

# INTUITIVE USER INTERFACES FOR MOBILE MANIPULATION TASKS

Submitted: 17<sup>th</sup> November 2014; accepted: 8<sup>th</sup> January 2015

Igor Zubrycki, Grzegorz Granosik

DOI: 10.1431/JAMRIS\_1-2015/6

## Abstract:

*This article describes interactive methods that can ease difficult manipulation tasks in Search & Rescue operations. We discuss the requirements that are necessary for a telemanipulation system to be successfully used. These include not just correctness of generated motion but also ergonomics, mobility and interactivity of the operator's interface. We show that grippers with one or more degrees of freedom can be intuitively controlled by different interface mechanisms, supported by 3D vision systems. Tests are performed both in the simulation environment and with real grippers. A practical pipeline for a direct control and learning the system is also presented.*

**Keywords:** *Intefraces, Human-Robot Interaction, ROS, Vision Systems*

## 1. Introduction

Successful gripping and manipulation is a key ability in exploratory and rescue tasks. Such operations require cooperation between people and robots, thereby giving the operator the ability to work in dangerous and/or inaccessible environments where experience, cognitive skills and decision making ability is needed [35].

However, telemanipulation in rescue or exploration tasks can be very complicated and stressful for operators. Robots are usually used in places where direct human activity is difficult (traversing through rubble), dangerous (inside a nuclear reactor) or impossible (outer space). Additionally, the operator's cognitive ability is limited by the range and modalities of a robot's sensors and transmission bandwidth. For the successful telemanipulation in exploratory or rescue tasks, the robot's control system must be equipped with a number of properties ranging from features related to the nature of the task; through to human psychology, preferences and skills; and up to ergonomics. We have compiled a list of the most important features – collected into the three categories described below. Although these requirements concern the whole control system, in this article we focus only on the task of the gripper control.

**Certainty and confidence.** Those operators who trust their control interfaces can use their robots with higher speeds and in more aggressive maneuvers which results in a much faster realization of the task. It also lessens operators' fatigue because they do not need to always add a safety margin in their decision making. Instead of focusing on basic robotic be-

haviours, they can work at the higher level of task abstraction, such as planning of the whole route.

By contrast, a low level of trust, especially in a stressful situation could result in failure of the whole mission. For example, the first robot used in exploring the Fukushima reactor – Quince – was immobilised when the operator became distrustful of the robot's automated path planning algorithm and switched to manual control mode, allowing the robot to become entangled in its own power tether [38].

Robot designers can increase confidence through well chosen and clearly presented feedback information about the state of the mission (i.e. gripping stage, encountered problems, expected problems), and collision prediction and prevention.

**Limited training.** Currently, mobile field robots are mostly used by teams of experts for very specialised tasks such as bomb diffusion or deep sea exploration. This, by itself, requires an enormous proficiency, experience and skills. Robots could be used in a much wider range of exploratory and rescue tasks. This would, however, require a simple interface so that the training was unnecessary or restricted to a minimum. Sometimes in rescue situations, it is necessary that the equipment is run by untrained people who happen to be available at the time. This might be a case of mass usage of robots—possibly controlled by amateurs. For example, during the Fukushima catastrophe, robots were controlled by previously untrained workers of the power plant who had to learn on the go [28, 38].

**Ergonomics, feasibility and cost.** There are systems today that provide operators with full immersion – through virtual reality (VR) and force/tactile feedback. However, these systems are expensive and hard to use in real rescue scenarios where the operator is also involved in other mission tasks. In such a situation, the operator must be able to modulate his or her focus [5].

As rescue and exploration tasks usually take a long time, users should not be forced to perform unnatural movements (e.g. very small and precise or very large) too frequently or to keep an uncomfortable posture.

In the rescue scenario, an operator usually wants to be near the centre of the action so the control interface should be mobile and robust. Operators may have to work in very confined spaces. Long term costs of the interface should also be minimised which suggests using commercial off-the-shelf (COTS) elements that do not wear out easily and are easily repaired or replaced.

The interface should give the operator the ability

to fully use the gripper-manipulator capabilities. The robot should be controllable in its full workspace with full agility and without noticeable delays.

The above assumptions are realised to varying degrees by currently used gripper and control systems of rescue-exploratory robots. The next two sections present some state-of-the-art solutions while section 4 explains how our proposals address our assumptions.

## 2. Grippers Used in Mobile Control

Grippers used in mobile manipulation work in a completely different environment to industrial grippers and, therefore, they have to be universal, impact resistant and robust. Some typical examples are shown in Fig. 1.

Grippers with one degree of freedom (1 DOF) have some indisputable advantages: low price, availability, ease of control, robustness and strength. However, universal 1 DOF grippers cannot manipulate objects in complex ways while dedicated 1 DOF grippers have limited task range. Therefore, a manipulator needs to provide all necessary degrees of freedom to perform the task and the control interface has to be prepared accordingly. In our opinion, the best solution would be treating both devices as an integrated system – the single controller should steer both the gripper and manipulator.

Underactuated grippers such as Velo (shown in Fig. 1b), Barret Hand [40], or FetchHand have a larger number of joints than actuators, and use tendons and springs in the drive system. These mechanical linkages make the gripper's fingers wrap around an object, making the grip more stable through a more even force distribution. Because of mechanical fit to the object's shape (even without a precise knowledge of contact forces), gripping is easier, especially when the object's shape is not known in advance.

Ability to entwine an object is usually limited to specific sizes of objects, otherwise the grips are sub-optimal. With some types of grips this is not possible at all. Therefore, the operator must be aware of the gripper's limitations. This is compensated by a comparably lower price and a less complicated control system than in the case of fully actuated, dexterous grippers.

Dexterous, multi-finger grippers give operators the widest control of the gripping process. They have several degrees of freedom, ranging from the 7 DOF – Schunk SDH and 12 DOF – NASA Robonaut (shown in Fig. 1c) to the 13 DOF – DLR/HIT Hand-2 [16, 20]. A dexterous gripper can be thought of as a group of manipulators working in a shared workspace. Their controllers are fairly advanced, being built with FPGA units and DSP processors, and having CAN, USB, Ethernet or RS232 communication. To effectively exploit their actuation capabilities, they have a range of proprioceptive sensors: temperature, position, torque and a range of exteroceptive sensors: tactile matrices, vision systems and distance sensors [4, 11, 16, 20].

Designers use a variety of approaches choosing the shape of grippers and the distribution of drives.

Designers use a variety of approaches, choosing the shape of the grippers and the distribution of drives. Finger drive motors can be located inside fingers and are usually accompanied by high ratio gears. When drives are outside the fingers (e.g. in a forearm), forces are transferred through the tendons, as in Shadow Hand [7]. Dexterous grippers or hands can be anthropomorphic – their proportions, number of fingers and placement would be similar to a human's or sometimes not. In this case, we usually find three fingers, placed around the palm.

The dexterity of these artificial hands gives the operator the ability for inter-finger object manipulations – regrasping, finger gaiting, in-grasp manipulation, rolling and sliding. They also give additional agility which may be necessary to grasp objects when there is some obstacle in the manipulator's workspace [23]. They also require the most advanced methods of control because fingers must be controlled concurrently in a coordinated fashion.

In our research and experiments we have used the Schunk Dexterous Hand 2 (SDH-2). It has non-anthropomorphic structure with 7 DOF. Each of three identical fingers has two joints actuated by BLDC motors with harmonic drives. An additional motor with a worm gear located in the palm actuates symmetrical rotation of two fingers.

## 3. Current Control Interfaces

Joysticks, radio remote controls, teach pendants and gamepads are the most frequently used devices to control exploratory and rescue robotics. Most of them originate either from RC community where they are used to control model vehicles or from gaming community used in a wide range of computer games. Such controllers have various buttons, levers, and sticks providing a robust control of large number of functions. Most advanced versions of RC controllers have bi-directional data flow with telemetry functions [13] and are available in robust, anti-shock housings and are capable of controlling large industrial machinery such as cranes or telehandlers [2]. Gaming controllers such as PlayStation Controller or Wii Remote are also widely used for mobile robot control because of their ergonomics and familiarity as many users, even from military, are experienced in computer gaming [5]. Special consoles for teleoperation tasks with mobile robots are also available, giving users, in addition to buttons and joysticks, large screen for display of camera images and vehicle information [39].

Teleoperation of manipulators and grippers by moving various axes (or modifying the Cartesian coordinates) was considered even in the first telemanipulation experiments (handling radioactive materials) as slow and awkward, leading to the design of linked master-slave robots and haptic controllers [36, Ch. 31, p. 782].

In master-slave configurations the operator directly, through movement of his hand holding the controller's handle, controls orientation and position of the manipulator's gripper or tool. If the manipula-



(a) one degree of freedom gripper on Foster-Miller TALON [31]

(b) underactuated, Velo gripper from Willow Garage [1]

(c) dexterous, multifinger gripper of NASA's Robonaut [30]

**Fig. 1. Various gripper types used in exploratory and rescue robotics**

tor is equipped with torque/force sensors a bilateral control and force feedback can be incorporated. Modern, commercially available haptic controllers such as Omega devices from Force Dimension provide the bi-directional control with high number of degrees of freedom (7 in case of the Omega 7). In mobile manipulation they have been used for tasks ranging from telesurgery via unmanned aerial vehicle to teleoperation of multiple robots [22, 37].

While master-slave controllers are widely used in telesurgery, they are less popular in mobile manipulation scenarios. Main reasons are: the cost (e.g., Omega devices cost more than 20 000 USD, more affordable Novint Falcon around 200 USD but offer only 3 DOF) and size of the workspace limited by kinematic structure of the controller [12, 32].

Many modern robots, can be controlled through a graphical user interface on the computer. The operator can see the robot with its environment and move the robot using a mouse in various ways – controlling each joint with sliders and moving robot in inverse kinematics mode, or semi-autonomously by indicating surfaces to grab [19]. For control of dexterous grippers by computer mouse input, dimensionality reduction can be implemented for grasp planning [6].

The most advanced option is the master-slave control with virtual reality, which can be particularly effective when there is haptic feedback provided to the user. VR immerses operator in robot's environment. This type of systems are used to control humanoid robots (Robonaut [16], Tops [35]), because joint mapping, in this case, can be done one to one.

#### 4. Proposed Interfaces

We have developed several interfaces that are based on integrated vision systems such as Kinect and Leap Motion. Their APIs allow for precise tracking of the position and orientation of the human's hand or other objects.

We have focused our research on tools to support an intuitive and precise approaching and gripping an object. As grippers of different kind require different interfaces, so that they could be fully utilized, we have designed different interfaces for grippers with one degree of freedom and different for dexterous grippers with multiple degrees of freedom.

Our goals in designing interfaces described in subsections below were directly connected to postulates in section 1, in particular:

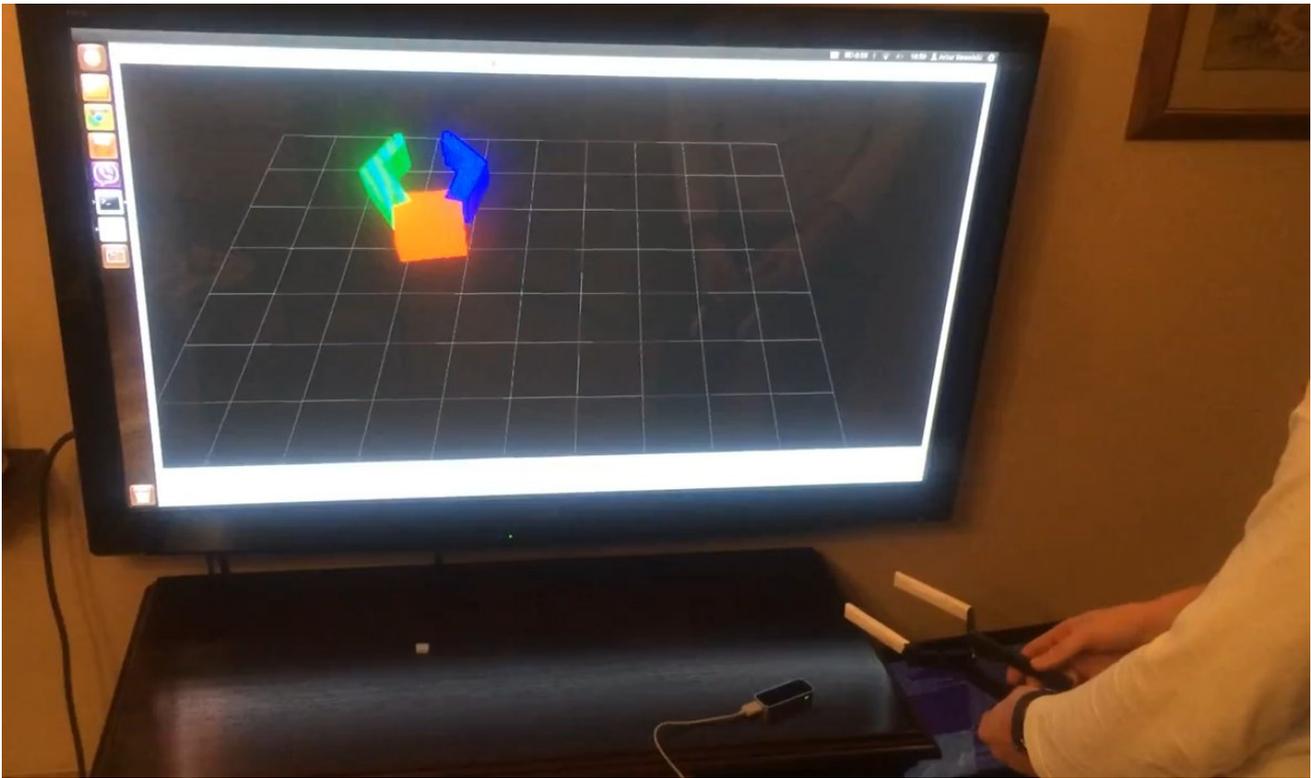
- 1) An untrained user should be able to pick up an object such as a tool using our interface
- 2) A grip should be of good quality, i.e. stable and adequate to the task
- 3) Interface should be of a marginal cost compared to a cost of mobile robot
- 4) Interface could be used in an environment similar to Fukushima scenario, i.e. cramped room or outside
- 5) Operator should be able to move his attention away from the task
- 6) Operator's work environment should allow him for multi hour work

##### 4.1. Visual Gripper

Visual gripper is a software tool for controlling position, orientation and gripper opening-closing. Operator uses a mockup gripper that is tracked by the vision system to control a real gripper placed on a manipulator (Fig. 2).

Our solution exploits Leap Motion's ability to track "tools" defined as elongated, straight objects. For each such "tool" API provides its position and a direction vector with sub-millimeter precision.

For testing this interface we have constructed a simple prop mechanism, that corresponded geometrically (though not visually) to the actual gripper, and that would be easily detectable by the vision system.



**Fig. 2. Visual Gripper controlling gripper's model through mechanical gripper mockup with 1 DOF [18]**

For such kinematic structure, imitating gripper with 1 DOF (see Fig. 3), Leap Motion gives coordinates of points  $O_1$  and  $O_2$ , and respective direction vectors  $\mathbf{z}_1$  and  $\mathbf{z}_2$ . Based on these information, we can calculate length of vector  $\|\mathbf{a}\|$  and using previously measured dimensions of a mock up mechanism we can calculate the angle of jaw opening  $\beta$ .

Position and orientation of the mechanism can be unambiguously determined using two perpendicular vectors:  $\mathbf{z}_1$  and  $\mathbf{x}_1 = (\mathbf{z}_1 \times \mathbf{a}) / \|\mathbf{z}_1 \times \mathbf{a}\|$  and coordinates of the point  $O_1$ . Homogeneous transformation matrix from sensor's coordinate frame to the frame located in the point  $O_1$  can be described by (1). Similar calculations can be made for the corresponding point on the second jaw of the mechanism. Additionally, knowing the angle  $\beta$ , we can determine the position and orientation of the center of mechanism  $O_3$  in the sensor's coordinate frame. Therefore, we obtained the fully functional interface with 7 degrees of freedom, to possibly control any manipulator with a gripper.

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{z}_1 \times \mathbf{x}_1 & \mathbf{z}_1 & O_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

We have implemented "Visual Gripper" in the ROS. Program, as one of ROS nodes, retrieves sensor data from Leap Motion (orientation and position of tracked elements), realises previously described algorithm and formats data in a way that can be visualised in Rviz (one of the ROS tools). In the current implementation the gripper's model moves freely in space as shown in Fig. 2, but the same data can be used to solve inverse kinematics for a specific manipulator holding the grip-

per. Video demonstration of the interface can be accessed at [48].

Using a prop input devices is an idea with a long history in human computer interaction, especially in gaming with examples such as lightgun [3] or prop stick [24]. In area of robotics various haptic interfaces used prop tools (such as prop pen in Phantom Omni haptic controller) for more precise control.

Integrated vision systems have been used to control different mobile manipulators with grippers. Kinect has been used to teleoperate upper body limbs of Nao robot with Wii Remote controlling jaw opening of nao hands [42]. Leap Motion ver. 2 can be used for gripper-manipulator pair control, because its API provides direct information about human's hand opening [26]. A two-hand control interface for manipulator with 3-fingers gripper has been developed by Gibaru et al. [14].

Telemanipulation using props meets well our specifications, in case of 1 DOF gripper. Users can easily manipulate position and orientation just by moving prop. Gripper affordances, its limits on gripper opening and its controls are also clear. This reduces need for training and gives users trust in what the system will be doing.

Our interface is also extremely affordable. Controller itself requires only a Leap Motion sensor (less than 100 USD) and a prop gripper – which can be made from some LEGO parts or 3D printed. This is not the case with full mechanical master controllers. Compared to game controllers it provides a much better conceptual model of gripper control and more flexibility.

Finally, interface can easily be used in field opera-

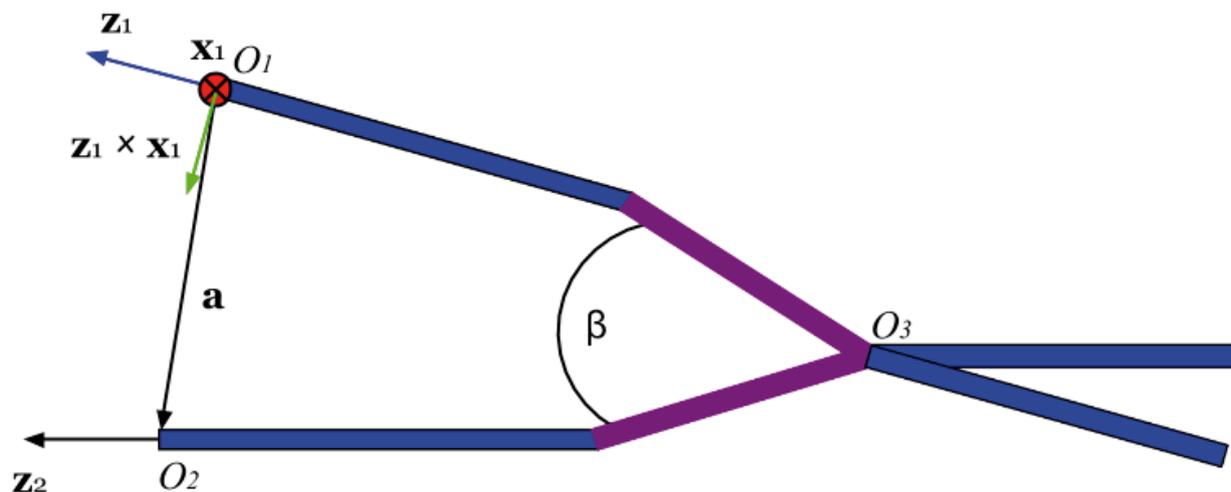


Fig. 3. A diagram of the mechanical gripper mockup with characteristic elements

tions. The operator needs only to access a laptop computer for visualisation and a 50 cm cube free space to operate the prop.

#### 4.2. Multi-finger Gripper Control – Hybrid Controller

Designing a gripper control interface for multi-finger, dexterous grippers using only vision systems is a very challenging task, due to such phenomena as self-occlusion [49]. However, integrated vision systems are still extremely useful – as they can very precisely track position and orientation of objects in 3D as well as give information about users movements and gestures. In our study, we have achieved best results using hybrid approach, where readings from the sensor glove were supplemented with information from the vision system (Leap Motion or Three Gears).

Our SDH gripper control system uses two sources of information: readings from flexion sensors mounted in the sensor glove and data from the vision system. Rotation angles of gripper's fingers are controlled by flexion of operator's fingers while the position and orientation of entire Schunk Hand and the mode of operation can be changed based on vision sensor. The block diagram of the proposed control system is shown in Fig. 4, while a detailed description of its various functions is given in following subsections.

#### 4.3. Related Work

Mapping between human hand and robotic hand can be done in several ways. They can be divided to a few types: joint-to-joint mappings, fingertip mappings, mappings through virtual objects and pose mappings.

Joint to joint mappings can be efficiently used when the kinematic structures of human hand and robot gripper are similar. In such case joints of the human hand are directly associated to joints of the robot gripper. This is usually possible only when robot hand structure is anthropomorphic [35]. In case of differ-

ent number of joints a dimensionality reduction can be used [6].

Fingertip mapping assumes that although the hand structure can be different operator wants to have fingertips of his own hand and that of the gripper's fingers in the same position (or scaled proportionately) – so that he could grasp an object (this approach assumes fingertips as contact points). In case of different number of fingers a notion of virtual fingers can be used [21, 34].

Mapping through the virtual objects algorithm matches the mappings in the way that similar objects (sphere, cylinder) could be grasped using human hand and a gripper. In Griffin et. al [17] authors assume that object is held by thumb and the index finger. Algorithm then recreates the same relations between such virtual object and gripper's fingers (with workspace matching). Gioioso et. al, extend virtual sphere method with the idea of synergies [15], using *paradigmatic hand's synergies* as an input vector [15].

Pose mapping methods try to correlate human hand poses and robot hand poses through the recognition of grip. Authors of [43] use neural networks for recognition and fixed joint mappings after each grip is recognised. In [9] model based approach using Hidden Markov Models is used, where grasp is classified through characteristic dynamic changes in fingertip positions or arm movement trajectories.

#### 4.4. Controlling Dexterous Gripper Through Grip Recognition

Humans have excellent, "built in" ability to use their hands in manipulation and can optimally choose grasps subject to task, object and their own hands constraints. In our system we want operators to use their abilities even if the gripper they are controlling is of different structure than human hand.

A different structure of the gripper can mean that a similar task will require a different pose, but with the similar properties such as compliance, force or form

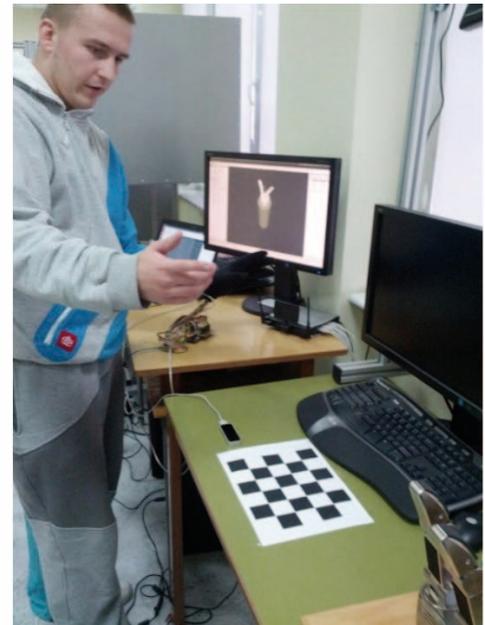
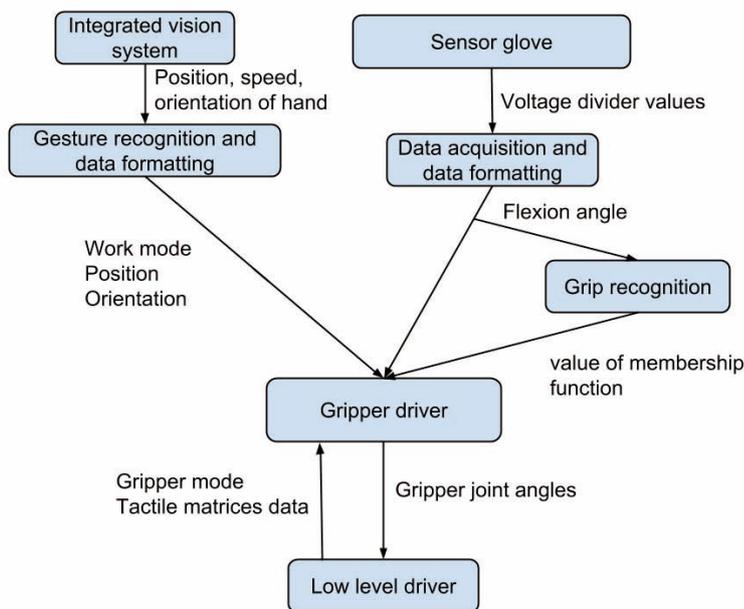


Fig. 4. Diagram of the hybrid controller and photograph of the test stand

closure, or manipulability [8] to that of human's hand.

According to Napier [29] grip – posture of hand chosen for the grasp is mainly influenced by goal of the task, while size or shape of objects are used by humans only to tune the chosen pattern. This means that through recognizing grips we will be able to recognize the intent of user and use this information to provide accurate mapping function. Through this function changes in users posture will still have an effect on position of gripper joints.

#### 4.5. Mapping Function Between 5 Fingers of the Human Hand and Three Fingers of SDH

Controlling robot joints with the sensor glove seems to be very attractive – it offers an intuitive operation and uses natural human's ability to precisely manipulate with own hand. This type of control is often used when controlling anthropomorphic grippers with similar structure to human's kinematics (eg. Robonaut hand); then the simple master-slave (1:1) mapping can be applied. With grippers of a different structure and number of fingers, such as SDH, a problem of mapping is more complicated.

The SDH has 7 degrees of freedom, therefore, human hand has enough agility to control it through some movements, but its kinematics is different than human's hand (except the number of the fingers, they have different flexion ranges, and human cannot rotate fingers around the base of the palm). First of our approaches assumed arbitrary choice of fingers to further directly control gripper's movement [45]. In our second approach, we propose here the mapping based on recognizing the user's intention. It is still based on a detection of grip types, but then the obtained information is blended to a single behavior using some characteristic features rather than the strict classification.

The preliminary tests with recognising several specific grip types showed a problem when a hand pose was classified into two grips with the similar

certainty. Particularly, precision grip had been mistaken as lateral and wedge grips [44]. In such case, some rapid and frequent switching between gripper poses could be observed, which is highly unintuitive and frustrating to the user.

The solution comes with the second part of our proposal: blending the mapping. Instead of the discrete switching to the grip type that has the biggest value of the classification function in the moment, we calculate finger movements using a weighted sum of mapping matrices with the algorithm described below.

Sensor data acquired from the sensor glove (namely readings from flex sensors) are filtered to obtain vector  $\mathbf{v}$  (size 10x1). The Support Vector Classifier generates a new vector with values of membership function of each  $\mathbf{v}$  to each class [44]. Which means that for  $T = 7$  considered grips, we calculate for each  $k = 1..T$ , according to equation (2), a distance from a point with coordinates  $\mathbf{v}$  to the decision hyperplane (i.e., the signed number, where a positive value means the inside of space limited by decision hyperplane). This hyperplane crates two sets of points: belonging to the learned set of grips and not belonging to it (one-versus-all classification).

$$f_k(\mathbf{v}) = \sum_{i=0}^N \alpha_i y_i K(\mathbf{v}_i, \mathbf{v}) + \rho \quad (2)$$

where  $K$  is a Kernel,  $\alpha_i$  Lagrange multiplier,  $\mathbf{v}_i$  support vector,  $\rho$  free term,  $y_i$  a membership class of a support vector to a grip (1 – vector is a member, -1 – vector is not a member). For  $T = 7$  possible mapping types, corresponding to different grips, we calculate values of kernel functions for  $N = 70$  support vectors.

We have used a LIBLINEAR implementation of SVC classification with linear kernel, through scikit-learn package [33]. It uses one-vs-the-rest strategy for mul-

ticlass classification. Implementation details and exact formulation of classification problem can be found in [10].

Based on the values of membership function, we build a vector of normalised membership weights  $\tilde{\mathbf{f}}$ , which  $k$ -th element has a form (3):

$$\tilde{f}_k = \frac{\exp(f_k)}{\sum_{j=1}^T \exp(f_j)} \quad (3)$$

Vector of SDH gripper control signals  $\mathbf{u}$  can be calculated using formula (4).

$$\mathbf{u} = \sum_{k=1}^T \tilde{f}_k \Theta_k \mathbf{v}^* \quad (4)$$

Where  $\Theta_k$  is a matrix of coefficients for the  $k$ -th grip (size  $7 \times 11$ , where the number of rows corresponds to the number of DOF of the SHD gripper) and  $\mathbf{v}^*$  is an extended by a free term vector  $\mathbf{v}$ . Since vector  $\mathbf{v}^*$  is constant for a given state of the glove we can substitute 5

$$\mathbf{F} = \sum_{k=1}^T \tilde{f}_k \Theta_k \quad (5)$$

and transform equation (4) to the form (6):

$$\mathbf{u} = \mathbf{F} \mathbf{v}^* \quad (6)$$

which gives a smooth transition between the mappings, and further seamless transition between poses, and therefore more intuitive gripper operation. Video presentation of the glove control with grip recognition and smooth transitions between different grips can be accessed at [46].

Vector  $\mathbf{u}$  consists of reference joint positions, being a direct command for a low level PD tracking controller of the gripper. Its stability is realised through the standard PD tuning procedure.

#### 4.6. Initialization of Gripper Mapping Through Sparse Methods

To calculate a pose of the gripper, it is necessary to determine appropriate values of the coefficient matrix  $\Theta_k$ . We have employed a machine learning procedure for this matter: we have gripped objects with our SDH using a particular grip and in the same time posing a hand in the sensor glove in a way that would be equivalent to this grip (compare Fig. 6). After a few experiments (and having a number of data collected) we can proceed with an initialization of the coefficient matrix, that could be further calibrated for particular user needs (as described in the next sub-section).

A pose in each grip is a function of flexion of user's joints but as movement of fingers in particular grip is highly correlated a ordinary least squares regression method for finding coefficients in matrix  $\Theta_k$  could lead to estimates with large variance and reduced accuracy. It is preferable that the mapping would use only

most representative features therefore increasing robustness – as accidental movement in joints that are not representative would not change grippers pose.

To algorithmically find such mapping, instead of arbitrary choice of features we have used a LASSO (Least Absolute Shrinkage and Selection Operator) regression [41]. It is a variable selection method for regression that minimizes residual sum of squares subject to a  $l_1$  norm constraint.

$$\theta_{k,i} = \underset{\theta}{\operatorname{argmin}} \left\{ \sum_{n=1}^N (x_{i,n} - \theta \mathbf{v}_n^*)^2 \right\} \text{s.t.} \sum_{j=1}^{11} |\theta_j| \leq t \quad (7)$$

where  $\theta_{k,i}$  is  $i$ -th row of the matrix  $\Theta_k$ ,  $x_{i,n}$  is an angle position of gripper's joint  $i$  for the  $k$ -th grip, corresponding to the data vector  $\mathbf{v}_n^*$  collected from the sensor glove,  $\theta_j$  is the  $j$ -th element of  $\theta$ ,  $t$  is a maximum value of a sum of absolute values of coefficients.

As the sum of absolute values of regression coefficients must be less or equal to a determined value  $t$ , resulting mapping will have a number of coefficients equal to zero, and hence,  $\Theta_k$  is a sparse matrix, as shown in Fig. 5.

We have implemented this learning procedure using a *scikit-learn* python package with the version of LASSO model that uses a coordinate descent to fit the model [33].

#### 4.7. Online Calibration of Mapping the Grips

If the mapping is not ideal – for example there is new operator, with different hand sizes – mapping can be corrected by this operator using the calibration procedure:

- 1) The user specifies which grip ( $k$  number) will be calibrated. His or her hand must be preset to the pose corresponding to this grip (e.g., Fig. 6 shows calibration of the spherical grip). Readings from the sensor glove are collected to vector  $\mathbf{v}$  and after adding free term we obtain the vector  $\mathbf{v}^*$ ,
- 2) User makes the correction  $p$  for joint  $i$  of the SDH gripper by moving this joint to the new position (e.g., using keyboard, mouse or joystick),
- 3) This correction changes current mapping row-vector  $\theta_{k,i}$  according to equation (8).

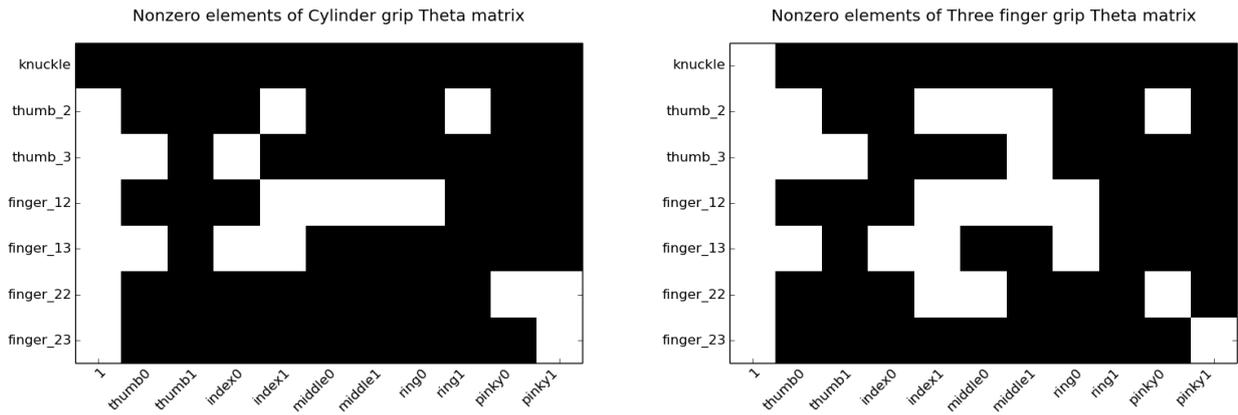
$$\theta_{k,i}(n) = (1 - \lambda\beta) \cdot \theta_{k,i}(n-1) + \beta p \cdot (\mathbf{v}^*)^T \quad (8)$$

where:  $\theta_{k,i}$  is the  $i$ -th row of the matrix  $\Theta_k$ ,  $\lambda \in [0, \frac{1}{\beta})$  is the regularisation parameter,  $\beta$  - the learning rate factor, and  $(\cdot)^T$  denotes the transpose of the matrix.

The consecutive corrections  $p$  minimizes on-line the quality function (stochastic gradient descent algorithm) described by equation (9):

$$J(\theta_{k,i}) = \frac{1}{2M} \left( \sum_{n=1}^M p^2 + \lambda \sum_{j=1}^{11} \theta_{k,i,j}^2 \right) \quad (9)$$

where:  $M$  is a number of corrections,  $j$  indicates elements of vector  $\theta_{k,i}$ .



**Fig. 5. Example of sparse matrices  $\Theta_k$ , being the result of LASSO regression, white indicates nonzero coefficients; vertical axis – SDH joints, horizontal axis – measurements from a sensor glove related to particular joints in the human’s hand**



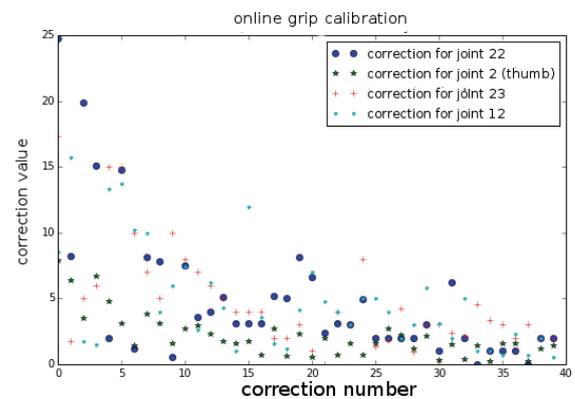
**Fig. 6. Calibration of the spherical grip (from left): hand and gripper open, hand and gripper at the beginning of the calibration process (large discrepancies in the pose of two fingers), spherical grip after calibration**

Minimisation a regularised form of a quality function leads to simple vectors  $\Theta_k$  with only few nonzero coefficients.

Using our approach, the operator can adjust gripper mappings every time when mapping of the glove readings to the pose of the gripper does not perfectly correspond to his/her intentions. This gives necessary flexibility for a range of situations from changing operators to changing sensors. Calibration process, from a completely uncalibrated (without prior initialization) start, is shown in Fig. 7.

**4.8. Gestural Control**

We propose to use gestural interactions as a way to additionally control gripper. Gestures, provide a way to pass symbolic commands to the system through movement and pose of hands. Using specific movement to control machines has become very popular with advent of smartphones and tablets with touchscreens – using fingers the user can not only click but



**Fig. 7. Calibration process using stochastic gradient descent.**

also swipe, drag, pinch or rotate to generate specific behaviors of the computer system.

Using integrated vision system, we have also access to the hand gesture recognizer that categorises gestures and also provides information about direction, speed, position of gesture and some additional information unique for particular gesture (e.g., angle of hand rotation or “strength” of hands grip).

Mapping functions described in previous sections give an intuitive method to control a gripper in mobile manipulation task. However, for reasons described in Section 1 it is very desirable to safely pause or stop manipulation process. In the paper [5] authors concluded that without a way to interrupt manipulation process, controlling is extremely tiring and stressful as the manipulator all the time follows any operator’s move. Moreover, the whole process is more prone to mistakes and slower.

To relieve the operator, we have introduced several control modes that can be changed using operator’s free hand (in mid-air, without taking eyes off the object), as shown in Fig. 4. There are three modes of operation: gripper control, model control (preparation) and stop. Any movement of the gripper can be planned and tested on the model, before actually moving the real SDH, we have called our solution Shadow Hand [49].

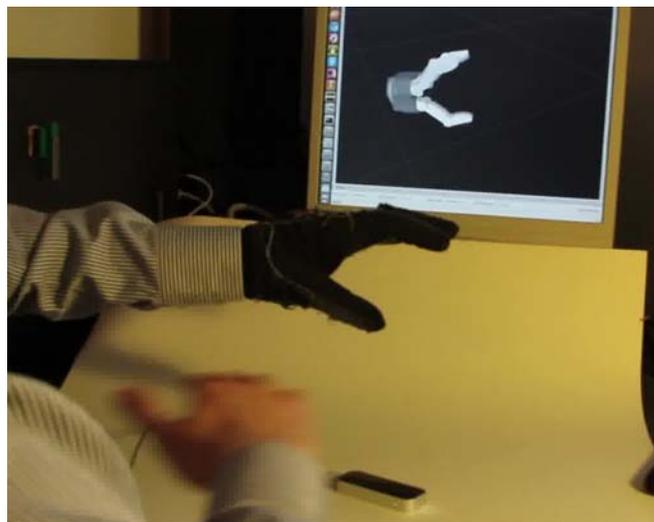
Similar solution is used in the telemanipulation of the space robot Robonaut, where three modes – freeze, thaw, and index are switched by voice commands [16]. Operator also can change modes without losing focus on the object (as in our solution) but voice commands can be problematic in high noise environment of rescue tasks. Additionally hand movement gives information that can be easier to interpret – speed of hand movement is easier to control by the operator than for example, pitch of his/her voice.

To successfully create a gestural interface that would have features explained in chapter one we had to take into account several aspects of the design:

**Using limited number of familiar gestures.** Gestures as a symbolic method of communication are based on memorisation, that is they depend on users knowledge and background [25]. To be able to use a gesture, user must know it beforehand, therefore if we want our interface to be used without (or with limited) prior training all used gestures must be easy to memorise and in limited number, otherwise users will be prone to become tired and make mistakes [27]. Currently we are using swipe gesture to change mode and grab gesture to stop Fig 8.

**Clear and reliable feedback.** Gestures are interpretable, so there is a risk that the system will misinterpret command or user will send wrong command. A working system will provide feedback stating whether the command was understood and what is the current status of the system. In our approach we use voice signal – program says that state has changed and what is current value. Different states show also different colors on the screen.

**A clear delineation of different modalities.** If direct control of the gripper and gestures are done in the same time and in the same space, it could lead to



*Fig. 8. Swipe gesture used to change gripper control mode.*

confusion and unintended movements. In our system, the gestural interface is limited to the hand that is not using sensor glove and the movement has to be done in some particular space. Therefore, there is limited chance that user would move his/her hand only in order to change the state of machine and as a result also change gripper’s pose or vice-versa by moving hand in glove, change gripper’s mode.

We have used LeapMotion API’s gesture recognition. System can robustly recognise number of gestures, its speed and direction [49]. Basic gestural information is passed to ROS node that decides, basing on gesture probability (information that is also produced by LeapMotion API) whether mode should be changed and to what – that is based on a state machine.

Video presentation of hybrid control interface with direct and gestural control can be accessed at [47]. Four work modes, gripper control, manipulator control, simultaneous control and freeze are accessed through gestures, while second hand equipped with sensor glove directly controls gripper position and orientation (through visual tracking of hand position) and grip pose through grip recognition and mixing of mappings.

Program communicates through ROS using topics as well as produces adequate sounds using PyGames python library. Information of a current state and a change of state can be also seen as a text on a screen and change of color in program GUI.

## 5. Summary and Further Work

We have presented our solutions for intuitive and practical control of mobile grippers. They meet most of the requirements for control interfaces in rescue/exploratory robotics.

In our further work we will be using feedback information from tactile sensors mounted on the gripper for more interactive grasps. We plan to examine ways of communicating feedback without full force control.

Our main goal is to test interfaces with real users. For that matter we are working with users of rescue robots and we are building a test stand for telemanipulation task.

## ACKNOWLEDGEMENTS

Research was partially supported by the National Centre for Research and Development under grant No. PBS1/A3/8/2012.

## AUTHORS

**Igor Zubrycki\*** – Lodz University of Technology, Stefanowskiego 18/22, 90-924 Lodz, e-mail: igorzubrycki@gmail.com, www: www.robotyka.p.lodz.pl.

**Grzegorz Granosik** – Lodz University of Technology, Stefanowskiego 18/22, 90-924 Lodz, e-mail: granosik@p.lodz.pl, www: www.robotyka.p.lodz.pl.

\*Corresponding author

## REFERENCES

- [1] E. Ackerman. “Willow garage introduces velo 2g adaptive gripper”. <http://spectrum.ieee.org/automaton/robotics/robotics-hardware/willow-garage-introduces-velo-2g-adaptive-gripper>.
- [2] Autecsafty.com. “Autec: industrial radio remote controls for all applications”. <http://www.autecsafty.com/en/products>, 2015.
- [3] R. Baer. “Recording crt light gun and method”. <http://www.google.com/patents/US3599221>, August 10 1971. US Patent 3,599,221.
- [4] D. Balek, R. Kelley, “Using gripper mounted infrared proximity sensors for robot feedback control”. In: *Proc. of the 1985 IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1985, 282–287. DOI: 10.1109/ROBOT.1985.1087328.
- [5] B. Bruggemann, B. Gaspers, A. Ciossek, J. Pellenz, and N. Kroll, “Comparison of different control methods for mobile manipulation using standardized tests”. In: *IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, 1–2. DOI: 10.1109/SSRR.2013.6719378.
- [6] M. Ciocarlie, C. Goldfeder, and P. K. Allen, “Dimensionality reduction for hand-independent dexterous robotic grasping”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems: San Diego, CA, 29 October-2 November 2007*, 2007, 3270–3275. DOI: 10.1109/IROS.2007.4399227.
- [7] S. R. Company. “Shadow dexterous hand”. <http://www.shadowrobot.com/products/dexterous-hand/>.
- [8] M. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks”, *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, 1989, 269–279. DOI: 10.1109/70.34763.
- [9] S. Ekvall, D. Kragic, “Grasp recognition for programming by demonstration”. In: *Robotics and Automation. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, 748–753. DOI: 10.1109/ROBOT.2005.1570207.
- [10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, “LIBLINEAR: A library for large linear classification”, *Journal of Machine Learning Research*, vol. 9, 2008, 1871–1874.
- [11] T. Fischer, “Sensor system for controlling a multifingered gripper on a robot arm”. In: *Proc. of the Int. Conf. on Intelligent Robots and Systems*, vol. 3, 1998, 1509–1514. DOI: 10.1109/IROS.1998.724809.
- [12] Forcimension.com. “Force dimension - product brochures”. <http://www.forcimension.com/products>, 2015.
- [13] Futaba-rc.com. “Futaba® product manuals”. <http://www.futaba-rc.com/downloads/manuals.html>, 2015.
- [14] O. Gibaru. “Control of a ur10 robot and a 3-finger robot gripper thanks to a leapmotion”. <https://www.youtube.com/watch?v=YJ5s3wxKZ30>, 2014.
- [15] G. Gioioso, G. Salvietti, M. Malvezzi, D. Praticchizzo, “Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain”, *IEEE Transactions on Robotics*, vol. 29, no. 4, 2013, 825–837. DOI: 10.1109/TRO.2013.2252251.
- [16] S. Goza, R. Ambrose, M. A. Diftler, and I. M. Spain, “Telepresence control of the nasa/darpa robonaut on a mobility platform”. In: *Proc. of the SIGCHI Conference on human factors in computing systems*, 2004, 623–629. DOI: 10.1145/985692.985771.
- [17] W. B. Griffin, R. P. Findley, M. L. Turner, and M. R. Cutkosky, “Calibration and mapping of a human hand for dexterous telemanipulation”. In: *ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2000, 1–8.
- [18] A. Kowalski, *Wykorzystanie czujnika leapmotion do sterowania chwytaka* (Using the leapmotion sensor to control the gripper), BSc Thesis, Lodz University of Technology, 2014. (in Polish)
- [19] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, D. Gossow, “Strategies for human-in-the-loop robotic grasping”. In: *7<sup>th</sup> ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, 2012, 1–8. DOI: 10.1145/2157689.2157691.
- [20] H. Liu, K. Wu, P. Meusel, et al., “Multisensory five-finger dexterous hand: The DLR/HIT hand II”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS*, 2008, 3692–3697. DOI: 10.1109/IROS.2008.4650624.

- [21] J. Liu, Y. Zhang, "Mapping human hand motion to dexterous robotic hand". In: *IEEE International Conference on Robotics and Biomimetics. ROBIO 2007*, 2007, 829–834.
- [22] M. J. Lum, J. Rosen, H. King, et al., "Telesurgery via unmanned aerial vehicle (UAV) with a field deployable surgical robot", *Stud Health Technol Inform*, vol. 125, 2007, 313–5.
- [23] R. R. Ma, A. M. Dollar, "On dexterity and dexterous manipulation". In: *15<sup>th</sup> Int. Conf. on Advanced Robotics (ICAR)*, 2011, 1–7. DOI: 10.1109/ICAR.2011.6088576.
- [24] R. Marks. "Prop input device and method for mapping an object from a two-dimensional camera image to a three-dimensional space for controlling action in a game program". <http://www.google.com/patents/EP1176559A2?cl=en>, January 30 2002. EP Patent App. EP20,010,306,264.
- [25] D. McNeill, "So you think gestures are nonverbal?", *Psychological Review*, vol. 92, no. 3, 1985, 350.
- [26] L. Motion. "Leap motion release notes and known issues". <https://developer.leapmotion.com/features/faq>. Accessed 27 Dec. 2014.
- [27] M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, "Memorability of pre-designed and user-defined gesture sets". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2013, 1099–1108. DOI: 10.1145/2470654.2466142.
- [28] K. Nagatani, S. Kiribayashi, Y. Okada, et al., "Redesign of rescue mobile robot quince". In: *IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2011, 13–18. DOI: /10.1109/SSRR.2011.6106794.
- [29] J. R. Napier, "The prehensile movements of the human hand", *Journal of bone and joint surgery*, vol. 38, no. 4, 1956, 902–913.
- [30] NASA. "Robonaut 2, the next generation dexterous robot". [http://www.nasa.gov/mission\\_pages/station/multimedia/robonaut\\_photos\\_prt.htm](http://www.nasa.gov/mission_pages/station/multimedia/robonaut_photos_prt.htm).
- [31] U. Navy. "Us navy 090512-n-2013o-013 a mark ii talon robot from explosive ordnance disposal mobile unit 5, det. japan, is used to inspect a suspicious package during a force protection-anti-terrorism training exercise". <http://commons.wikimedia.org/wiki/>. Accessed 16 Feb 2014.
- [32] A. Okamura. *ME 327 Design and Control of Haptic Systems: Lecture Notes*. Stanford University, Stanford, USA, March 2014.
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine learning in python", *Journal of Machine Learning Research*, vol. 12, 2011, 2825–2830.
- [34] A. Peer, S. Einenkel, and M. Buss, "Multi-fingered telemanipulation-mapping of a human hand to a three finger gripper". In: *The 17<sup>th</sup> IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2008, 465–470. DOI: 10.1109/ROMAN.2008.4600710.
- [35] M. Shimamoto, "Teleoperator/telepresence system (TOPS) concept verification model (CVM) development". In: *NASA. Lyndon B. Johnson Space Center, The Sixth Annual Workshop on Space Operations Applications and Research (SOAR 1992) (SEE N 93-32097 12-99)*, 1993, 149–155.
- [36] B. Siciliano and O. Khatib, *Springer handbook of robotics*, Springer, 2008. DOI: 10.1007/978-3-540-30301-5.
- [37] H. I. Son, L. Chuang, A. Franchi, J. Kim, D. Lee, S.-W. Lee, H. Bulthoff, and P. Giordano, "Measuring an operator's maneuverability performance in the haptic teleoperation of multiple robots". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3039–3046. DOI: 10.1109/IROS.2011.6094618.
- [38] Spectrum.ieee.org. "Fukushima robot operator writes tell-all blog - IEEE Spectrum". <http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries>, 2014.
- [39] Taurob.com. "Universal teleoperation console | taurob". <http://taurob.com/produkte/universal-teleoperation-console/>, 2015.
- [40] B. technology. "Bh8-282 datasheet". <http://web.barrett.com>.
- [41] R. Tibshirani, "Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996, 267–288.
- [42] T. Veltrop. "Kinect teleoperation of humanoid robot". <https://www.youtube.com/watch?v=GdSfLyZl4N0>, 2011.
- [43] T. Wojtara and K. Nonami, "Hand posture detection by neural network and grasp mapping for a master slave hand system". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004). Proceedings.*, vol. 1, 2004, 866–871. DOI: 10.1109/IROS.2004.1389461.
- [44] I. Zubrycki and G. Granosik, "Grip recognition and control of 3-finger gripper with sensor glove". In: *Proc. of Int. Conference on Robotics and Artificial Intelligence. Problems and perspective. (RAIPAP'13), Brest, Belarus, 4-6 November 2013*, 2013.
- [45] I. Zubrycki and G. Granosik, "Test setup for multi-finger gripper control based on robot operating system (ros)". In: *Proc. of 9<sup>th</sup> Int. Workshop on Robot Motion and Control*, 2013, 135–140. DOI : 10.1109/RoMoCo.2013.6614598.
- [46] I. Zubrycki and G. Granosik. "Demo of sensor glove to dexterous gripper mapping". <https://>

- [//www.youtube.com/watch?v=e5bEyEErEc4](https://www.youtube.com/watch?v=e5bEyEErEc4), 2014. Accessed: 2014-24-12.
- [47] I. Zubrycki and G. Granosik, "Hybrid control of Schunk Dexterous Hand". <https://www.youtube.com/watch?v=CPTsFYXaRLs>, 2014. Accessed: 2014-24-12.
- [48] I. Zubrycki and G. Granosik, "Leap motion sensor for gripper control". <https://www.youtube.com/watch?v=0v04DKdigds>, 2014. Accessed: 2014-24-12.
- [49] I. Zubrycki and G. Granosik, "Using integrated vision systems: three gears and leap motion, to control a 3-finger dexterous gripper". In: *Recent Advances in Automation, Robotics and Measuring Techniques*, 2014, 553-564. DOI: 10.1007/978-3-319-05353-0\_52.