

Reactive Motion Control for Human–Robot Tactile Interaction*

Thomas Wösch, Wendelin Feiten
Corporate Technology
Siemens AG
81730 Munich, Germany
{thomas.woesch,wendelin.feiten}@mchp.siemens.de

Abstract

In the field of service robotics, robots serve and assist human beings. It is natural for humans to directly interact with the robot via tactile interfaces. This paper introduces several kinds of tactile interactions between a user and the robot as well as interactions of the robot with the environment. All interactions are implemented in a single paradigm: Forces measured from tactile sensors result in motion vectors at the contact points. The motion vectors from different sensors are superimposed and then determine the robot's joint velocities. We present results from our experimental setup consisting of an 8 degrees of freedom manipulator arm mounted on a mobile platform. In the illustrated example, a human interacts with the robot using only the tactile interface.

1 Introduction

One of the most challenging parts in robotics during the recent years was and still is *service robotics*. In general service robots perform no repetitive tasks. The working environment is dynamic and in most cases only partially known. Powerful planning and control structures are demanded to move collision-free. Because these robots serve, assist and work with humans their first priority is to guarantee the humans security. The motions should look human-like. No one will trust a machine that acts in a way that seems unpredictable for an observer. The robot must have interaction possibilities that appear natural to a user, e.g. natural language interaction or gesture recognition.

Subject of this paper is the direct interaction between a human and a robot. We focus on tactile interaction using a new kind of tactile sensor, briefly described in section 3. Different interaction possibilities are introduced in section 4, ranging from special motion modes to a *tactile alphabet* for setting these modes. When the system gets in contact with the environment or a human, the tactile sensors provide information about the contact points and the

pressures. Appropriate to the activated mode, we determine for each contact point a motion vector describing a motion for that contact point, see sections 5-6. How this motion vector results into a motion of the robot is described in section 7. The experimental results are presented in section 8. A short description of our experimental setup is also given. The paper is concluded in section 9.

2 Related Work

Specially in the field of service robotics user-robot interaction is of crucial interest. [1] introduces a robot that uses patterns of interactions similar to those found when people interact with each other. This makes the robot motions appear more confidential for the humans in the robots environment. Speech and contact panels are used to interact with the robot. Motivational systems for an autonomous robot for sophisticated human-robot interaction that express the robots mood are investigated [2, 3]. [4] uses tactile information from a force/torque sensor mounted on the tool center point of the robot to modify the tool center points trajectory.

In literature there can be found various interaction possibilities between a human and a robot. Since this paper is focused on tactile interactions, we will only consider those. [5] proposes an artificial skin for a snake like manipulator, using a resistive principle. [6] introduces a modularized sensitive skin using infrared sensors. An acoustic resonant tensor cell for tactile sensing is shown in [7]. [8] proposes a full sensor flexible suit made of conductive fabric and strings. A often used principle is the capacitive one, as described in [9, 10]. A disadvantage of most of these sensor systems is, that their shock absorbency is low. The sensor information can be used to control a robot platform or manipulator arm.

To control a robot system in an environment of average complexity a lot of work was done during recent years. Very popular for controlling a robot have been reactive control schemes based on artificial forces [11, 12, 13]. A reason therefore might be the easiness how they were used and can be extended to higher-dimensional problems with

*This work was partially funded under contract 01IN601A2 and 01IL902DO by BMBF, Morpha project.

more than four degrees of freedom (DOF). In addition they are faster than planning techniques [14, 15] and can work on-line. Their major drawback and the reason why slower planning techniques are still required in robot motion planning is, that reactive mechanisms suffer from local minima. Frameworks [16] where reactive control schemes are merged with planning components, to combine the advantages of reactive controllers (fastness) and the advantages of planners (find a solution whenever one exists) have been proposed.

3 Tactile Sensor

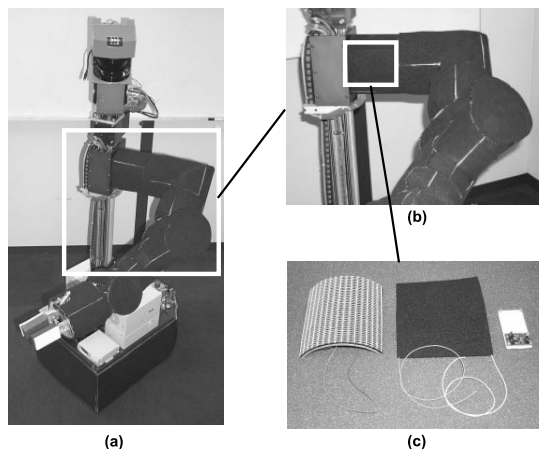


Figure 1: Experimental setup with tactile sensors.

Tactile sensor segments¹ are mounted all over the robot, as illustrated in Figure 1a. Figure 1c shows one tactile sensor segment in detail. The segment consists of two conductive foams, separated by an isolating net. When pressure is exerted on the foams, both foams get in contact. The higher the pressure, the lower is the resistance between the foams at the contact spot. The point of pressure and the pressure itself is measured. Therefore each tactile sensor segment is connected to a micro-controller. When more than one sensor segment is used, every controller is connected via a CAN bus. The segments are referenced by unique identification numbers. Each controller can be considered as master. The bus sharing is done according the *master-master* principle. To get sensor information no polling is required. Efficient data acquisition can be done by this way.

4 Tactile Interaction

Our experimental setup allows us to investigate different types of tactile interaction. These comprise interaction of the user with the robot as well as interaction of the robot with the environment. This section explains these different

types of interaction with its different modes. In the next section it is shown how all of these types can implemented in a single paradigm.

4.1 User guided robot motion

Among the first implementations of user guided robot motion is the guidance of a tool using force torque sensor and zero force control. The force and torque measured by a sensor are compensated by those induced by the tool. The remaining part is assigned to the user's impact on the tool and controlled to zero by appropriate robot motions.

Similarly, with our tactile skin the user can apply a force on the robot and make it move. In contrast to classical zero force control, since the tactile skin is not influenced by robot workpieces or tools we don't have to compensate for those influences. Instead, the measured forces can be used directly.

In the simplest case, the robot will evade the touch. How the motion is derived from the contact force is explained in section 5. However, our tactile skin allows us to implement a lot more complex modes of tactile interaction than this simple *Push* mode. If the user touches the robot and the robot is in *Pull* mode, retracting the contact point will make the robot follow to keep the contact. These modes are not restricted to single contacts, but are extended to multiple contacts with the robot. Each individual sensor segment can be set to *Push* or *Pull* individually, and the resulting robot motions can be superimposed. This also allows for grasping a robot link and guiding it precisely. This can be used to interactively control the redundant degrees of freedom.

4.2 Tactile robot-human communication on a symbolic level

Since the robot has different modes of tactile interaction, mechanisms are needed to switch between these modes. One intuitive way to set the modes is to use the tactile sensor itself. Therefore a meaning is assigned to certain contact points, to the time when they are pressed and the order in which they are pressed. This may be compared to the Morse-alphabet. For example, two short impacts with a short break could set a given segment to *Push* mode, and two short impacts with a longer break could set it to *Pull* mode.

Of course, the information given to the robot can also be more complex. For example, touching the inner gripper pads in an alternating fashion with a certain frequency - left-right-left-right in one second - means to the robot: close the gripper and grasp the object between the fingers, i.e. sets the robot to *Grasp* mode.

Along these lines, a general tactile input alphabet can be defined. However, care has to be taken that this alphabet doesn't interfere with other tasks of the tactile sensor, and that it is intuitive for the user. This also implies that similar

¹Siemens patent, cat. no. 19959703.0

motions are used to convey similar information. Continuing the example of grasping, the robot indicates that it is ready to hand over an object by moving the object left-right-left-right in one second. Then, the user grasps the object and tells the robot that he or she now is holding the object and the robot should release it by wriggling it left-right-left-right in one second between the robot fingers.

Finding the optimal tactile alphabet requires a lot of experiments and is far from easy. In this investigation, inspiration might be found in communication theory.

4.3 Tactile interaction of the robot with the environment

Obviously obstacle detection is a very important feature for an autonomous robot. With the tactile sensor, the robot can detect contacts all over its surface. Therefore, if the robot moves autonomously (not under user supervision or tactile guidance), the tactile sensor segments are set to *Obstacle Detection* mode. In this mode, each contact in the direction of motion is interpreted as a collision and the motion is stopped and then reversed. Since the contact in this case is longer than the short pulses used in the tactile alphabet, even in this mode tactile communication on the symbolic level is still available.

The tactile sensor on our robot can be used to build a model of the environment. In *Exploration* mode, the robot will actively (though carefully) seek contact with the environment and, with reference to the known robot platform configuration [17], insert vertices into a 3D surface representation.

4.4 Interactive tactile sensor calibration

Due to inhomogeneities of the sensor material and variations in the mechanical construction, the tactile sensor system has to be calibrated. This calibration is based on a geometrical model of the robot, with the robot surface patches associated to tactile sensor segments. The calibration software displays this model on a screen, highlighting characteristic surface points on the robot one by one (see Figure 2a). In our experiments, we choose these points to be the corners of the rectangular surface patches, with additional points on more complicated surface patches. For each highlighted point on the screen, the user decides which spot on which tactile sensor segment corresponds to this point and exerts a force on it (see Figure 2b). From this contact, the sensor system automatically determines the corresponding sensor segment and adds the segment id and the corresponding tuple of voltages to a list. In section 5 it is shown how such a tuple of voltages is aligned to points on the robot.

4.5 Mode switching

The modes of tactile interaction need to be appropriate for the current state and goals of the robot. As has been discussed above, an important mechanism to set the mode of

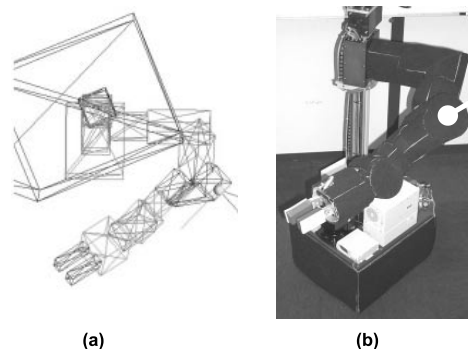


Figure 2: Interactive tactile sensor calibration

tactile interaction is by direct user command. This can be done using the tactile sensor itself, but also using other information channels like natural language interaction, gesture recognition or keyboard commands.

Tactile sensor mode switching is also part of the autonomous robot operation. Just as other sensor modalities are selected (together with the target state of the environment in the image of this sensor), the mode of the tactile sensor appropriate for achieving a given subgoal is part of the task description (see [18]).

5 Force generation

Tactile sensors are mounted on our whole experimental system, as shown in Figure 1a. In this section we will only consider one tactile sensor segment. The way of pressing on the sensor (see Figure 3a) and determining the corresponding force (see Figure 3b) for the motion generation (see section 6) is explained.

When a force is exerted on the sensor shown in Figure 3a, the sensor system evaluates the pressure point

$$\underline{u} \in V \subset R^2$$

that lies in between the voltage tuple domain $V \subset R^2$ of the sensor, as shown in Figure 3c. The vertices

$$\underline{v}_i \in V \subset R^2, \quad \forall i = 1, \dots, 4$$

describe the voltage tuples aligned to the points on the corners of the tactile sensor. The voltage tuple set is given by a convex quadrangle. Introducing the vectors

$$\begin{aligned} \hat{\underline{v}}_1 &= \underline{v}_4 - \underline{v}_1 \\ \hat{\underline{v}}_2 &= \underline{v}_2 - \underline{v}_1 \\ \hat{\underline{v}}_3 &= \underline{v}_3 - \underline{v}_2 \end{aligned}$$

the convex combination of the pressure point can be described by

$$\underline{u} = \underline{v}_1 + \lambda \hat{\underline{v}}_1 + \alpha (\hat{\underline{v}}_2 + \lambda (\hat{\underline{v}}_3 - \hat{\underline{v}}_1)) \quad (1)$$

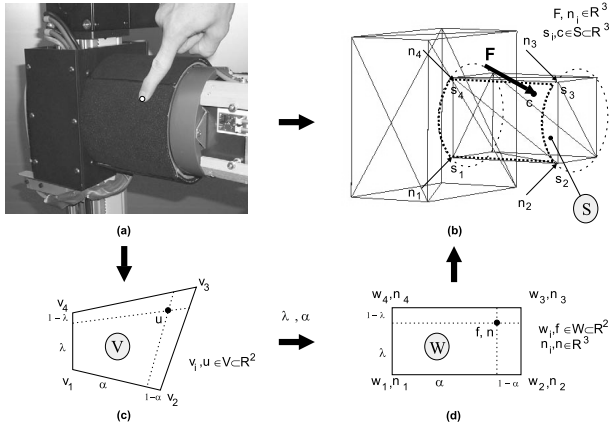


Figure 3: Force generation and contact point evaluation by pressing a tactile sensor.

with

$$\lambda, \alpha : [0, 1] \quad . \quad (2)$$

Using the notation

$$\underline{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \quad \underline{v}_1 = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

$$\hat{\underline{v}}_1 = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad \hat{\underline{v}}_2 = \begin{pmatrix} b_x \\ b_y \end{pmatrix} \quad \hat{\underline{v}}_3 = \begin{pmatrix} c_x \\ c_y \end{pmatrix} \quad ,$$

the parameters λ and α are obtained by

$$\lambda = \frac{1}{2F} \left(\pm D + \sqrt{-4AB + C^2} \right) \quad (3)$$

$$\alpha = \frac{1}{2A} \left(\pm E + \sqrt{-4AB + C^2} \right) \quad (4)$$

with

$$\begin{aligned} A &= a_y b_x - a_x b_y + b_y c_x - b_x c_y \\ B &= a_y u_x - a_x u_y - a_y v_x + a_x v_y \\ C &= c_y u_x - c_x u_y - a_y (b_x + u_x - v_x) - \\ &\quad - c_y v_x + a_x (b_y + u_y - v_y) + c_x v_y \\ D &= c_y u_x - c_x u_y + a_y (b_x - u_x + v_x) + \\ &\quad + c_x v_y - c_y v_x - a_x (b_y - u_y + v_y) \\ E &= -c_y u_x + c_x u_y + a_y (b_x + u_x - v_x) + \\ &\quad + c_y v_x - a_x (b_y + u_y - v_y) - c_x v_y \\ F &= a_y c_x - a_x c_y \end{aligned}$$

The voltage domain V is aligned with a virtual convex quadrangle $W \in R^2$, as illustrated in Figure 3d. Point

$$\underline{f} \in W \subset R^2$$

in the domain of the virtual surface W equals the original pressure point \underline{p} in the voltage duple domain V . Using the vectors

$$\begin{aligned} \hat{\underline{w}}_1 &= \underline{w}_4 - \underline{w}_1 \\ \hat{\underline{w}}_2 &= \underline{w}_2 - \underline{w}_1 \\ \hat{\underline{w}}_3 &= \underline{w}_3 - \underline{w}_4 \quad . \end{aligned}$$

and the parameters (3) and (4), the point \underline{f} is given by the convex combination

$$\underline{f} = \underline{w}_1 + \lambda \hat{\underline{w}}_1 + \alpha (\hat{\underline{w}}_2 + \lambda (\hat{\underline{w}}_3 - \hat{\underline{w}}_1)) \quad . \quad (5)$$

For each corner of the virtual convex quadrangle W in Figure 3d additional vectors

$$\underline{n}_i \in R^3 \quad , \quad \forall i = 1, \dots, 4 \quad (6)$$

are defined. These vectors describe a direction for the vertices \underline{w}_i ($\forall i = 1, \dots, 4$) of the virtual quadrangle. When a point inside the convex quadrangle is considered, the corresponding direction vector is approximated. With the parameters (3), (4) and the vectors

$$\begin{aligned} \hat{\underline{n}}_1 &= \underline{n}_4 - \underline{n}_1 \\ \hat{\underline{n}}_2 &= \underline{n}_2 - \underline{n}_1 \\ \hat{\underline{n}}_3 &= \underline{n}_3 - \underline{n}_4 \quad . \end{aligned}$$

the vector $\underline{n} \in R^3$ describes a direction for the point \underline{f} in the convex quadrangle W by

$$\underline{n} = \underline{n}_1 + \lambda \hat{\underline{n}}_1 + \alpha (\hat{\underline{n}}_2 + \lambda (\hat{\underline{n}}_3 - \hat{\underline{n}}_1)) \quad . \quad (7)$$

The vertices \underline{w}_i ($\forall i = 1, \dots, 4$) of the quadrangle shown in Figure 3d are aligned with the corners

$$\underline{s}_i \in R^3 \quad , \quad \forall i = 1, \dots, 4$$

on the robot link (see Figure 3b). A simple coordinate transformation is used to describe the alignment. When

$$\underline{\underline{M}} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

describes the coordinate transformation between a point on the virtual surface and the robot link, point \underline{f} (5) can be transformed into the contact point $\underline{c} \in S \subset R^3$

$$\underline{c} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{pmatrix} \underline{f} \\ 0 \end{pmatrix} + \begin{pmatrix} m_{14} \\ m_{24} \\ m_{34} \end{pmatrix} \quad (8)$$

on the robot link.

Figure 3b shows the force \underline{F} acting on the contact point \underline{c} of the robot. Using the approximated direction vector \underline{n} (7) as the direction of the force and considering the measured pressure k_{press} , the force at the contact point can be written as

$$\underline{F} = k_{press} \frac{\underline{n}}{\|\underline{n}\|} \quad (9)$$

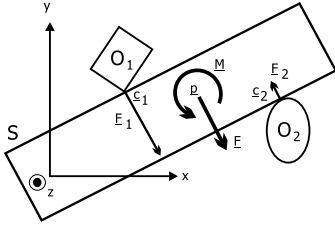


Figure 4: Simplified robot link with two collisions.

6 Motion in Work Space

When the tactile skin detects a contact, a force \underline{F} (9) and a contact point \underline{c} (8) is generated as explained in section 5. For each tactile sensor segment such a force and contact point is evaluated.

One robot link can consist of more than one tactile sensor segments, but can only move in one direction at the same time. Thus all generated forces on a link are summed, resulting in the overall force $\underline{F}_{\text{link}} \in R^3$

$$\underline{F}_{\text{link}} = \sum_{n=1}^N \underline{F}_n \quad (10)$$

Figure 4 shows a simplified robot link. Two obstacles O_1 and O_2 are in contact with the link. The contact points are $\underline{c}_1 \in R^3$ and $\underline{c}_2 \in R^3$. $\underline{F}_1 \in R^3$ and $\underline{F}_2 \in R^3$ describe the generated forces. The control point $\underline{p} \in R^3$

$$\underline{p} = \frac{1}{N} \sum_{n=1}^N \underline{c}_n \quad (11)$$

is used to describe a motion for the link, where the point is attached. N represents the number of contacts. The distance vectors $\underline{r}_i \in R^3$

$$\underline{r}_n = \underline{c}_n - \underline{p}, \quad \forall n = 1, \dots, N \quad (12)$$

describe the distance between the contact points and the control point. Using these distances and the forces generated on the robot link, a torque $\underline{M}_{\text{link}} \in R^3$ acting on the control point can be determined by

$$\underline{M}_{\text{link}} = \sum_{n=1}^N \underline{r}_n \times \underline{F}_n \quad (13)$$

The force (10) describes the translational part of the motion for the control point (11) and the torque (13) describes the rotational part of the motion for the control point. They are summarized in the motion vector $\underline{\delta} \in R^6$

$$\underline{\delta} = \begin{pmatrix} \underline{F}_{\text{link}} \\ \underline{M}_{\text{link}} \end{pmatrix} \quad (14)$$

When there are R robot links to be considered, a motion vector $\underline{\delta}^i$ ($i = 1, \dots, R$) is evaluated for each link depending on the forces acting on that link. Each link consists of one control point \underline{p}_i ($i = 1, \dots, R$) where the motion vector is acting.

7 Motion in Joint Space

A motion in joint space results into a movement of the robot links. Section 6 describes how a motion vector $\underline{\delta}$ (14), describing a motion for a control point \underline{p} (11) on a robot link is derived. This vector has to be transformed into the joint space, to finally result into a motion of the robot links.

[19] uses the transpose Jacobian to transform a vector into the joint space. The Jacobian $\underline{J}_{\underline{p}}$ expresses the motion of the control point \underline{p} relative to the robot link where the control point is attached. The joint torques are evaluated by

$$\underline{\tau} = \left(\underline{J}_{\underline{p}} \right)^T \underline{\delta} \quad (15)$$

When R robot links are to be considered, each link contributes a torque vector $\underline{\tau}^i$ ($\forall i = 1, \dots, R$) to a total torque. The superimposed joint torques can be used to command a motion of the robot links.

8 Experimental Results

We have tested the tactile interaction on a real system. It consists of a manipulator arm with 8 DOF. The arm is mounted on a non-holonomic platform. The whole arm and the sides of the platform are covered by tactile sensors. There are 45 sensors mounted on the robot in total. We used a conventional proportional controller to command arm and base. Limitations such as joint limits, velocity limits or acceleration limits were taken into consideration. The person in Figure 5a-d interacts with the robot. The implementation of the proposed method results in human-like motions. Generated motions fit harmonically into the human-machine interaction scenario. The user is able to freely interact with the robot. A collision detection system is running in the background. Potential self-collisions are detected by the system before the arm moves. The collision detection system and the tactile interaction system are based on the same principle of generating forces acting on the robot. Both systems work smoothly together at the same time.

9 Summary, Conclusion and Future Work

The paper introduces different tactile interaction modes. The interaction can be a human-robot interaction or an environment-robot interaction. When the robot is in *Push* or *Pull* mode, a human can guide the robot. The tactile sensor is also used as an additional source to get information about the environment. In the interaction mode *Obstacle Detection* the robot updates an environment model,

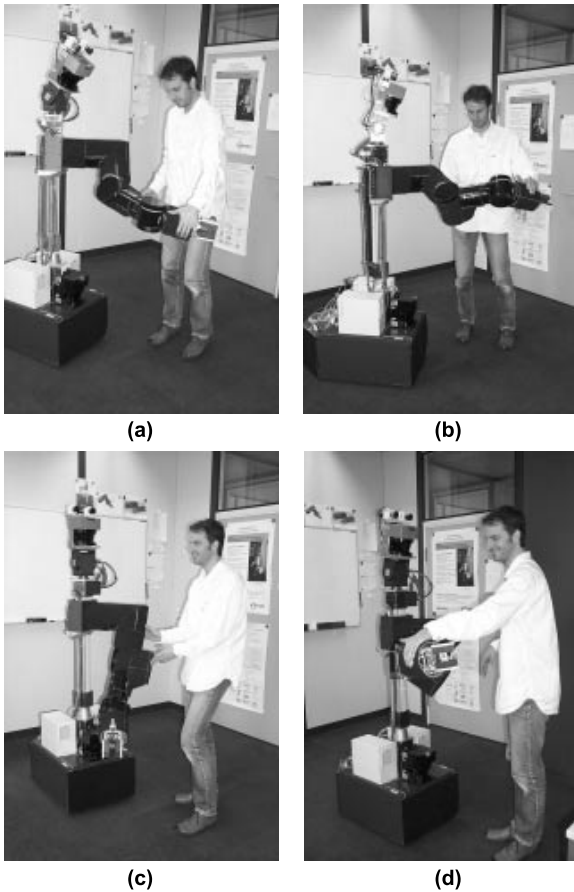


Figure 5: Human–Robot interaction

when collision information is available. The modes can be switched using the tactile sensors according to a tactile alphabet. All modes are implemented in a single paradigm: Forces are combined with torques, resulting into motion vectors. The joint velocities are determined according these motion vectors. Experimental results show motions that appear human-like and fit well into the human-robot interaction scenario.

When the robot is in *Push* or *Pull* mode, currently only the amount of pressure is considered. Further work will be done by considering the length of time of contacting the tactile sensor. A varying amount of pressure over time might also be of interest. The introduced algorithm can fail in local minima. Therefore it will be combined with planning components. The redundant DOF in our setup will be utilized for tactile interaction and object manipulation at the same time. Merging the proposed tactile interaction modes with general interaction modalities, e.g. natural language interaction, to a multi-modal interaction system will be investigated.

References

- [1] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles R. Rosenberg, Nicholas Roy, Jamieson Schulte, and Dirk Schulz, “MINERVA: A tour-guide robot that learns,” in *KI - Künstliche Intelligenz*, 1999, pp. 14–26.
- [2] Cynthia Breazeal, “A motivational system for regulating human-robot interaction,” in *AAAI/IAAI*, 1998, pp. 54–61.
- [3] J. Schulte, C. Rosenberg, and S. Thrun, “Spontaneous short-term interaction with mobile robots in public places,” in *International Conference on Robotics and Automation*, 1999.
- [4] R. Voyles and P. Khosla, “Tactile gestures for human/robot interaction,” in *International Conference on Intelligent Robots and Systems*, Pittsburgh, 1995.
- [5] Petra Bohner and Michael Hoffmann, “An artificial sensitive skin for reactive planning and execution of manipulator paths,” in *International Symposium on Intelligent Robotic Systems SIRS’95*, Pisa, Italy, 1995.
- [6] D.Um, B.Stankovic, K. Giles, T. Hammond, and V. Lumelsky, “A modularized sensitive skin for motion planning in uncertain environments,” in *International Conference on Robotics and Automation*, Leuven, May 1998.
- [7] H. Shinoda, “Acoustic resonant sensor cell for tactile sensing,” in *International Conference on Robotics and Automation*, 1997.
- [8] Masayuki Inaba, “A full-body tactile sensor suit using electrically conductive fabric and strings,” in *Workshop on Intelligent Robots and Systems*, 1996.
- [9] J.L.Novak and J.T.Feddema, “A capacitance-based proximity sensor for whole arm obstacle avoidance,” in *International Conference on Robotics and Automation*, 1992.
- [10] H. Shinoda and H. Oasa, “Passive wireless sensing element for sensitive skin,” in *International Conference on Intelligent Robots and Systems*, 2000, pp. 1516–1521.
- [11] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [12] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [13] H. P. Xie, R. V. Patel, S. Kalaycioglu, and H. Asmer, “Real-time collision avoidance for a redundant manipulator in an unstructured environment,” in *International Conference on Intelligent Robots and Systems*, October 1998, p. 1925.
- [14] L. E. Kavraki and J. C. Latombe, “Randomized preprocessing of configuration space for path planning: Articulated robots,” in *International Conference on Intelligent Robots and Systems*, 1994, pp. 1764–1772.
- [15] C. Nissoux, T. Simeon, and J. P. Laumond, “Visibility based probabilistic roadmaps,” in *Intelligent Robots and Systems*, 1999.
- [16] S. Quinlan and O. Khatib, “Elastic bands – connecting path planning and control,” Tech. Rep., Robotics Laboratory Computer Science Department Stanford University, 1993.
- [17] Gisbert Lawitzky, “Das Navigationssystem SINAS,” in *Proceedings Robotik 2000, VDI-Berichte 1582*, Düsseldorf, June 2000, pp. 77–82, VDI-Verlag.
- [18] G. v. Wichert, Th. Wösch, St. Gutmann, and G. Lawitzky, “Mobman – Ein mobiler Manipulator für Alltagsumgebungen,” in *Autonome Mobile Systeme (AMS’00)*, R. Dillmann, H. Wörn, and M. v. Ehr, Eds. 2000, Informatik aktuell, pp. 55–62, Springer Verlag, Heidelberg.
- [19] Mark W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons Inc., 1989.