in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

Low-Cost Whole-Body Touch Interaction for Manual Motion Control of a Mobile Service Robot

Steffen Müller, Christof Schröter, and Horst-Michael Gross*

Ilmenau University of Technology, Thuringia, Germany,
steffen.mueller@tu-ilmenau.de
http://tu-ilmenau.de/neurob

Abstract. Mobile service robots for interaction with people need to be easily maneuverable by their users, even if physical restrictions make a manual pushing and pulling impossible. In this paper, we present a low cost approach that allows for intuitive tactile control of a mobile service robot while preserving constraints of a differential drive and obstacle avoidance. The robot's enclosure has been equipped with capacitive touch sensors able to recognize proximity of the user's hands. By simulating forces applied by the touching hands, a desired motion command for the robot is derived and combined with other motion objectives in a local motion planner (based on Dynamic Window Approach in our case). User tests showed that this haptic control is intuitively understandable and outperforms a solution using direction buttons on the robot's touch screen.

1 Introduction

This paper deals with the problem of intuitive human control of a mobile service robot. In our current SERROGA-project (SERvice RObotics for health (Gesundheits) Assistance) [14], we are developing a service robot for elderly that is supposed to maneuver in the narrow home environment of elderly users. The robot's main communication channel is a tiltable touch display, which can be used from a standing or sitting position. For service delivery in a home environment, the robot in many cases has to find and approach a user autonomously to engage in interaction. However, even with the best person detection and navigation algorithms, the user might want the robot to take a different position, making a manual control interface necessary, especially if the robot is too heavy to be pushed and pulled directly. In our case, the robot with a weight of about 40kg (see Fig. 1(a)) can only be pushed manually with a high degree of internal friction in the geared motors, requiring high force and even a suitable bent over handling position to prevent the robot from tilting. The idea presented in

* This work has received funding from the Federal State of Thuringia and the European Social Fund (OP 2007-2013) under grant agreement N501/2009 to the project SERROGA (project number 2011FGR0107).

G. Herrmann et al. (Eds.): ICSR 2013, LNAI 8239, pp. 229–238, 2013.

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

this paper is a low-cost tactile interface allowing for intuitive local control of the robot's position by means of touching the enclosure, which is usable from a standing or a sitting position. This physical interaction should give the user the impression he/she is still simply pushing the robot, while the friction and perceived mass is reduced to an easily manageable amount. In order to prevent from accidental collisions with furniture and other obstacles, the amplified motion command is filtered by the local obstacle avoidance, which is also used for autonomous navigation. Therefore, the robot will evade the obstacles, while the user coarsely pushes in the desired direction. We know, that highly sophisticated force measuring sensors exist, that simulate an artificial skin, but for a mass market product they are too expensive. We could show, that even with a simple sensor, a very intuitive motion control interface could be realized, that improved suitability for daily use a lot.

In the following, the paper will introduce our robot and discuss the state-of-the-art regarding tactile sensors on robots briefly. Subsequently, the touch sensor we use is explained, before the physical model for generating the manual motion command is described. After that, an overview of the software architecture, especially the navigation framework, is presented, showing how the manual command is integrated with obstacle avoidance. Finally, some experiments with users show the benefit against display interaction for navigation.

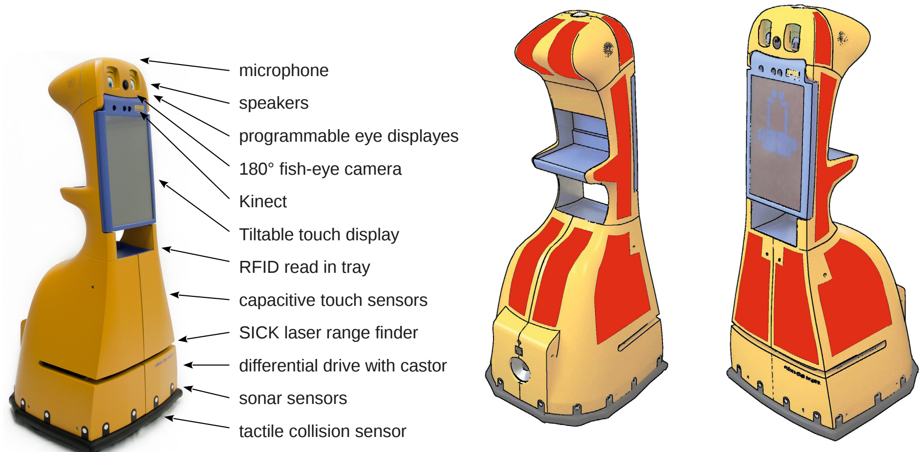
2 The Robot

The platform used (see Fig. 1(a)) is a Scitos-G3, which was developed in the EU-funded project CompanionAble [1]. It is a mobile service robot based on a differential drive with a castor that limits the possible motion to two degrees of freedom. It can move in combinations of rotation and forward/backward translation. Fortunately, the small ground shape of the robot nearly allows for rotation in place and provides excellent maneuverability even in narrow environments. The two driven wheels have a diameter of 20 cm and a distance of 36 cm. They are driven by geared DC motors, which have a high internal inertia making manual moving difficult. Besides the drive, the robot has a tiltable touch display and a head with two OLED displays as eyes.

The robot's covering is formed by a free form enclosure that is made from 4mm polyamide plastic in a rapid prototyping process. It consists of three removable covers for the base and the rear head as well as two side panels and the forehead that are load-bearing parts.

For navigation and user perception, the robot is equipped with a SICK laser range finder at a height of 21cm and an array of 14 ultra sonic sensors at the base. Furthermore, a Kinect sensor and a 180°fish-eye camera in the frontal head assist obstacle and person detection. For user interaction, the robot additionally is equipped with a microphone and a sound system.

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK



(a) (b)

Fig. 1. a) Scitos-G3 (by Metralabs GmbH Ilmenau) robot with the main interaction devices and sensors, b)Placement of capacitive touch sensor electrodes at the inside of the enclosure (red areas)

3 Touch Sensors and Related Work

Before the implementation of our touch sensitive robot enclosure is described, the state-of-the-art regarding tactile sensors will be discussed briefly here.

Tactile sensors comprise various devices that can either measure the position and quality of a contact of objects (mainly the human fingers or hands) with a technical system like a robot, or additionally give quantitative information on the force applied on the mechanical parts of a robot. Sometimes even directional information can be perceived.

[2] identified optical, capacitive and resistive effects as possible sources for information. These effects have been applied for realization of touch displays for many years and can also be brought to other surfaces, e.g. the enclosure of robots. Also acoustic approaches using surface acoustic waves are known from touch displays. An interesting approach is the usage of a flexible textile material, able to cover non-planar surfaces [3].

All these sensors are comparatively cost efficient, but come along with some disadvantages. Capacitive sensors are limited to conductive contact objects, and only qualitative contact events can be observed omitting the direction of applied forces and power of contact.

Alternatively, more expensive techniques are emerging, able to measure force and pressure at a tactile surface. In 1993, a force sensitive transducer technology was applied in [4] for measuring contact forces with a robot. Nowadays, various projects work on artificial skin for robots, that should combine perception capabilities for various stimuli qualities.

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

In [5], a mobile nursing robot is presented for example, which uses an array of discrete pressure sensors allowing for a control of manipulators in order to interact safely with human users.

Also for stationary robotic arms, force and torque sensors in the joints can be used for inference of external forces applied by interaction partners in order to manipulate the robot's position [6].

In [7] the design of cover parts for a humanoid robot is presented, that can measure 3-dimensional force vectors applied to the limbs and body of the robot.

An application that comes closer to ours is the dance partner robot Ms DanceR [8]. Force and torque sensors in the robot's body joints and drives are used to deduce an external guiding command from the human dance partner. This group also used force measurement for assistive transportation devices that reinforce the power of a user while manipulating huge weights.

In our work, we concentrated on a cheap and easy to integrate solution for a robot without any limbs or extra-ordinary drives that enable a force feedback. This has lead to the previously mentioned capacitive technique which has been implemented finally.

When a touch sensor is designed, a compromise on the spacial resolution of sensitive area and the number of simultaneously distinguishable contact points needs to be found. If the quality of contact is known, like finger tips on a touch screen, it is possible to interpolate the position of a contact between specially arranged electrodes, which on the one hand is very exact in the position as known from touch screens, but on the other hand reduces the number of simultaneous contact points. In our case, the touch events are more versatile and of larger area than finger tips, making interpolation difficult. Touching the robot with a full hand should be recognized as well as just sliding fingers over it.

Furthermore, the complex shape of the electrodes makes it difficult to integrate it into a non-planar shape of a robot enclosure. Since we do not need a high spatial resolution, and because we want to superimpose touch events at multiple points on the robot's enclosure, a simple layout of the electrodes has been chosen for testing, that can improved in the future.

4 Capacitive Touch Sensors in Robot's Enclosure

For human-robot interaction, the parts of the robot's enclosure have been equipped with 12 laminar electrodes made from aluminum foil, that are attached to the inside of the plastic covers: four electrodes in the head, two in the side panels and six in the base. Fig. 1(b) illustrates the position of the electrodes as red areas. The electrodes are connected to an Atmega16 micro controller implementing the Atmel QTouch® technology [9] to measure the capacitance of the electrodes, that is affected by a finger or hand near the surface. QTouch needs a minimum of additional circuit elements and thus is very simple to implement. The principle of meassurement is a reference capacitor that is charged in steps, where the sensor electrode's capacity determines the number of steps needed. By means of the 12 independent channels, contact to multiple areas of the robot's

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

cover can be evaluated simultanously due to a cyclic sheme of measuring. The sensor's sensitivity ranges over about two decades, beginning when the hand is hovering 1 to 2 cm over the surface, significantly indicating contact of fingers close to the electrodes, and saturating when the full hand is placed on the electrode's area. The sensors' measurements are sampled periodically at 100Hz. This rather high sampling frequency allows for a recursive band-pass filtering of the raw sensor readings reducing noise to a minimum (low-pass) and enabling a self-calibration (high-pass). This high-pass filter removes drift in the sensor readings and is useful to find the individual operating point of each sensor.

This preprocessing is done directly on the micro-controller, which is sending the resulting touch signals to the robot PC via USB, as floating point numbers in [0, 1] range and at a frequency of 10Hz.

The electrodes have been placed to cover regions of low curvature and span the whole surface beginning from the opening for the laser range finder up to the robot's head. The movable display has been omitted, because of the difficult cable routing and the ability to notice touch at the display anyway. Effectively, almost the complete orange area of the robot (see Fig. 1(a)) is able to perceive contact to human fingers.

5 Touch Based Motion Control

The idea behind the manual motion control is to simulate a physical model of the robot as an inertial mass that is subject to restrictions of the differential drive and external forces and moments that result from the touch interaction. This model has own damping and friction parameters that are magnitudes lower than the real robot's friction. Using that model to compute a velocity, we control the active drives of the robot to achieve a motion that corresponds to the tactile user inputs. By means of this approach, the mass and friction of the robot appear to be reduced and an intuitive control is facilitated. As we will show later, the simulated robot's velocity command is combined with local obstacle avoidance and other navigation tasks to enable safety and integration of internal and external tasks.

For the virtual robot model, we assume that touching the robot at a certain point is associated with a fixedly oriented force vector $\boldsymbol{F}$ that is pointing approximately perpendicularly to the robots outer shape at the offset $\boldsymbol{r}$. These offsets and directions have been configured manually in order to find the most convenient motion behaviour. Fig. 2 is illustrating the principle for a touch sensor at the back of the robot. Touching it there will cause a slight rotation to the left due to the large angle between $\boldsymbol{F}$ and $\boldsymbol{r}$, as well as a forward motion.

It should be mentioned, that the assumption of perpendicular force is not fully eligible in all situations, since people do not push the robot exclusively. In user studies, we observed some intents to rotate the robot by grasping the robot's head and turning it. This would result in forces tangential to the robot's enclosing. In fact, this causes irritations sometimes, if the resulting motion does not match the users intention. We will discuss a possible improvement of our

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

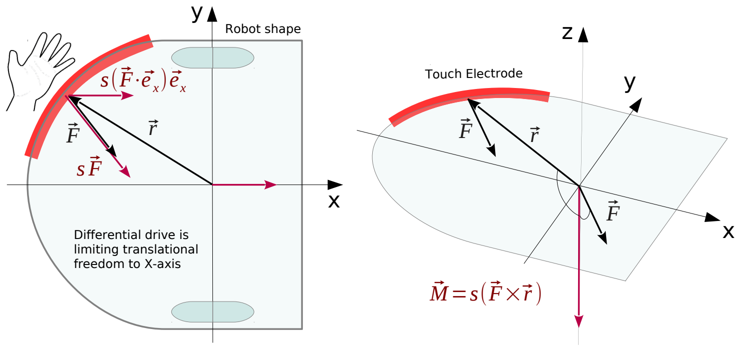


Fig. 2. Illustration of force model used for determining the robot model's acceleration and angular acceleration. Both are using the D shaped footprint of the robot. Left: projection of the unit force vector $\boldsymbol{F}$ scaled by sensor reading s on the x-axis $\boldsymbol{e}$, Right: cross product of offset vector $\boldsymbol{r}$ and force vector $\boldsymbol{F}$ lead to a rotational moment $\boldsymbol{M}$.

approach relating to this problem in the outlook chapter at the end of this paper.

The value of the virtual force applied to the robot model is assumed to be proportionally to the touch sensor reading s, which is representing the contact area and proximity of the user's hand.

From the modeled force vectors and the respective offset position on the robot, we can compute a summarized turning moment $\boldsymbol{M}^*$ and a translational force $\boldsymbol{F}^*$. Considering the limitations of the differential drive (no sidestepping), the force vectors $\boldsymbol{F}_i$ are projected onto the internal x-axis before summarizing over all available sensors: y and z components of the force vectors are absorbed by the wheels in reality.

$$\boldsymbol{F}^* = \boldsymbol{e}_x \sum_i s_i \left(\boldsymbol{F}_i \cdot \boldsymbol{e}_x\right) \tag{1}$$

Here, $\boldsymbol{e}_x$ is the unit vector in x direction. Turning moments result from forces directed not exactly towards the center of mass. The overall turning moment is:

$$\boldsymbol{M}^* = \sum_i s_i \boldsymbol{F}_i \times \boldsymbol{r}_i \tag{2}$$

From the total translational force and turning moment, the accelerations and subsequently the resulting translation and rotation velocity can be simulated, by assuming an artificial friction term f, as well as a hypothetical mass m and moment of inertia L. The model of internal friction is quadratic in the current velocity, which helps limiting the maximum velocity.

Since we model the motion in the robot's internal coordinates, the system state is reduced to a scalar velocity v (translational velocity along the x-axis) and rotational velocity ω. To keep the model in synchronization, we use the

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

actual robot velocity $\tilde{v}_{t-\Delta t}$ and $\tilde{\omega}_{t-\Delta t}$ from the last time step for the prediction of the new target state v_t, ω_t

$$v_t = \tilde{v}_{t-\Delta t}(1 - f|\tilde{v}_{t-\Delta t}|) + \frac{|\boldsymbol{F}^*|}{m}\Delta t \tag{3}$$

$$\omega_t = \tilde{\omega}_{t-\Delta t}(1 - f|\tilde{\omega}_{t-\Delta t}|) + \frac{|\boldsymbol{M}^*|}{L}\Delta t \tag{4}$$

In the most simple case, the resulting model state variables v_t and ω_t can be sent to the robot hardware directly in order to generate a reinforced motion. In our case, the translational and rotational velocity need to be transformed into two wheel speeds v_r and v_l, which are the targets for the hardware controllers. Using the wheel distance d this is straight forward:

$$v_r = v_t + \omega_t \frac{d}{2} \tag{5}$$

$$v_l = v_t - \omega_t \frac{d}{2} \tag{6}$$

However, for the intended application it is not sufficient to directly send the motion command to the hardware, since we want to combine the manual motion command from our robot model with the obstacle avoidance and other internal navigation tasks running in background. The next section, therefore, introduces our navigation architecture in an overview.

6 Navigation Architecture

Besides the hardware design, considerable effort has been spent on a versatile reusable modular software. The robot's navigation software is based on the MIRA middleware [10]. Furthermore, the generic navigation concept of [11] is used in order to combine aims of different navigation tasks.

The navigation concept uses a Dynamic Window Approach [12] for motion planning, that is operating on a two-dimensional search space, spanned by the velocities of the robot's left and right wheel. This way, the differential drive can be modeled very efficiently. The adaptive dynamic window consists of a cell discretization of variable size, for which a set of so-called objectives yields a cost estimation for the respective wheel speed combination. If an objective decides that a specific speed is not allowed, the respective cell is marked as forbidden. Otherwise, the costs of all active objectives are summed up, before the cell with minimum cost is used to select the desired wheel velocities, which is then sent to the hardware controller as a target value.

The flexible and extensible set of objectives is a benefit of the used navigation concept. For autonomous operation, there are a path following objective, a distance objective for avoiding dynamic obstacles, a speed and no-go objective, as well as a heading direction objective. The tactile control is enabled by a further

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

objective. Purpose and mechanisms of these objectives are briefly described in the following:

The *path following objective* is based on the E* planning algorithm [13] that operates on a hierarchical occupancy map, which is updated online using the laser range sensor and the Kinect depth data. The planner yields a map containing the distance to a target position, which is sampled with the predicted position for each wheel speed combination. Reducing distance to the target corresponds to low costs and vice versa.

The *distance objective* is based on the occupancy grid map of the local vicinity, including obstacles perceived by the robot's sensors. Applying a distance transformation, the distance to the next obstacle is stored in the map for fast query. Considering the robot's maximum physical acceleration, the cells in the dynamic window can be forbidden if a collision would be unavoidable using the respective speed. That way, we can keep large distance to obstacles when driving fast, but are also able to pass through narrow doors with a low speed.

The *no-go and speed objectives* also contain a global map of allowed speed vectors. This enables realization of one way areas and thus e.g. socially acceptable navigation behaviors. By means of setting the allowed speed to zero, it is also possible to completely prevent the robot from entering forbidden regions.

The *heading objective* is used to orient the robot when arriving at a target by scoring the difference of the predicted orientation and the target orientation. A very low weight for the heading objective makes it only relevant when the target is reached and the other objectives do not produce significant differences in their vote anymore.

Tactile motion commands are incorporated into the motion planner by a *tactile control objective*, which calculates costs $C(\hat{v}_r, \hat{v}_l)$ for a wheel speed combination that increases with the distance to the desired speed v_r and v_l of the robot model.

$$C(\hat{v}_r, \hat{v}_l) = 1 - exp\left(\frac{-(v_r - \hat{v}_r)^2 - (v_l - \hat{v}_l)^2}{\sigma^2}\right) \quad (7)$$

The parameter σ determines the steepness of the radial cost function.

7 Resulting Motion Behavior

In order to give an impression on the usability of the touch-based navigation control, we conducted a short contest with 10 members of our staff in the age of 25 to 35. The subjects were tech-savvy but did not navigate the robot in before. The aim was steer the robot to three distinct positions in our living lab that have been marked on the floor. The test users first were offered a display-based navigation GUI, consisting of four direction buttons and a stop button on the touch screen. These buttons set the desired velocity similarly to the tactile control objective. Afterwards, the users were asked to play the same game using the body touch control. It turned out, that this was much faster than using the buttons on the screen, the average time for the tour was only one minute compared to 1:43 minutes for Button based control.

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

That confirms what users reported in the interview after the test: Most of them liked the touch control and would prefer it over the screen button based interaction. Nevertheless, as described above, some strange situations occurred, when the users did not respect the assumption of perpendicular forces (pushing only). Once informed on the rules, also these users managed to navigate the robot safely with the touch interface.

Table 1. Durations of navigation task for the test users in the experiment using GUI buttons compared to touch control

	duration of tests for individual users in minutes									
GUI	1:45	2:00	1:48	2:34	1:50	1:22	1:19	1:29	1:15	1:50
touch	0:51	0:52	0:54	1:01	1:10	1:00	0:52	1:06	1:00	1:18

The benefits of the touch-based control result from the analoguous speed control and that it is reachable from nearly any position. In many situations with the GUI based control, the user had to walk around the moving robot when turning it around, which is avoided by the touch based navigation.

Recently, additional usability studies have been conducted with elderly users, that besides the haptic control also comprise a remote control mode using a tablet pc and the autonomous navigation behaviours of the robot. Analysis of this study is still ongoing work.

8 Conclusion and Outlook

It could be shown that an intuitive input modality for local manual robot control can be implemented with very simple capacitive touch sensors placed within the enclosure of a mobile robot. The existing obstacle avoidance capabilities of the robot can easily be combined with the manual control due to a modular navigation concept, which is based on a dynamic window approach.

User studies showed that people favor the touch motion control over a GUI-button-based local motion control.

Some drawbacks could be observed due to the coarse spatial resolution of the only 12 sensor areas. There are parts of the robot's cover that are oriented in different directions but are in the sensing range of the same electrode. In these cases, the direction of the virtual force does not necessarily correspond to the surface normal of the enclosure, which causes a wrong rotation direction in some cases. Also the assumption that the force applied by the user is always perpendicular to the surface is inappropriate. Depending on the relative position of the user to the desired direction of movement, the robot may be pulled sometimes instead of being pushed only. In these cases, the resulting force is tangential to the surface, which is not modeled yet.

Thus, one aspect for optimization is the number of sensor electrodes, which needs to be increased in order to reflect the different parts of the robot's surface

in: Proc. 5th Int. Conf. on Social Robotics (ICSR 2013), Bristol, UK

better. The second option to overcome the drawbacks of the undirected observations of contact is to spend more effort in the mapping from touch sensor readings to the desired motion command. We plan to apply machine-learning approaches in order to learn the mapping from example data. People's interaction behaviour is to be observed while being instructed to move the robot in certain directions. A function approximation can be trained with the values of the touch sensors, the relative position of the user, and the current velocity of the robot as input and the motion vector to the desired position as a target.

References

1. Gross, H.-M., et al.: Progress in Developing a Socially Assistive Mobile Home Robot Companion for the Elderly with Mild Cognitive Impairment. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011), pp. 2430–2437 (2011)
2. Weiss, K., Worn, H.: Resistive tactile sensor matrices using inter-electrode sampling. In: Proc. 31st Conf. IEEE Industrial Electronics Society (IECON 2005), Raleigh, North Carolina, USA, pp. 1949–1954 (2005)
3. Pan, Z., Cui, H., Zhu, Z.: A flexible full-body tactile sensor of low cost and minimal connections. In: Proc. IEEE Intl. Conf. on Systems, Man and Cybernetics (IEEE-SMC 2003), Washington, D.C., USA, vol. 3, pp. 2368–2373 (2003)
4. McMath, W.S., Colven, M.D., Yeung, S.K., Petriu, E.M.: Tactile pattern recognition using neural networks. In: Proc. Intl. Conf. on Industrial Electronics, Control, and Instrumentation (IECON 1993), Lahaina, Hawaii, vol. 3, pp. 1391–1394 (1993)
5. Mukai, T., et al.: Development of the Tactile Sensor System of a Human-Interactive Robot 'RI-MAN'. IEEE Transactions on Robotics 24(2), 505–512 (2008)
6. Cannata, G., Denei, S., Mastrogiovanni, F.: Contact based robot control through tactile maps. In: Proc. 49th IEEE Conference on Decision and Control (CDC 2010), Atlanta, Georgia, USA, pp. 3578–3583 (2010)
7. Iwata, H., Sugano, S.: Whole-body covering tactile interface for human robot coordination. In: Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA 2002), Washington, D.C., USA, vol. 4, pp. 3818–3824 (2002)
8. Kosuge, K., Hayashi, T., Hirata, Y., Tobiyama, R.: Dance partner robot - Ms DanceR. In: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2003), Las Vegas, Nevada, USA, vol. 4, pp. 3459–3464 (2003)
9. QTouch technology website, `http://www.atmel.com/products/touchsolutions/bsw/qtouch.aspx`
10. Einhorn, E., Stricker, R., Gross, H.-M., Langner, T., Martin, C.: MIRA - Middleware for Robotic Applications. To appear: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012), Vilamoura, Portugal (2012)
11. Einhorn, E., Langner, T.: Pilot - Modular Robot Navigation for Real-World Applications. In: Proc. 55th Int. Scientific Colloquium, Ilmenau, Germany, pp. 382–387 (2010)
12. Fox, D., et al.: The Dynamic Window Approach to Collision Avoidance. IEEE Robotics & Automation Magazine 4(1), 23–33 (1997)
13. Philippsen, R., Siegwart, R.: An Interpolated Dynamic Navigation Function. In: Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA 2005), Barcelona, Spain, pp. 3782–3789 (2005)
14. `http://www.serroga.de`