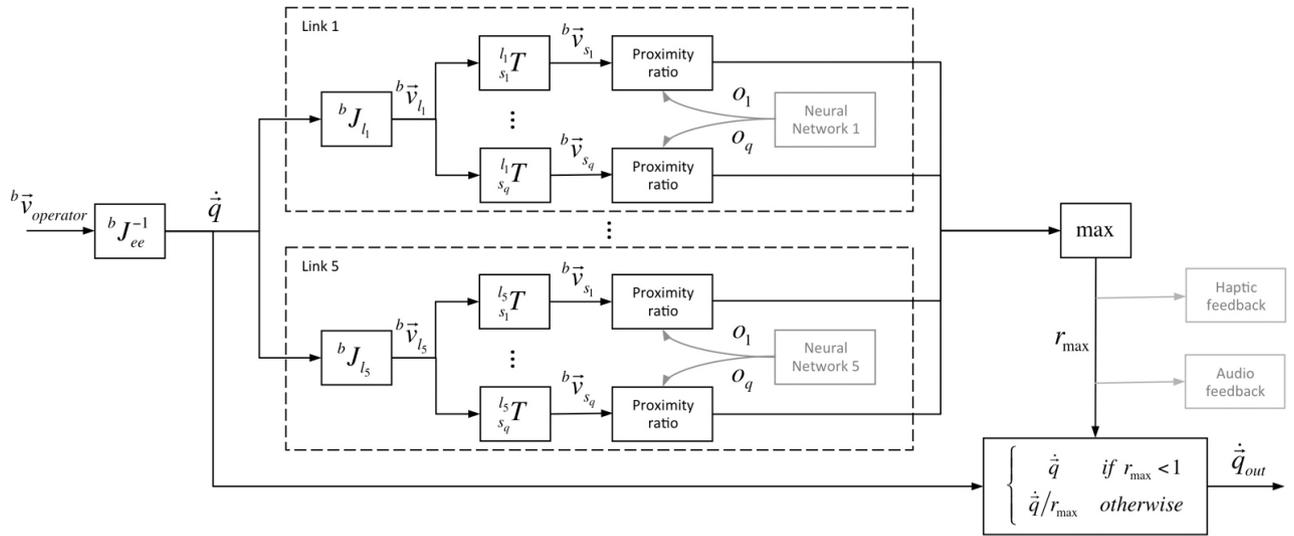


**(a)** Association of activation in virtual proximity sensors with collision sensed with the environment during operator-commanded movements ( $\vec{v}_{robot} = \vec{v}_{operator}$ ). **(b)** Reduction of operator velocity ( $\|\vec{v}_{robot}\| < \|\vec{v}_{operator}\|$ ) using projected displacement ( $\vec{v}_k$ ), direction ( $\hat{d}_k$ ) of virtual proximity sensor  $k$ , and its NN-dependent "distance"  $e_k$ .

**Figure 6:** The system behaviour before (b), and after (c) a collision. During the collision the system has learned to increase the influence of a given proximity sensor  $k$  on the velocity of the robot (for that link),  $\vec{v}_{robot}$ . That is, the activation of proximity sensor  $k$  will now allow a lower velocity than that provided by the operator,  $\vec{v}_{operator}$ , in the direction of that sensor.

network to produce the maximum proximity ratio for that link, as described above.

The output robot joint velocities,  $\vec{q}_{out}$ , are the operator-commanded joint velocities  $\vec{q}$  divided by this ratio. The behavior will only activate if the maximum proximity ratio ( $r_{max}$ ) exceeds one. This enables the limitation of velocity based on the learned virtual sensor usage of the complete manipulator. The same maximum proximity ratio was used to generate force feedback for the haptic input device. The force component added was calculated as shown in Equation (6) using a normalized version of the maximum proximity ratio ( $r_{norm}$ ) and the gains  $\vec{k}_{prox}$ .



**Figure 7:** Schematic of the adaptive aid extended to a multi-link manipulator. The  $b$  superscript denotes the robot base frame and grey color indicates external modules. The velocities commanded by the operator,  $b\vec{v}_{operator}$ , are resolved into joint velocities, and then Cartesian velocities for the virtual proximity sensors  $s_1, \dots, s_q$  for each link. The proximity ratio for each sensor is then calculated based on the Cartesian sensor velocity and the output of the link-specific neural network. The maximum proximity ratio over all sensors on all links,  $r_{max}$ , is used to limit the robot joint velocities and to provide audio and haptic feedback.

```

for  $k = 1$  to  $q$  do
     $proj_k = \vec{v}_k \cdot \hat{d}_k$ ;
     $r_k = \frac{\alpha_{proj} + \beta_{proj} proj_k}{e_k}$ ;
end
 $r_{max,link} = \max_k(r_k)$ ;

```

**Algorithm 1:** The maximum proximity ratio for a given link, based on the translational velocities of the virtual proximity sensors,  $\vec{v}_k$ , the outputs of the link-specific neural network, represented by  $e_k$ , and the direction of the respective virtual proximity sensors,  $\hat{d}_k$ .

This provided the operator with direct feedback on collisions, and on learned proximity usage, that is, when and to what degree the system was reducing the velocity of the robot manipulator. A redundant audio feedback was also provided. This consisted of simple tones being played with breaks in between, similar to the system for notifying the driver of obstacles when reversing a car. The frequency of the alternation was proportional to the current maximum proximity ratio.

$$\vec{F}_{prox} = r_{norm} \vec{k}_{prox} \Delta \hat{P},$$

where:

$$\vec{k}_{prox} = [k_x, k_y, k_z]^T.$$

(6)

## 4 Experiment 1: Adaptive Aid on Tele-Assistance

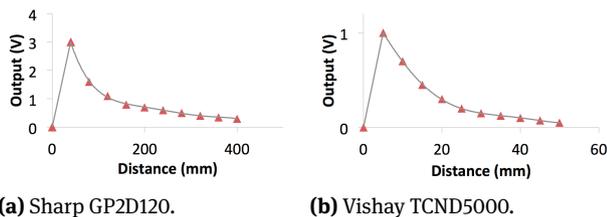
### 4.1 Introduction

This section describes the first experimental evaluation of the system developed. The objective of the experiment was to evaluate the improvements in performance with the adaptive aid for a situation where the user of an assistive manipulator is aided on manipulation tasks by a remote operator over an internet connection. A balanced within-subject experiment design was used, with two conditions: a) benchmark with no aid provided, and b) aid with the controller described above (after adaptation). The performance metrics used were the time to complete the tasks and the subjective workload (NASA-TLX) [28]. NASA-TLX has previously been used in, for example, experiments on robotic telepresence [29]. An internet connection with a round-trip variable time-delay of 300 ms (standard deviation of 30 ms) was approximated (see Section 4.3.2).

### 4.2 Implementation

For this experiment there were 68 infrared proximity sensors in total, of which 18 were simulated as Vishay TCND5000 with a maximum sensed distance of 50 mm. See Fig. 4. These were all distributed over the end-effector.

The remaining sensors were simulated as Sharp GP2D120, with a maximum sensed distance of 400 mm. All proximity sensors had a simulated  $10^\circ$  field of view, represented in the simulation by a square 6 by 6 array of point distance measurements. The lowest of the 36 point distance measurements was used at any time. The voltage output of each proximity sensor,  $p_j$ , was simulated based on the distance measured and the calibration specifications of the different sensor types. See Fig. 8. That is, the signal used by the neural network was increased with decreasing distance measured (in the nominal range of the sensor).



**Figure 8:** Plots of simulated voltage output for the proximity sensors used. Gray line is output assumed, red triangles indicate calibration data points for the physical sensors.

## 4.3 Experiment Method

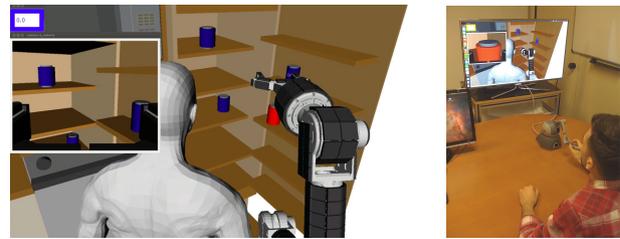
### 4.3.1 Participants

The participants were 9 undergraduate students of UC3M, 5 male and 4 female. A 10th participant was not able to finish all sessions due to other commitments, and was therefore not included in the analysis. All participants were right-handed. 2 had previous experience with 3D input devices. The mean age was 19.9, with a range from 19 to 21. Each participant was paid €10 for participation.

### 4.3.2 Simulated Environment and Time-Delay

A tele-assistance scenario was simulated, as seen in Fig. 9. The OpenRAVE [30] virtual environment was used, running at approx. 50 Hz. The participants were given the simulated view from one camera mounted behind the wheelchair-user, and one mounted on the end-effector of the robot. A time-varying round-trip time delay was simulated, with a mean of 300 ms and a standard deviation of 30 ms. The variation of the time delay was random, using a Gaussian noise low-pass filtered at 0.1 Hz. Given that the robot was virtual and there were no hard limits as to when

the robot should react, the full time delay was added to the user input only.



**(a)** The view provided to the participants, with end-effector camera-view and timer. **(b)** Physical setup for the experiment.

**Figure 9:** The first simulated tele-assistance experiment.

### 4.3.3 Tasks Performed

The tasks performed involved moving the end-effector of the robot from an initial resting position (see Fig. 9a) to a pre-grasp position around one of 5 simulated cans in the virtual environment. 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 were positioned around the thickest part of the can, stopped or with a small remaining velocity magnitude. The participants controlled the Cartesian  $x$ ,  $z$ , *pitch* and *yaw* velocities of the robot end-effector, in the end-effector local frame. The timer changed color to red and incremented 10 seconds if any part of the robot collided with the environment, the physical model of the user, or any of the target cans. For all trials the participants were instructed to attempt to achieve the lowest times possible, while keeping in mind that collisions were costly in terms of time.

### 4.3.4 Physical Setup

The physical experiment setup can be seen in Fig. 9b. The input device used was a Sensable PHANTOM Omni haptic device, as described in Section 3.1. The two camera views simulated seen in Fig. 9a were displayed on a 40 inch (approx. 102 cm) display (Samsung 3D TV, UE40D8000), at a distance of about 2 meters. A colored timer was also shown on the display.

### 4.3.5 Procedure

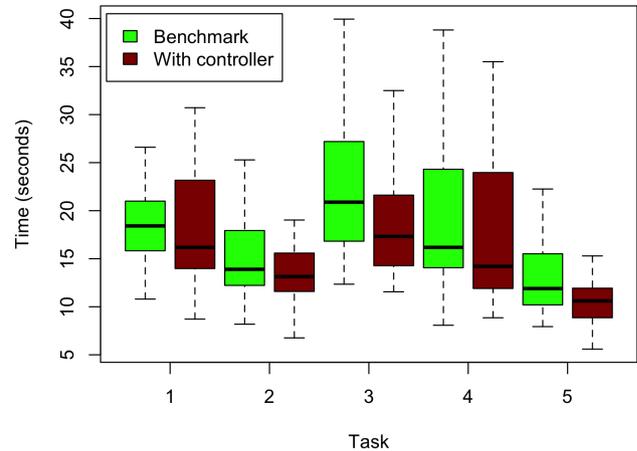
Each participant performed 3 days of testing, with about one hour of commitment each day. Each day consisted in 4 sessions. The first day was used for training only. On the second and third day the tasks were performed with or without (benchmark condition) the aid of the controller. Each participant was tested on 6 repetitions of each of the 5 tasks for each condition. The order of the conditions were assigned randomly to each participant. Two training sessions were given before measuring the performance for each condition, with performance being measured over the last two sessions only. In the condition with aid, the adaptation was only active during the training sessions. That is, each participant was told to attempt to achieve a comfortable level of aid, and could decide when the training should be ended. Then the adaptation was disabled, and each participant was given two sessions to establish the performance using the neural network weights learned during training.

## 4.4 Results and Discussion

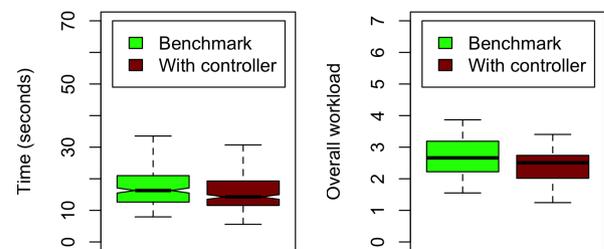
The completion time for each task with and without the aid of the controller can be seen in Fig. 10. All tasks had a lower median with the controller. Wilcoxon signed-rank tests showed that there were statistically significant differences for tasks 3 ( $Z = 2.547, p = 0.008$ ) and task 5 ( $Z = 2.666, p = 0.004$ ). As shown in Fig. 11a there was a significant 12.6% reduction of the median overall completion time. A Wilcoxon signed-rank test was used ( $Z = 2.666, p = 0.004$ ).

Fig. 12 shows the translations of the end-effector performed by one participant for the two conditions. A similar strategy seems to be used for solving the tasks with and without the proximity-based haptic aid. The results for the individual scales of the subjective workload measures (NASA-TLX) can be seen in Fig. 13. There was a statistically significant 23.1% reduction in the median Temporal Demand ( $TD$ ) with the controller ( $Z = 2.025, p = 0.039$ ), again using the Wilcoxon signed-rank tests. Physical Demand ( $PD$ ) and Frustration ( $FR$ ) were higher with the controller, however  $PD$  was given the lowest weight by the participants. See Fig. 11b for the comparison of the overall workload. The plot shows a slight reduction in the median, but this difference was not statistically significant.

Fig. 14 shows an example trajectory with the aid of the controller. For this specific attempt the system mainly slowed the participant down in the last 8 seconds of the trajectory. During the first 8 seconds a gross movement in free space was performed, where the robot moved ex-



**Figure 10:** Completion time for each task for the first experiment, with and without (benchmark condition) the aid of the controller. Box plots based on data from 9 participants with 6 repetitions per task ( $N = 54$ ). Outliers not shown for clarity. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile.

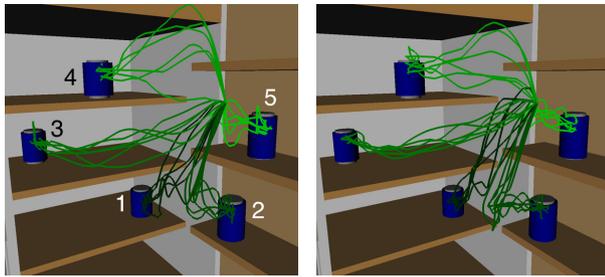


**(a)** Completion time. Box plot based on data from 9 participants with 6 repetitions on 5 tasks ( $N = 270$ ). Notch based on 95% confidence interval for the median. Outliers not shown for clarity.

**(b)** Overall NASA-TLX subjective workload. Box plot based on data from 9 participants ( $N = 9$ ), with overall workload calculated from scales and weights [28].

**Figure 11:** Overall results for the first experiment, with and without (benchmark condition) the aid of the controller. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile.

actly as the user commanded. When the end-effector of the robot entered in-between the shelves the system aided by: 1) applying opposing forces to the operator's input device displacements, and 2) reducing the Cartesian velocity. The aid was based on the previously learned proximity usage. This corresponded to the final fine-tuning of the position of the fingers with respect to the target can. Though only one example, this is the type of help that we hope may give



(a) Benchmark condition. Task numbering also shown. (b) With controller.

Figure 12: Example translational trajectories, for participant 6.

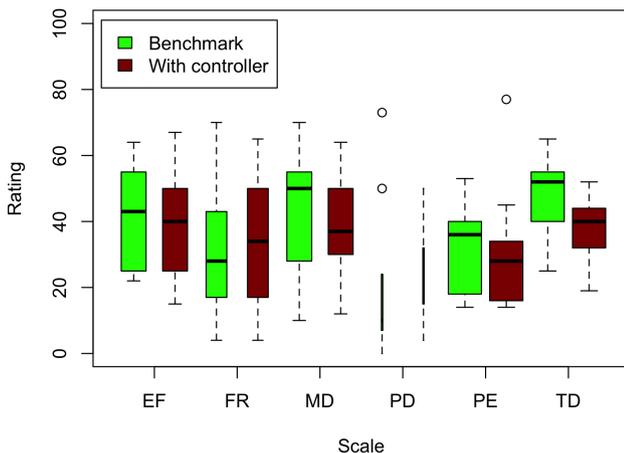


Figure 13: NASA-TLX subjective workload ratings for each scale for the first experiment, with and without (benchmark condition) the aid of the controller. Box plots based on data from 9 participants ( $N = 9$ ). The width of each box represents the mean weight given to the respective scale by the participants. The median of *PD* for the benchmark condition was 10, with the controller 16. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile. Scales: *EF* - Effort, *FR* - Frustration, *MD* - Mental Demand, *PD* - Physical Demand, *PE* - Performance, *TD* - Temporal Demand.

the operator the confidence to move with a higher velocity during the gross segment of a targeted movement.

Overall, the results give some indications that the intervention of the system improved the quantitative performance, while also improving (or at least not significantly worsening) the qualitative experience from the operator's perspective. Thus the approach holds some promise for mitigating the type of variable time-delay used. A key aspect is the adaptation performed, tuning the usage of the 68 proximity sensors in real-time to the needs of each operator.

## 5 Experiment 2: Exploring Longer Time Delays and Generalization

### 5.1 Introduction

A second experiment was performed, to test performance on a longer (600 ms) and more variable time delay, and to attempt some generalisation to unseen tasks.

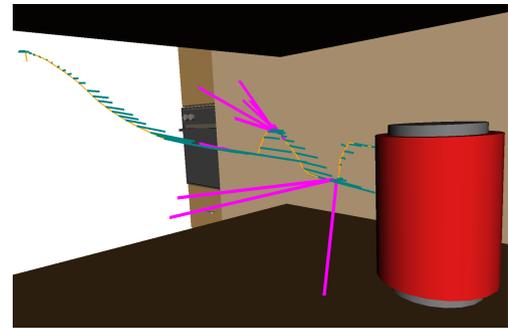
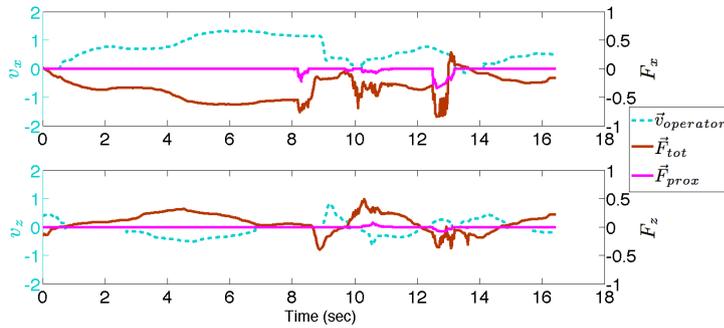
### 5.2 Implementation

There were three changes to the implementation from the first experiment. The first change made was to differentiate the collision and proximity activation in the audio feedback. That is, the collisions were now signalled by low frequency pulses of static length. The help provided by the proximity sensors were still signalled by variable length pulses of higher frequency. The second change made was to limit the connectivity of the collision and proximity sensors. Sensors further away than 0.25 m and with a rotation that differed by more than  $70^\circ$  were not connected. This was done to help avoid associating proximity sensors with collisions that occurred in the other extremes of a link. The third change was to replace all the proximity sensors with digital Silicon Labs Si1143 sensors. All the proximity sensors had a simulated  $30^\circ$  field of view, represented in the simulation by a square 6 by 6 array of point distance measurements. The lowest of the 36 point distance measurements was used at any time. The voltage output of each proximity sensor,  $p_j$ , was simulated based on the distance measured and the calibration specifications. See Fig. 15 for the distribution used, and Section 6 for the first physical implementation using these sensors.

### 5.3 Experiment Method

#### 5.3.1 Participants

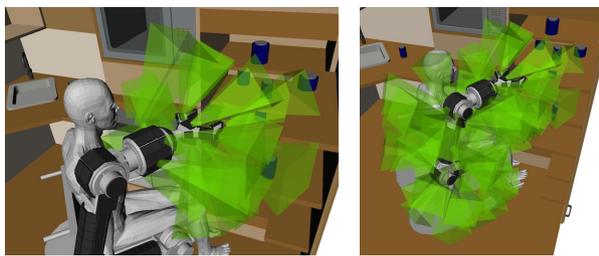
The participants were 8 undergraduate students of UC3M, 4 male and 4 female. 5 participants were right-handed, 3 were left-handed. 3 had previous experience with 3D input devices. The mean age was 23.0, with a range from 20 to 31. Each participant was paid €10 for participation.



(a) Time-varying profiles for the  $x$  and  $z$  components of three variables: i) the velocity commanded by the operator,  $\vec{v}_{operator}$ , ii) the total haptic force felt by the participant  $\vec{F}_{tot}$ , and iii) the force component due to the aid by the controller,  $\vec{F}_{prox}$ .

(b) Corresponding end-effector trajectory. Dark green lines indicate the  $\hat{X}$  axis of the end-effector at each point (length indicates velocity), and pink lines indicate the aid-related haptic force,  $\vec{F}_{prox}$ .

Figure 14: Visualization of a trajectory for one attempt by participant 6 on task 3 with the controller active.



(a) Sensors on last link.

(b) All sensors.

Figure 15: Field of view of simulated proximity sensors for the second experiment (Silicon Labs Si1143 sensors).

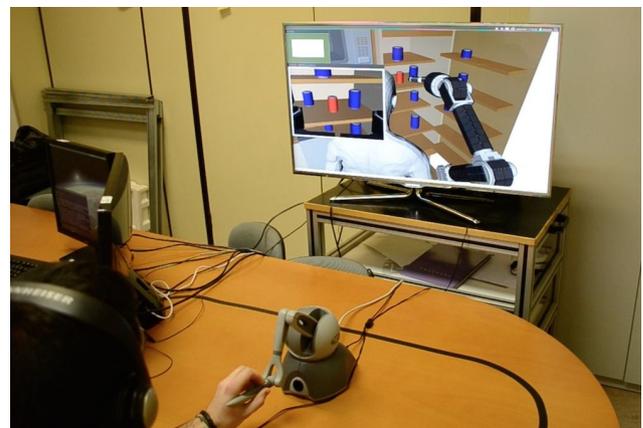


Figure 16: The setup for the second experiment.

### 5.3.2 Simulated Environment and Time-Delay

A very similar tele-assistance scenario as used in the first experiment was simulated, as seen in Fig. 16. See section 4.3.2 for details. A longer time delay was used, with a mean of 600 ms. It was also made much more variable, with a standard deviation of 120 ms. This meant the delay could vary from close to none up to over a second. The variation of the time delay was random, using a Gaussian noise low-pass filtered at 0.1 Hz, as in the first experiment.

### 5.3.3 Tasks Performed

The 5 tasks performed in the first experiment were included, but also 5 more target cans, as can be seen in Fig. 17. The additional 5 cans were not used when adapting the controller, but only when measuring the performance. Again, the tasks performed involved moving the end-effector of the robot from an initial resting position (see Fig. 9a) to a pre-grasp position around one of the sim-

ulated cans in the virtual environment. See Section 4.3.3 for details.

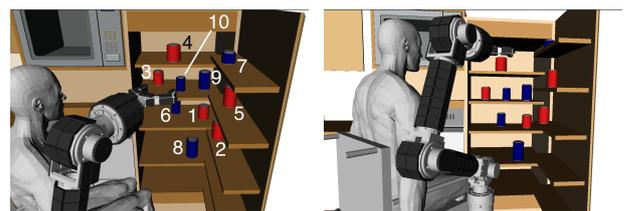


Figure 17: Two views of the tasks for the second experiment. The 5 red cans were used for both learning and testing (tasks 1 to 5), the 5 blue cans only for testing (tasks 6 to 10).

### 5.3.4 Physical Setup

The same physical setup as for the first experiment was used, as can be seen in Fig. 16. Section 4.3.4 contains the details also for this experiment.

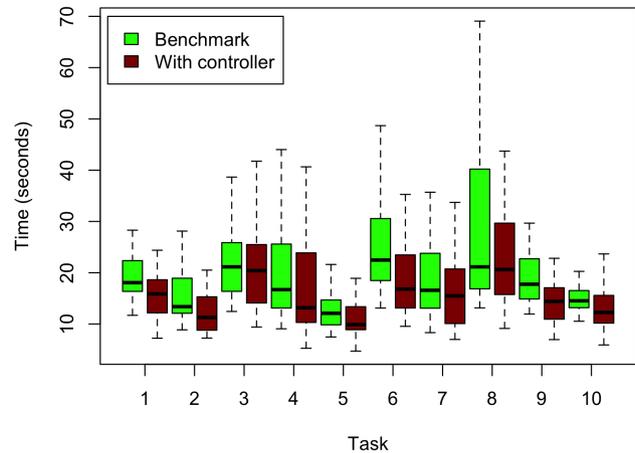
### 5.3.5 Procedure

Each participant performed 3 days of testing, with about one hour of commitment each day. Each day consisted in 3 sessions. Each session had 20 attempts in total. The first day was used for training only. On the second and third day the tasks were performed with or without (benchmark condition) the aid of the controller. Each participant was tested on 4 repetitions of each of the 10 tasks for each condition. The order of the conditions were assigned randomly to each participant (balanced within-subject design). 1 training session was given before measuring performance for each condition, with performance being measured over the last 2 sessions only. In the assisted condition, the adaptation was only active during the training session. This was the only session when only the 5 original target cans were used. 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 disabled, and each participant was given 2 sessions to establish the performance using the neural network weights learned during the training.

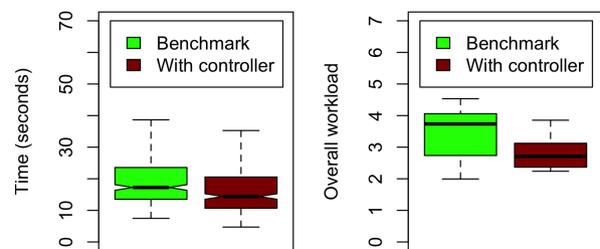
## 5.4 Results and Discussion

The completion time for each task for the second experiment can be seen in Fig. 18. All tasks had a lower median with the controller. Wilcoxon signed-rank tests showed that there were statistically significant differences for task 5 ( $Z = 2.521, p = 0.008$ ) and task 10 ( $Z = 2.381, p = 0.016$ ). As shown in Fig. 11a there was a significant 16.5% reduction of the median of the overall completion time. A Wilcoxon signed-rank test was used ( $Z = 2.381, p = 0.016$ ).

The results for the individual scales of the subjective workload measures (NASA-TLX) can be seen in Fig. 20. The Wilcoxon signed-rank tests showed that there was a statistically significant 20.0% reduction in the median Temporal Demand ( $TD$ ) with the controller ( $Z = 2.252, p = 0.031$ ). Physical Demand ( $PD$ ) was higher with the controller, but like in the first experiment it was given the lowest weight by the participants. All other scales were lower, but not statistically significant. See Fig. 19b for the comparison of the overall workload. The plot shows a considerable reduction



**Figure 18:** Completion time for each task for the second experiment, with and without (benchmark condition) the aid of the controller. Box plots based on data from 8 participants with 4 repetitions per task ( $N = 32$ ). Outliers not shown for clarity. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile.



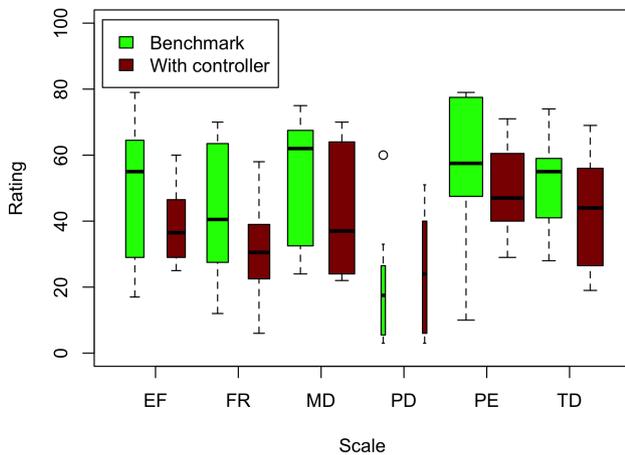
**(a)** Completion time. Box plot based on data from 8 participants with 4 repetitions on 10 tasks ( $N = 320$ ). Notch based on 95% confidence interval for the median. Outliers not shown for clarity.

**(b)** Overall NASA-TLX subjective workload. Box plot based on data from 8 participants ( $N = 8$ ), with overall workload calculated from scales and weights [28].

**Figure 19:** Overall results for the second experiment, with and without (benchmark condition) the aid of the controller. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile.

in the median, but this difference was only weakly statistically significant with the Wilcoxon signed-rank test ( $Z = 1.820, p = 0.078$ ).

Fig. 21 shows the development of the system during the session when the learning was activated. The weights, and the resulting level of assistance, varies from participant to participant. All participants, except participant 6, started the session with several collisions during the 5 first trials. This likely gave a fast response about the type of as-



**Figure 20:** NASA-TLX subjective workload ratings for each scale for the second experiment, with and without (benchmark condition) the aid of the controller. Box plots based on data from 8 participants ( $N = 8$ ). The width of each box represents the mean weight given to the respective scale by the participants. The median of *PD* for the benchmark condition was 17.5, with the controller 24. The upper whisker represents the most extreme data point below the limit: 1.5 times the interquartile range beyond the third quartile. And similarly for the lower whisker and the first quartile. Scales: *EF* - Effort, *FR* - Frustration, *MD* - Mental Demand, *PD* - Physical Demand, *PE* - Performance, *TD* - Temporal Demand.

sistance the system could provide. That is, they typically moved the robot arm at a velocity that was slightly above what they could normally control. The collisions experienced led to increased NN weights specific to where the collisions were sensed, and the operator would then feel an increased reduction of the commanded velocity in the direction of obstacles in similar situations. After the initial increase most participants then had smaller adjustments to their weights, with participant 1 and 7 choosing to freeze the weights before ending the 20 attempts of the session.

It can be seen that some collisions did not noticeably increase the NN weights, for example in trial 6 for participant 1. This would typically indicate that there was an insufficient coverage of proximity sensors for that location, a limitation of the sparse sensing used. It should also be noted that the different tasks typically provoked very different levels of assistance for a given participant. Finally, it is interesting that two participants with very different NN weights, participant 1 and 2, were both in the top 3 with respect to performance for the experiment (completion time). This indicates that it is possible to use quite different strategies for taking advantage of the assistance. The range of strategies can likely be explored more quickly because the robot adapts in real-time.

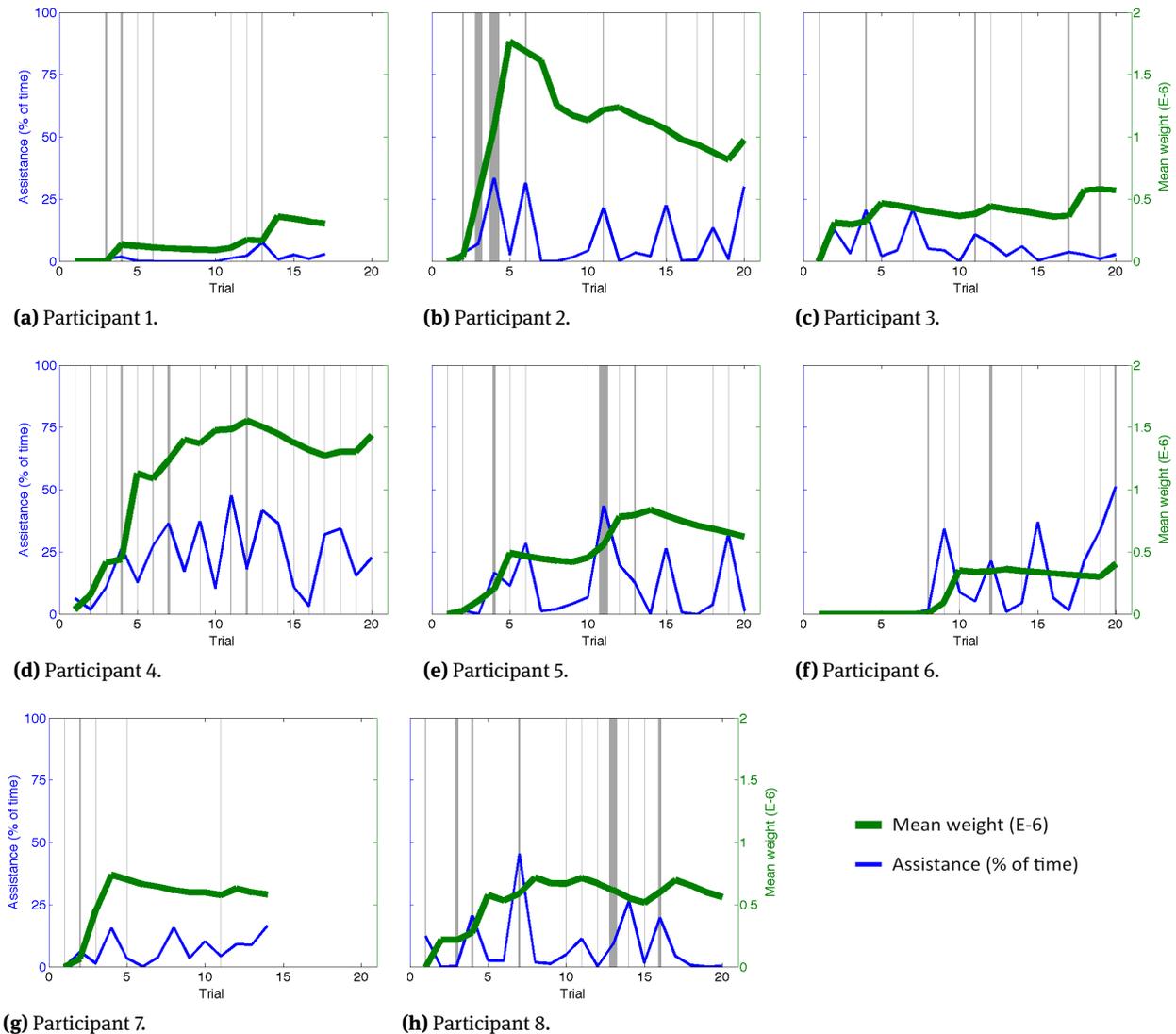
## 6 Towards a Physical Implementation

This section will outline the current progress on implementing the controller on a real assistive manipulator, the AMOR robot of Exact Dynamics in Holland. It is hoped that the work outlined can lead to practical systems that can adapt to, and assist, the user of similar physical robots in the near future. A first implementation of distributed proximity sensing on the hand can be seen in Fig. 22, based on infrared Silicon Labs Si1143 sensors. Each sensor has its own digital circuit, and the sensors readings can be accessed over an I2C bus. A great advantage of these sensors is that they work from approx. 5 cm to 40 cm. However, as with other distance sensors based on infrared light, they are noisy when used on black, transparent or shiny surfaces. Redundancy could be achieved through sensors based on other physical modalities, like sound waves.

Fig. 23 shows the same type of proximity sensors on the body of the AMOR, with integration well underway. The current plan is to have local information-gathering on each link of the robot using an Arduino Nano board. Each board will communicate the readings to a central controller over wires or Bluetooth. Initial compatibility trials with the PMD Nano and the Microsoft Kinect 3D sensors is also under way. These sensors also use infrared light around 850 nm, but with spatial modulation that should not significantly interfere with the temporal modulation of the distance sensors. The current results do indeed indicate that there is little or no interference between the different sensors. More testing is needed to confirm this during usage on the robot platform. Combining static, hand-mounted and distributed infrared sensing would increase robustness and provide redundancy in case of occlusions and sensor malfunction. The current implementation does not yet include a tactile skin, but there are several promising technological alternatives under development, and in use. For example for full-body tactile sensing on humanoid robots [35].

## 7 General Discussion

The neuronal units of the NN used here are linear and the networks do not have hidden layers. The term "neural network" is still used here, but it is clear that the complexity of the learning algorithm is on the extreme low end of typical NNs. This low algorithmic complexity is made possible by the extremely simple, and task relevant, informa-

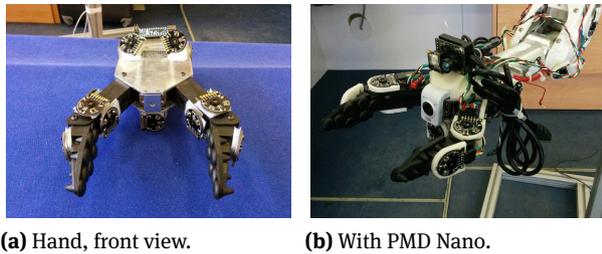


**Figure 21:** Visualization of the adaptation by the system (and the operator) during the session with learning in the controller. Mean values for each trial are shown. The mean weights of the neural network for the last link are drawn in a thick green line. Starting from zero, the weights grow with collisions with the environment, and decay slowly otherwise. Collisions are indicated by vertical grey bars, with the thickness signifying the summed estimated force of the collisions for that trial. Thin blue lines indicate the percentage of time during each trial that the participant was given assistance by the system. That is, how often the system reduced the Cartesian velocity commanded based on the learned proximity sensor usage (when  $r_{max} > 1$ ).

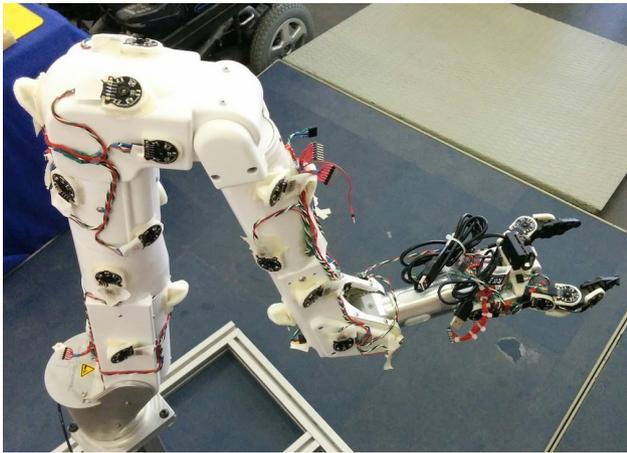
tion coming from the robot's main sensory apparatus, the distributed collision and proximity sensing. The simplicity also influences the scalability of the approach, which can easily scale to 1000s of sensors without compromising the ability to learn and provide aid in real-time. More traditional approaches, like predictive displays, employ holistic whole-scene sensors, combined with geometric models of the robot and its environment. The approach presented is complementary rather than in competition, and can help reduce the need for maintaining exact models of every aspect of the world. For example by seeing "behind"

the arm, where you otherwise would have to rely on sensor data stored previously.

The two experiments presented used able-bodied participants (undergraduate and graduate students of both sexes) as the robot teleoperators. We do believe that actual end-users should be included as participants in experiments on assistive technology in general. However, this study examined a controller to aid the tele-operation of an assistive robot by an able-bodied operator from a distance. The operator is in this case meant to help the disabled or elderly user control the robot on especially difficult tasks.



**Figure 22:** Development of distributed proximity sensing for the AMOR hand, based on infrared sensors (Silicon Labs Si1143) and a PMD Nano depth camera. In-hand web camera also seen.



**Figure 23:** Overview of the current implementation of proximity sensors on the body of the AMOR robot manipulator. Data processing and communications units not shown.

The operator can be a professional, a care provider, or a family member. The able-bodied participants used seem a reasonable choice for this group of operators, but it would be interesting to include more mature participants and actual health care providers in future studies. The experiments used also simulated the round-trip delays caused by teleoperation over an internet connection. The exact mean and variability of such delays would differ greatly from location to location, but the quite distinct delays used here should represent an interesting subset of actual delays. This should give an indication that the system can perform under a range of different conditions. As a comparison, Rodríguez-Seda, Lee and Spong [16] included one-way time-varying delays with means ranging from 80 ms to 480 ms, Xiu et al. [14] approximated variable one-way delays of around 300 ms, while Davis, Smyth and McDowell [12] used a variable round-trip delay with a mean of 700 ms. However, future studies should include tele-operation over physical distances and real internet connections.

A somewhat controversial aspect about the approach presented is the need for collisions with the environment

for the system to learn to aid the operator. First, most current assistive devices are made with the idea that collisions are not acceptable. This makes perfect sense for robots with rigid structures, position/velocity control, and hard outer surfaces. However, the authors feel that physically assistive robots will necessarily need to become "softer" to be safe enough for use at home with elderly and disabled users, for example by incorporating passive compliance in the structure, which can make collisions much less risky. In fact, manipulators with passive compliance, like the human arm, can benefit from the physical interaction with the environment to simplify the control [36]. The work may be quite suitable for "softer" robot arms, given the real-time adjustment of the sensor usage, and the limited need for exact arm models. Second, most current assistive robots move very slowly. Partially because of the need to avoid collisions, and partially because the devices are challenging to control, especially multi-DOF manipulators. A good goal for such manipulators, and for prosthetics, must be to approach the performance of the human arm. In this context, safety, which typically means avoiding dangerous collisions, cannot easily be separated from movement speed, as targeted arm movements are strongly driven by the speed-accuracy trade-off. The study presented here attempts to take this into account on two levels: 1) through performance metrics, as the participants have to keep in mind the time lost by collisions, and 2) through the limitation of velocity in situations similar to those where collisions were experienced in the past, rather than an active collision avoidance system.

Nonetheless, the training of the system is not a fully resolved issue. A coarse tuning of the weights of the system could perhaps be done in simulation before moving to the real-world system. Exactly when to learn and when to freeze the weights in a real situation is also not clear. More research is needed to resolve these issues. Similarly with the plain dot-product used, which means local velocities that are perpendicular to the virtual proximity reading will not have an effect. This could be altered by the use of the  $\alpha_{proj}$  and  $\beta_{proj}$  parameters (see Algorithm 1), to for example slow the system down by a certain degree in all directions when close to an obstacle. The algorithm used for allowing the link-specific NNs to influence the movement of the whole manipulator is also quite restrictive. It can, for example, limit wrist motion by sensing proximity to an obstacle (previously learned) near the shoulder. This could likely be relaxed, for example if the arm is kinematically redundant.

Perhaps the main limitation of the work is that the learned assistance typically works best in the given scenario where the learning was performed, for example the

refrigerator scenario used here. The second experiment showed that the system can still provide aid on tasks that are similar to those on which it was trained. However, if the system is to be used also in other scenarios, for example when the user is at the office, a mechanism for switching between sets of learned weights is likely needed. Given the small size of each weight matrix, a large number of such sets can easily be stored. Task-oriented approaches have also been used for adapting the physical structure of assistive robot manipulators to the user's needs [31, 32]. One of the advantages of the system proposed here is that it should scale well also to much higher density proximity sampling, for example in approaches based on electric fields [33, 34]. It is also interesting to note that a behavioural approach that from an algorithmic point of view looks very simple, can provide quantitative and qualitative improvements in performance on a complex human-robot system.

The results should be seen in relation with the previous studies that applied a similar approach to direct control by users with simulated disabilities [5, 6]. If the benefits are confirmed with real end-users the same physical system could one day assist both on direct control by the user and on tele-assistance by a remote operator on more difficult tasks.

## 8 Conclusions and Future Work

An approach for providing adaptive aid in a tele-assistance scenario was presented. The disabled or elderly user of an assistive robot manipulator is here helped by a remote operator on tasks that are too complex to perform by direct control. The approach assumes a set of tasks that are regularly performed, on which the system can learn from the collisions with the environment. The learning adapts a proximity-based haptic aid for the remote operator. The adaptation occurs online during operation, by associating a set of distributed infrared proximity sensors to a coarse tactile skin. By using local sensor information to filter the received commands, the system can help mitigate some of the negative effects of the variable time-delays in the communication with the operator. Promising improvements in the time to complete tasks and the subjective temporal demand were found in two controlled virtual experiments with 9 and 8 participants.

Future work is needed to assess the performance on a larger set of tasks with more diversity, and on the physical implementation described here. It would also be interest-

ing to test a similar system under more severe time-delays, including those found in space teleoperation.

**Acknowledgement:** The research presented here has received funding from the ARCADIA project DPI2010-21047-CO2-0, 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, C. Balaguer, A portable light-weight climbing robot for personal assistance applications, *Industrial Robot: An International Journal*, 33 (2006), no. 4, 303-307
- [2] D.J. Kim, et al., How Autonomy Impacts Performance and Satisfaction: Results From a Study With Spinal Cord Injured Subjects Using an Assistive Robot, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 42 (2012), no. 1, 2-14
- [3] H. Jiang, T. Zhang, J.P. Wachs, B.S. Duerstock, Autonomous Performance of Multistep Activities with a Wheelchair Mounted Robotic Manipulator Using Body Dependent Positioning, *Workshop on Assistive Robotics for Individuals with Disabilities: HRI Issues and Beyond, IEEE/RSJ Int. Conf. on Intell. Robots and Systems*, 2014
- [4] D.J. Kim, Z. Wang, N. Paperno, A. Behal, System Design and Implementation of UCF-MANUS - An Intelligent Assistive Robotic Manipulator, *IEEE/ASME Transactions on Mechatronics*, 19 (2014), no. 1, 225-237
- [5] 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, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2012, Vilamoura, Portugal)*, IEEE, 2012, 4386-4391
- [6] M.F. Stoelen, V.F. Tejada, J.G. Victores, A. Jardón, F. Bonsignorio and C. Balaguer, Adaptive Collision-Limitation Behavior for an Assistive Manipulator, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2013, Tokyo, Japan)*, IEEE, 2013, 1143-1148
- [7] T.L. Chen et al., Robots for humanity: using assistive robotics to empower people with disabilities, *IEEE Robot. Automat. Mag.*, 20 (2013), no. 1, 30-39
- [8] C. Passenberg, A. Peer, M. Buss, A survey of environment-, operator-, and task-adapted controllers for teleoperation systems, *Mechatronics*, 20 (2010), no. 7, 787-801
- [9] P.M. Fitts, The Information Capacity of the human Motor System in Controlling the Amplitude of Movement, *Journal of Experimental Psychology*, 47 (1954), 381-391
- [10] J.T. Dennerlein, D.B. Martin, C. Hasser, Force-feedback improves performance for steering and combined steering-targeting tasks, *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI'00 (2000, The Hague, The Netherlands)*, vol. 2, no. 1, 423-429

- [11] J.C. Lane, C.R. Carignan, D.L. Akin, Advanced Operator Interface Design for Complex Space Telerobots, *Autonomous Robots*, 11 (2001), 49-58
- [12] J. Davis, C. Smyth, K. McDowell, The Effects of Time Lag on Driving Performance and a Possible Mitigation, *IEEE Transactions on Robotics*, 6 (2010), no. 3, 590-593
- [13] C. Basdogan, M.A. Srinivasan, Haptic Rendering in Virtual Environments. In K. Stanney (Ed.), *Virtual Environments Handbook*, Lawrence Erlbaum Associates, 2001, 117-134
- [14] Z. Xiu, A. Kitagawa, H. Tsukagoshi, C. Liu, M. Ido, P. Wu, Internet-based Tele-rehabilitation System - Bilateral Tele-control with Variable Time Delay, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2006, Beijing, China)*, IEEE, 2006, 5208-5213
- [15] C.R. Carignan, H.I. Krebs, Telerehabilitation robotics: Bright lights, big future?, *Journal of Rehabilitation Research & Development*, 43 (2006), no. 5, 695-710
- [16] E.J. Rodríguez-Seda, D. Lee, M.W. Spong, Experimental Comparison Study of Control Architectures for Bilateral Teleoperators, *IEEE Transactions on Robotics*, 25 (2009), no. 6, 1304-1318
- [17] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, H. Brussel, M. Nuttin, User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving, *Journal of Autonomous Robots*, 24 (2008), 193-211
- [18] T. Carlson, Y. Demiris, Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload, *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42 (2012), no. 3, 876-888
- [19] S.E. Everett, R.V. Dubey, Human-Machine Cooperative Telerobotics Using Uncertain Sensor or Model Data, *Proceedings of IEEE International Conference on Robotics and Automation (1998, Leuven, Belgium)*, IEEE, 1998, 1615-1622
- [20] D.A. Abbink, M. Mulder, E.R. Boer, Haptic shared control: smoothly shifting control authority?, *Cognition, Technology & Work*, 14 (2012), 19-28
- [21] P.F.M.J. Verschure, B.J.A. Krose, R. Pfeifer, Distributed adaptive control: The self-organization of structured behavior, *Robotics and Autonomous Systems*, 9 (1993), no. 3, 181-196
- [22] P.F.M.J. Verschure, T. Voegtlin, R.J. Douglas, Environmentally mediated synergy between perception and behaviour in mobile robots, *Nature*, 425 (2003), no. 6958, 620-624
- [23] R. Pfeifer, J. Bongard, *How the body shapes the way we think*, MIT Press, Cambridge, MA, USA, 2006
- [24] I. Pavlov, *Conditional Reflexes*, Dover Publications, New York, NY, USA, 1927/1960
- [25] H. Yousef, M. Boukallel, K. Althoefer, Tactile sensing for dexterous in-hand manipulation in robotics - A review, *Sensors and Actuators A: Physical*, 167 (2011), no. 2, 171-187
- [26] E. Cheung, V.J. Lumelsky, Proximity sensing in robot manipulator motion planning: system and implementation issues, *IEEE Transactions on Robotics and Automation*, 5 (1989), no. 5, 740-751
- [27] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, A.Y. Ng, Reactive grasping using optical proximity sensors, *Proceedings of IEEE International Conference on Robotics and Automation (2009, Kobe, Japan)*, IEEE, 2009, 2098-2105
- [28] S.G. Hart, L.E. Staveland, Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research, *Advances in psychology*, 52 (1988), 139-183
- [29] A. Kiselev, A. Loutfi, Using a Mental Workload Index as a Measure of Usability of a User Interface for Social Robotic Telepresence, in *Proceedings of the Ro-Man 2012 Workshop in Social Robotic Telepresence*, 2012, 3-6
- [30] R. Diankov, Automated Construction of Robotics Manipulation Programs, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2010
- [31] J.Y. Park, P.H. Chang, J.Y. Yang, Task-oriented design of robot kinematics using the Grid Method, *Advanced Robotics*, 17 (2003), no. 9, 879-907
- [32] A. Jardón, M.F. Stoelen, F. Bonsignorio, C. Balaguer, Task-Oriented Kinematic Design of a Symmetric Assistive Climbing Robot, *IEEE Transactions on Robotics*, 27 (2011), no. 6, 1132-1137
- [33] J.R. Smith et al., Electric field imaging pretouch for robotic graspers, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (2007, San Diego, CA, USA)*, IEEE, 2007, 676-683
- [34] S.E. Navarro et al., Methods for safe human-robot-interaction using capacitive tactile proximity sensors, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2013, Tokyo, Japan)*, IEEE, 2013, 1149-1154
- [35] G. Cannata, M. Maggiali, G. Metta, G. Sandini, An Embedded Artificial Skin for Humanoid Robots, *Proc. of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (2008, Seoul, Korea)*, IEEE, 2008.
- [36] R. Pfeifer, M. Lungarella, F. Iida, Self-organization, embodiment, and biologically inspired robotics, *Science*, 318 (2007), no. 5853, 1088-1093