Proceedings of the 9th International Workshop
on Robot Motion and Control, Wasowo Palace,
Wasowo, Poland, July 3-5, 2013

Test Setup for Multi-finger Gripper Control based on Robot Operating System (ROS)*

Igor Zubrycki[1] and Grzegorz Granosik[2]

Abstract— **This paper presents the concept for a test setup to prototype control algorithms for a multi-finger gripper. The human-robot interface has to provide enough degrees of freedom (DOF) to intuitively control advanced gripper and to accomplish this task a simple sensor glove equipped with flex and force sensors has been prepared for this project. The software architecture has to support both real hardware and simulation, as well as flexible communication standards, therefore, ROS architecture was employed in this matter. Paper presents some preliminary results for using sensor glove and simulated model of the three finger gripper.**

I. INTRODUCTION

In most of the robotic projects we have to solve two major control problems: (1) to provide human-robot interfaces that help accurately and intuitively operate the hardware, and (2) to organize the software architecture in the precise though flexible way to host all control algorithms and envision the possible directions of development. In the presented project we deal with manipulation using multi-finger gripper and have faced exactly these challenges. Advanced grippers, recently more and more popular, have enough degrees of freedom to perform difficult gripping tasks in both industrial and home environments. On the other hand, it is difficult and time consuming to operate them and program manipulation tasks by using standard input devices, like robot consoles and joysticks. Therefore, control of anthropomorphic robotic hands by sensor gloves has been extensively studied [1], [3], [6]–[8]. Even though, in our project we have focused on three-finger gripper we decided that, for sake of intuitiveness, it is worth to use sensor glove as an input interface. The detailed description of the system setup is shown in the next section. As the number of outputs from the glove exceeds the number of control inputs for the actuators, the simple master-slave operation with direct output-to-input mapping has to be replaced with additional data processing. Also the early stage of the project forced us to use the simulator instead of real manipulator with the gripper that are being built. Here comes the second of the challenges mentioned above the choice of software architecture. We would like to simply acquire and process data, control actuators both in the simulation environment and real hardware, and visualize results. The Robot Operating System (ROS) seems to be the solution fitting our needs. It is a thin, message-based,

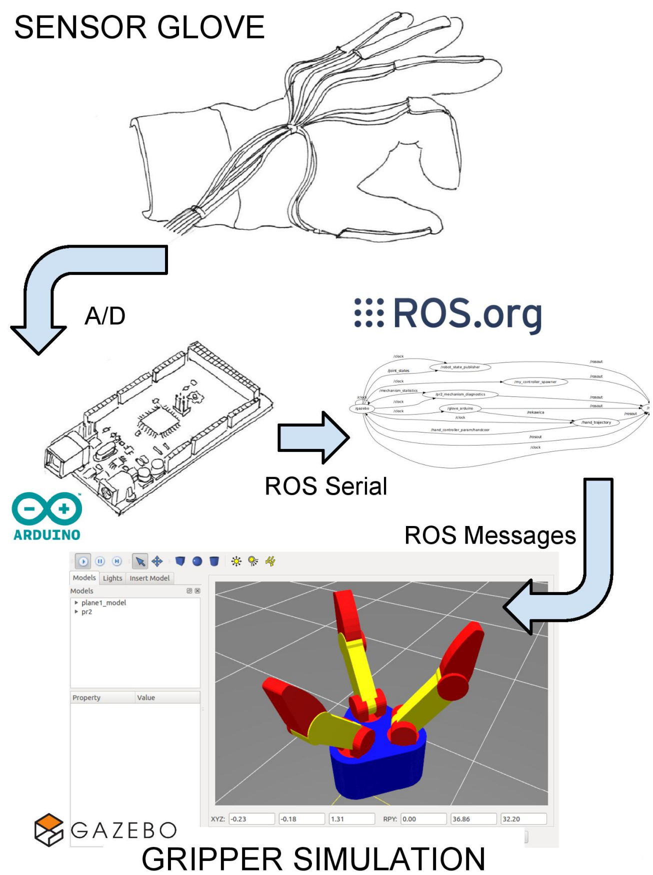

Fig. 1. Structure of test system

tool-based system designed for mobile manipulators. ROS is based on a Unix-like philosophy of building many small tools that are designed to work together [3]. The application of ROS into our project is presented in Section III while test results are shown in Section IV.

II. SYSTEM SETUP

A. Overview

The proposed test setup to prototype control algorithms for a multi-finger gripper (shown in Fig. 1) is composed of: in-house-made sensor glove, acquisition module based on Arduino board, PC computer running ROS core that provides connection between several software modules, and Gazebo simulator with model of three finger gripper. We envision

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

[1]I. Zubrycki is PhD candidate at Lodz University of Technology, Poland e-mail: igor.zubrycki@dokt.p.lodz.pl

[2]G. Granosik is with Lodz University of Technology, Poland, ph. +4842 6312554, fax +4842 6312551, e-mail: granosik@p.lodz.pl

that the system will be extended by a vision system to track the position and orientation of the glove, and therefore, to provide all data necessary to control the manipulator. Also the simulated hardware will be replaced by the new designed arm and gripper. We focus further report on the glove design and software implementation.

B. Glove design

The purpose of our sensor glove is to measure angles between finger phalanxes and forces of contact between hand and the environment. There are several different methods used successfully to measure joint angles in human hand. All of them, apart from the first one, are based on data gloves i.e. the sets of several sensors attached to different parts of hand, forming a glove.

- visual pose estimation. Methods in this category usually contain a few steps: building a 3D shape from point cloud, acquired from a 3D scanner (such as Microsoft Kinect) and estimating the hand pose based on feature matching [5]. This method does not require any physical sensors attached to hand, but has low accuracy compared to other methods and is prone to occlusions. It is also computationally expensive.
- motion-sensor based systems. These devices measure the movements using acceleration and angular velocity [6]. Then, by filtering and integrating these measurements, hand pose can be calculated, as well as the movement of the whole hand, relative to the starting position. There is no risk of occlusion, as the sensors are mounted directly on the hand. However, because this method involves integration, even small offset or drift in sensor measurement can produce large position errors in time.
- devices measuring physical change of the properties of material being bent or pressed. These applications use large range of sensors, such as sensitive elastomers [8], fiber optics [7], flex sensors, tactile sensors. They vary on price and on the number of joint angles that they are able to measure. When hand (wearing such glove) changes its pose, there is change in sensor readings as their shape has changed. Again, there is no risk of occlusion but these methods are prone to mechanical wear, because sensors make the continuous change of their shape. Most of them retain information about pose after being switched off.

Our first approach to hand pose estimation was to use the vision setup designed by 3Gear systems. It promises millimeter-level accuracy for measuring of the user's hands [9]. The company offers an API for their product which allows to use information about hand pose computed by system in own applications. Setup consists of two Kinect depth sensing cameras attached to meter-high metal frame, as shown in Fig. 2. We have built this setup and found out in testing that the system indeed is able to track position and orientation of a user's hands robustly. However, it is severely limited though in pose estimation. Even though the system uses two cameras, depth data received from

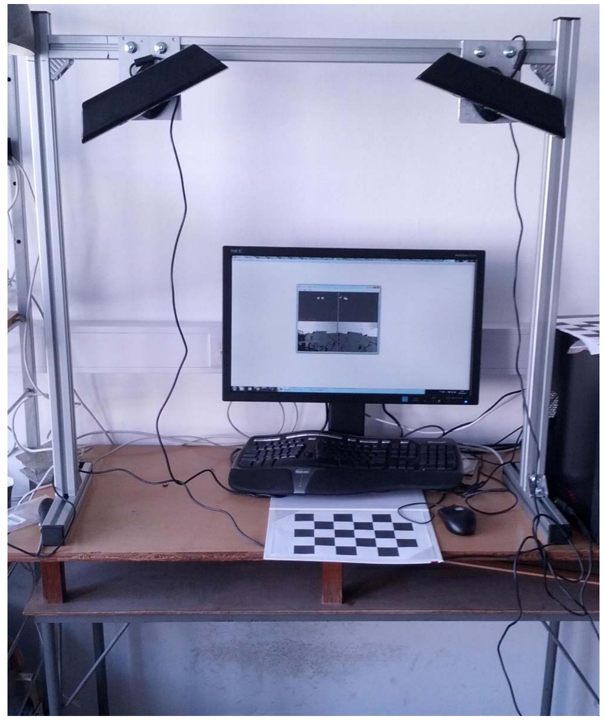

Fig. 2. Visual hand tracking setup, based on 3Gear Systems design

cameras has low resolution and high noise and fingers are frequently occluded which results in overall poor pose estimation (Fig. 3). We plan to use visual system only for hand position and orientation estimation. In our opinion to estimate individual finger poses it is necessary to use sensor gloves. There are a few commercially available sensor

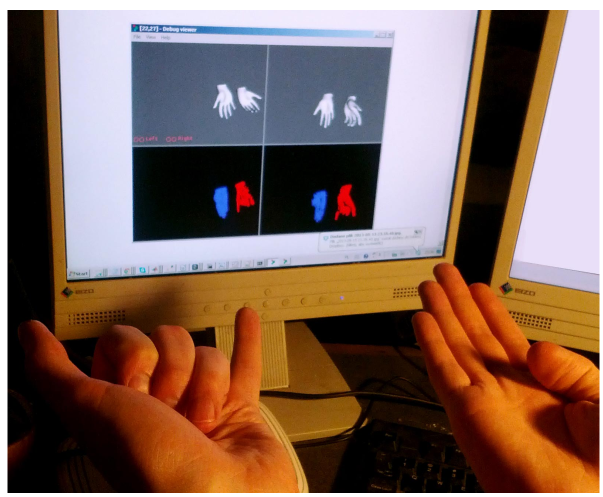

Fig. 3. Comparison of hand pose and estimate by 3Gear software. Pose cannot be estimated because of low sensor resolution and resulting inability to segment fingers.

gloves as well as a number of publications reporting house-made constructions (some mentioned above). These solutions vary in cost and performance, most of them require special technology to manufacture the glove. In our design we have decided on using off-the-shelf components to minimize the

Fig. 4. Sensor placement on our sensor glove (for left hand) [10]: a) inner glove, upper side flex sensors, b) inner glove, bottom side flex sensors c) outer glove, force sensors

cost and speed-up manufacturing. We have used flex sensors to measure hands kinematic configuration and force sensing resistors to provide a haptic characteristics of a grasp. Our glove consists of 10 flex sensors: five positioned on the upper side of the hand, as shown in Fig. 4a, another set of five is positioned on the bottom side of hand as shown in Fig. 4b, the force sensors are mounted on fingertips and in the middle of palm, as shown in Fig. 4c. The two-directional FS Flex sensors have 25 kOhm flat resistance and maximally 125 kOhms of bend resistance. They are combined with 10 kOhm resistors to compose voltage dividers. Finger joints have limited range of movement of about 90 degrees. At this range all sensors had a nearly linear characteristic, as shown in Fig. 5. One can see that these characteristics vary in

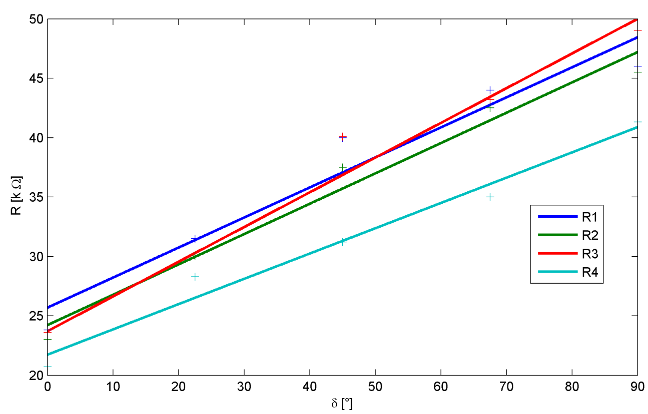

Fig. 5. Static characteristics of the flex sensors [12] (delta - bending angle, R - sensor's resistance)

both inclination coefficient and offset from sensor to sensor between 18.7 kOhm and 28.8 kOhm. This called for the need of individual calibration for each sensor. Force sensors have much less linear characteristic in the range of 0-15 N, as

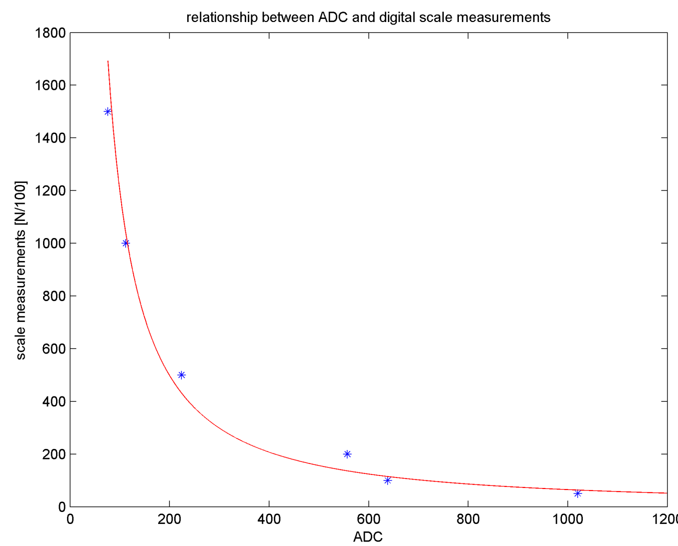

Fig. 6. Relationship between ADC and measured force

shown in Fig. 6, but they are more homogenous in the set of 6 sensors we have tested.

C. Data acquisition and transfer

Sensor glove is connected to the 16 channels of the 10-bit A/D converter built in Arduino Mega 2560 board. The acquired data is formatted into the ROS Message directly on the Arduino Board. Then data is serialized and sent to PC computer by USB port with 57600 baud rate. Special *rosserial* node controls communication, checks data correctness and publishes topics created by Arduino on the ROS System.

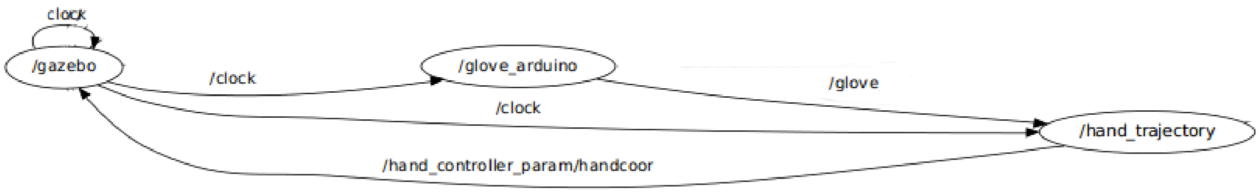

Fig. 7. Structure of the ROS based test environment - only nodes developed by authors are shown, scheme generated by rxgraph

III. Software environment

A. ROS architecture

As mentioned before the software of our setup is based on Robot Operating System. ROS is an open source middleware that provides a structured communications layer above the host operating systems - Linux in presented case. The system is composed of reusable libraries that are designed to work independently. The libraries are wrapped with a thin message-passing layer that enables them to be used by and make use of other ROS nodes. Messages are passed peer to peer and are not based on a specific programming language [3]. Software part of our design is based on several task-specific modules, each working as a separate node in ROS. The full structure of the test software is shown in Fig. 7 using one of the ROS tools called *rxgraph*. Currently they are connected to Gazebo simulation environment, where we can test control algorithms on a physically accurate gripper model. The most important part of the proposed solution is flexibility - this structure provides an easy way to switch from simulation to real robot gripper. Nodes were written in 2 different programming languages - C++ and Python. ROS itself does not limit node format or application (as long as they are able to use ROS messages) and provides several other language client libraries such as *rosjava*, *roslisp* or *roslua*. Arduino board which served as an acquisition and filtering device was programmed in Arduino board version of C++ programming language. Apart from standard Arduino libraries and headers, we used ros.h library from *ROS Serial* and Glove.h library, which was our ROS message format converted to an Arduino library, providing a way to easily serialise and send 16 measurements. One Arduino board can publish and subscribe to several different topics, each consisting of number of variables, limit being its size of RAM. Data from Arduino were transmitted through USB port to glove_arduino node, which de-serialised it and published to ROS. System allows many subscribers to the same topic, so data from Glove could be translated to gripper trajectory in one node and visualised by another (like *rxplot*). This provides convenient prototyping and testing environment. We have created a separate node to convert hand pose information to gripper trajectory. Algorithms used in this node must accurately control end-effector as well as be intuitive for user. For the proof-of-concept tests we have used very simple mapping of the movements of three human fingers: thumb, index and small into the three fingers of the gripper.

B. Gazebo simulator

Gazebo is an Open-Source Robot Simulator, designed to accurately reproduce dynamic interactions between robot and its environment. [1]. Each object is described by XML-based format, with information about its physical (mass, friction, bounce factor) and visual (color, texture, transparency) properties. Based on 3D CAD model of the multi finger gripper, we have created its *URDF* (Unified Robot Developement Format) file. This allowed us to accurately model its properties, such as inertia, static and viscous friction between joints, elasticity, and visual appearance, as shown in Fig. 10. To control a robot that is placed in Gazebo simulation world, we had to create a Gazebo plugin - special ROS node capable of generating control signals directly in simulation update loop. We adapted libraries from PR2 Stack (Willow Garage). Originally they were designed for PR2 robotic platform but using ROS we could easily construct inter-package dependency by specifying them in manifest.xml file. We have structured our system so that it subscribes to handcoor topic generated by hand_trajectory node. Currently, each joint is controlled by independent PID controller and appearance of new handcoor message triggers event, that changes target joint positions for each controller.

IV. Test results

We have tested proposed system at three levels: sensor, acquisition, and control. Sensor readings were presented in the Section II of this paper. Data acquisition was tested by recording measurements (sent from Arduino module) for different hand grasps. We used sensor glove to compare joint positions and force characteristics for 5 different grips and additional pose, namely: cylinder, three-point, precision, lateral, wedge grips and indicate pose. [11]. Flexion angles for each grip ware measured for eighty objects of with different shapes and sizes - aggregated statistical results are shown in Fig. 8. Lower and upper sensors mounted on specific finger correspond to the Fig. 4; side sensor is an additional flex sensor between thumb and index finger. It is essential

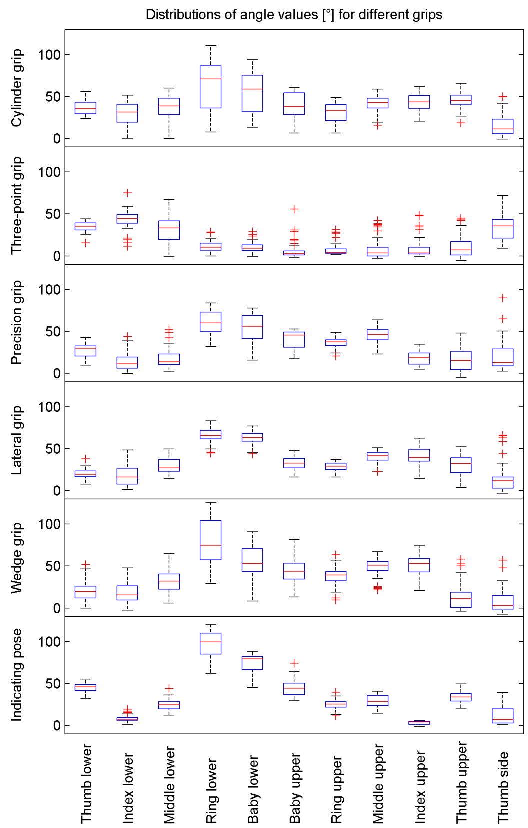

Fig. 8. Distributions of angle values for different grips. Central mark-median, edges of box: 25th and 75th percentiles. Red crosses: outliers

to have a robust mapping algorithm from human hand grip onto a gripper. Because our gripper is non-antropomorphic i.e. it has three rotatable fingers, joint-to-joint mapping, pose mapping or point-to-point mapping methods might not produce satisfactory results [13]. Our approach is to allow users to use extensively visual feedback to learn themselves gripper kinematics. Still, users have to have an intuitive way to control gripper's fingers rotation. Our approach is to rotate them based on hand grip recognition e.g. when the system detects spherical grasp movable fingers rotate to form symmetrical structure, when cylinder grasp is detected fingers rotate to become parallel one to another.

We have tested two algorithms for hand grip recognition. Support Vector Classification with RBF kernel and Forests of Randomized Trees. Both of these algorithms have good generalizability which is important for learning problems with small number of training vectors [14].

To train our models we have used 256 training vectors. Resulting models were tested on additional 71 testing vectors.We found out that SVC had better results with overall precision around 89%, compared to 68% for Forests of Randomized Trees. The best model was found for SVC with C=5 (parameter controlling smoothness of decision surface) $\gamma = 0.0125$ (parameter controlling size of the area of

TABLE I
CLASSIFICATION RESULTS FOR EACH GRIP CATEGORY

Grip name	*Precision* (1)	*Recall* (2)	F_1 score (3)
Cylinder grip	0.93	0.93	0.93
Three finger grip	0.77	1.00	0.87
Precision	0.70	0.78	0.74
Lateral	0.87	0.93	0.90
Wedge	1.00	0.87	0.93
Indicating pose	1.00	0.67	0.80
avg/total	0.89	0.87	0.87

$$Precision = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false positives}} \quad (1)$$

$$Recall = \frac{\#\text{ true positives}}{\#\text{ true positives} + \#\text{ false negatives}} \quad (2)$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

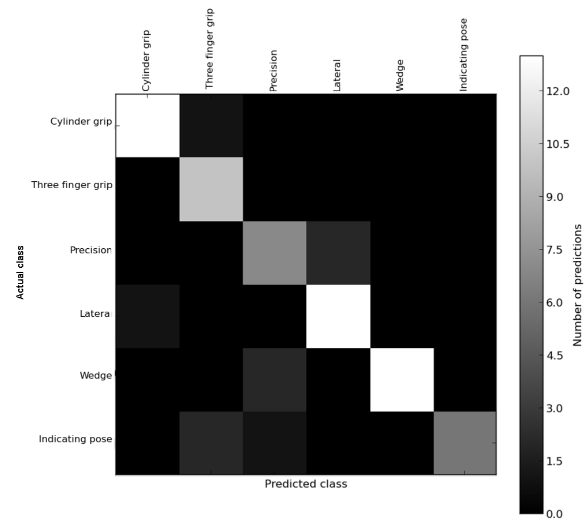

Fig. 9. Confusion matrix for gripper recognition model. In ideal and non-confused situation we should see white squares on the diagonal.

influence of single training example). Results for testing data are presented in Tab. I. Although, confusion matrix shown in Fig. 9 indicates that precision grip was least recognisable and mistaken for lateral grip but, this behaviour is acceptable for the purpose of rotating the fingers.

Finally, we have tested our control interface with a proportional mapping between glove movements and gripper motion as described in Section III. The ability of the robot fingers to rotate around the base of gripper was not used in these demonstrations. Results, presented in Fig. 10, 11, 12 show, that our test setup provides an easy and intuitive way to control multi finger gripper. Several users were able to move robot fingers in desired position, even if it required bending their own fingers in different way.

V. CONCLUSIONS

In this paper we have presented the idea of using sensor glove as an interface to control three finger gripper modeled

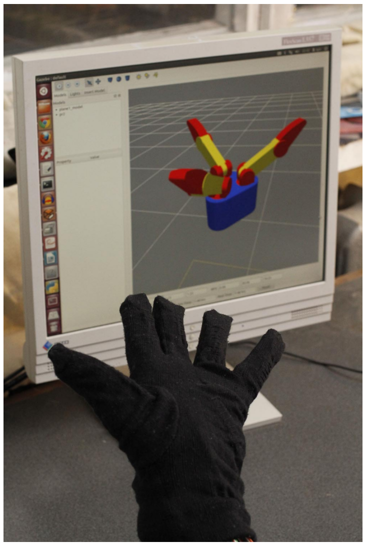

Fig. 10. First pose to test gripper interface

Fig. 11. Second pose to test gripper interface

in the Gazebo simulator. User interface and the model are connected with Robot Operating System. ROS, Gazebo and Arduino provide flexible and scalable platform to test different control methods for robotic gripper. The preliminary results show that our low-cost sensor glove can be used as an intuitive interface for gripper control. We ware able to distinguish various grips from glove's sensor readings with Support Vector Classification with good precision. This will allow us to create mapping function between user interface and three finger gripper. We believe that force sensors mounted on the finger tips will additionally support learning procedures.

ACKNOWLEDGMENT

Authors thank Mr. Mateusz Fraszczynski for building the sensor glove as his BSc project.

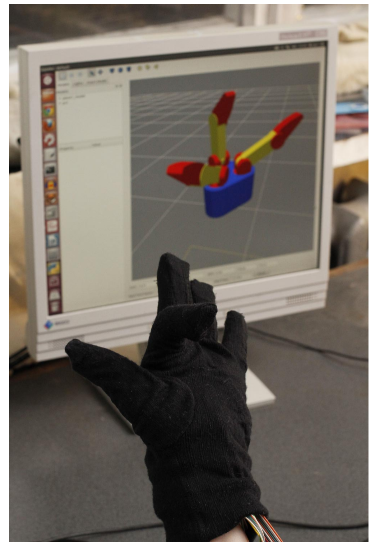

Fig. 12. Third pose to test gripper interface

REFERENCES

[1] Rosell, Jan et al. "Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures." Autonomous Robots 31.1 (2011): 87-102.
[2] Koenig, Nathan, and Andrew Howard. "Design and use paradigms for gazebo, an open-source multi-robot simulator." Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on 28 Sep. 2004: 2149-2154.
[3] S. Cousins, Welcome to ROS Topics, IEEE Robotics & Automation Magazine, Vol. 17, Issue 1, pp. 13 -14, 2010
[4] Matsuo, K.; Murakami, K.; Hasegawa, T.; Kurazume, R., A decision method for the placement of mechanical tactile elements for grasp type recognition, IEEE Sensors Conference, pp. 1472-1475, 2008
[5] Bimbo, Joao et al. "Object pose estimation and tracking by fusing visual and tactile information." Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on 13 Sep. 2012: 65-70. H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.
[6] Kim, Ji-Hwan, Nguyen Duc Thang, and Tae-Seong Kim. "3-d hand motion tracking and gesture recognition using a data glove." Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on 5 Jul. 2009: 1013-1018.
[7] da Silva, Alexandre Ferreira et al. "FBG sensing glove for monitoring hand posture." Sensors Journal, IEEE 11.10 (2011): 2442-2448.
[8] Tognetti, Alessandro et al. "Characterization of a novel data glove based on textile integrated sensors." Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE 30 Aug. 2006: 2510-2513.
[9] Threegear.com. "Technology - 3Gear Systems." 2013. Web. 16 May 2013. http://www.threegear.com/technology.html.
[10] Quigley, Morgan et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software 12 May. 2009.
[11] Wolf, Andreas, Ralf Steinmann, and Henrik Schunk. "Grippers in motion: the fascination of automated handling tasks." Springer, 2005.
[12] M. Fraszczyski, Interface design to control the multi-finger gripper, BSc Thesis, Lodz University of Technology, 2013
[13] Wang, Heng et al. "A mapping method for telemanipulation of the non-anthropomorphic robotic hands with initial experimental validation." Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on 18 Apr. 2005: 4218-4223.
[14] Ho, Tin Kam. "Random decision forests." Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on 14 Aug. 1995: 278-282.