Position Paper: Human Skills for Programming-by-Demonstration of Robots

Volker Schmirgel VolkerSchmirgel@kuka-roboter.de
Uwe E. Zimmermann UweZimmermann@kuka-roboter.de
KUKA Roboter GmbH
D-86165 Augsburg, Germany

Thomas Hulin Thomas.Hulin@dlr.de
Carsten Preusche Carsten.Preusche@dlr.de
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Oberpfaffenhofen, D-82234 Weßling, Germany

Editor: Massimo Bergamasco

Abstract

"Programming-by-Demonstration" (PbD) is a very intuitive and efficient means for programming robots for a specific task. One approach for PbD uses manual guided robots to show the task the robot has to perform. Even though manual guidance is a lot more intuitive than traditional programming, first experiences showed that there is still training needed to make full use of such a system. This paper discusses concepts for PbD and studies the required skills. The focus is less on task skills that are to be transferred to a robot but more on the skills for using a PbD system (system skills). Furthermore, several tools for PbD, like virtual fixtures, compliant motion or authoring are discussed. Two demonstrators for evaluating concepts and tools for PbD are proposed and described, one augmented reality and one virtual reality demonstrator, both based on the DLR/KUKA Light Weight Robot. These demonstrators are suited as skill transfer system for programming robots, as well as training platforms to teach novice users how to program a robot with PbD.

Keywords: robotics, human robot interaction, demonstration activity

1. Introduction

Robots are in use in industry and science for many decades, but programming of robotic systems is still a difficult and skillful task, carried out by a limited number of very well trained experts. Nowadays every system has to be installed and adjusted separately from other setups and individually programmed to solve the given task. In most industrial setups, especially in the automotive industry, this is only a minor problem, because those systems are programmed once and perform the same task over and over again up to several years. The programming time and costs are small compared to the overall life cycle costs. A complete different situation is found in new market segments and for innovative applications. Especially in small and medium enterprises robots are no longer performing a single, repetitive task. Instead small lot sizes have to be processed. Therefore robot systems have to cope more and more with higher variants of work pieces and greater variability of tasks.

One possible solution is to use sensors to handle the needed flexibility. But there are several technological limitations: The overall system, its initial programming and its operation is getting even more complex, the perception of the variable parameters is still not solved for all the needed use cases and above all the cognitive abilities of artificial systems are very limited. Therefore this solution is viable only if there are very limited deviations between work pieces and almost unemployable if there are deviations regarding the task to be performed.

Another approach is to efficiently reprogram/redefine the robot task by a human being, who has the needed capabilities regarding understanding, perception and cognition. Of course this programming has to be much faster, easier and more intuitive than traditional programming. This is especially true for small and medium enterprises where workers normally do not have a robotic background. One such method is the so called "Programming-by-Demonstration" (PbD) approach. The basic concept behind this method is to manually guide the robot and to intuitively "demonstrate" the missing information.

This concept has been in focus of research for several years, and one aim of the related work was and still is to drastically reduce the amount of needed knowledge and training for using such system. Even people, who never used a robot before, should be able to program the application immediately. But our experiences with PbD systems have shown that there is still the necessity of training to handle those systems and that the performance of skilled users is much higher than of untrained novices.

But more important is that learning robot programming for PbD systems needs a fundamentally different kind of training than for traditional programming. Traditional programming has a strong theoretical aspect. Therefore training is done in traditional manner using textbooks, lectures, presentations and some limited practical experiences to deepen the understanding and learn real life behavior of the robot. Cognitive understanding is a central part and trainees have to learn for example the coordinate systems, syntax and semantics of textual commands and underlying functionality of buttons on the control panel - sensorimotor skills are not involved at all. In PbD Systems the practical aspects are of much more importance and sensorimotor skills as well as hand-eye coordination are central elements. For many people such skills are easier to learn than abstract concepts. This holds especially for workers which often have strong practical and physical skills but no academic background. Nevertheless training is beneficial and necessary.

One conclusion is that training of the needed skills can not be done via textbooks and presentations anymore. Similar to other tasks which base on sensorimotor skills, like juggling or biking, this is mainly learned by performing the task and by observing and evaluating the output. The SKILLS project deals with the topic how such skills are acquired more easily. This paper will present first concepts how the needed skills for PbD systems could be trained efficiently. One important aspect is that the PbD system itself is used for training purposes, because the main supporting training element envisioned is haptic feedback, which is a basic functionality of PbD systems. Elementary functionalities, like virtual walls and compliance movements are not only useful for the PbD task itself but also a perfect tool for setting up training scenarios.

In the following section 2 the DLR/KUKA Light Weight Robot on which our work is based will be presented. The fundamental concepts of the focused PbD method as well as related approaches are presented in section 3 and the needed skills for using the PbD method are described in 4. The main part (section 5) of this paper is dedicated to the training of those skills by using several supporting technologies and tools for efficient skills transfer. An exemplary PbD application and two training setups, one using augmented and one using virtual reality technologies, are then presented as future research platforms in section 6 before in section 7 a brief conclusion and outlook complete this paper.

2. Description of the DLR/KUKA Light Weight Robot

The DLR Institute of Robotics and Mechatronics has been developing light-weight robots for almost 20 years now. The technology of the third version as presented in Fig. 1 has now been transferred to KUKA for industrial exploitation. In a joint effort DLR and KUKA are now improving the research prototype to an industrial product. So the DLR/KUKA Light Weight Robot (LWR) will be the first commercially available torque-controlled robot, whose features can be used for PbD. In the following the robot and its features will be described in detail. It will be the key-technology for the PbD Demonstrator of the SKILLS project.

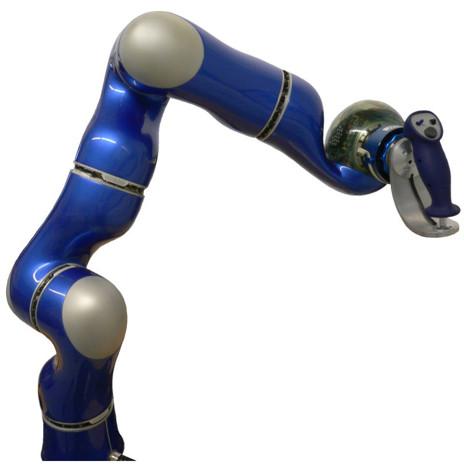


Figure 1: Light Weight Robot with connected joystick handle

The LWR is a light-weight, flexible, revolute joint robot, which by its overall sensory equipment is especially dedicated for work in the sensitive area of human interaction (Hirzinger et al., 2002). The robot's size, power and manipulation capabilities are fairly similar to that of a human arm. With its seven serially linked joints, the robot possesses a redundant cinematics that allows for nullspace movement valuable for avoiding collisions and optimizing its configuration. The robot can be connected to any gripper or tool by a standard robot interface flange, which can also be operated over internal supply lines.

The robot weighs just $14\,kg$ and is able to handle loads up to $14\,kg$. Very light gears, powerful motors and weight optimized brakes have been integrated into the robot. Electronics, including the power converters, is integrated into the robot arm. The integrated sensors are most progressive - each of the LWR's joints has a motor position sensor and a sensor for joint position and joint torque. Thus, the robot can be operated position, velocity and torque controlled. This results in a highly dynamical system with active vibration damping. An additional 6-DoF force-torque sensor can be mounted on the wrist of the robot. This sensor can measure very precisely external forces, e.g. applied by a human operator.

The update rate of the Cartesian loop of the LWR runs at $1\,kHz$. Thus, all commanded and sensed data, including joint angles and joint torques, are updated every millisecond on the connected real-time computer. For skills capturing, these data packets can be processed directly on the real-time computer or sent to another computer. The fast control cycles and the joint-level torque control allow for very fast responses to external forces. Therefore, the robot can be used in numerous fields of application, e.g., in human-robot collaboration scenarios (Grunwald et al., 2003)(De Luca et al., 2006), as generic industrial robot, as advanced manipulator for force torque controlled actions (Albu-Schäffer and Hirzinger, 2002) and as powerful haptic device (Preusche et al., 2001).

Table 1: Specifications of the Light Weight Robot

Total Weight	$14\,kg$
Max. Payload	$14\,kg$
Max. Joint Speed	$120\,°/s$
Sensor Angular Resolution	20”
Maximum Reach	$936\,mm$
Nr. of Axes	7 (R-P-R-P-R-P-R)
Motors	DLR-Robodrive
Gears	Harmonic Drive
Sensors (each Joint)	2 Position, 1 Torque Sensor
Brakes	Electromagnetic Safety Brake
Power Supply	$48\,V$ DC
Control	Position-, Torque,- Impedance Control
Control Cycles	Current $40\,kHz$; Joint $3\,kHz$; Cartesian $1\,kHz$
Electronics	Integrated Electronics, internal Cabling, Communications by optical SERCOS-Bus

3. Concept for Programming-by-Demonstration

Directly programming machines requires expertise on programming languages in addition to knowledge of the machine itself. Besides the native textual based robot languages provided by the robot manufacturer, there are several approaches to make robot programming easier, whether through visual based programming using icons (Bischoff et al., 2002), CAD similar offline programming and simulation tools with or without automatic path planning (Chen and McCarragher, 2000), task oriented programming methods (Diethers et al., 2003) and multimodal instructions based on speech and gesture (McGuire et al., 2002). A survey of robot programming systems has been given by Biggs and MacDonald (Biggs and MacDonald, 2003).

One very interesting approach is the Programming-by-Demonstration (PbD) method (Billard et al., 2008). Even though the basic idea, that the human operator simply shows what the robot has to do, is easy to understand, several different concepts regarding PbD are existing. For example Biggs and MacDonald classify even traditional teach/touch-up methods used for several years in industrial robotics as a PbD system. Here the user moves the robot with buttons, joysticks or 6D-mice to the desired position and this position is stored in the program. This traditional method is of course not the focus of the presented work.

Another category which is out of scope of this paper are systems where the human is tracked while performing a task. Those systems try to capture the meaningful operations and automatically transfer them into a robot executable program. The ongoing research in this field is more related to cognitive science and is still far away from industrial use. Even though such systems are also sometimes denoted as PbD in the literature several other terms are used, like "Programming/Teaching by Showing", "Programming by Example", "Learning from Demonstration" or "Programming by Imitation" (Maeda et al., 2002)(Pardowitz et al., 2007). In this paper we will use the latter to refer to such systems.

Within this work PbD is defined as using manual guidance. This means that the user is in direct contact with the robot and that physical Human-Robot-Interaction (pHRI) is the main difference to Programming by Imitation concepts. Manual Guidance allows to move the robot by applying a force and the robot has to behave in a compliant way. This compliance can be achieved by different technologies. Robots which are backdriveable are one possibility, but most robots used in industrial applications are not backdriveable. To make manual guidance possible for non-backdriveable robots they have to be equipped with additional sensing capabilities, for example force-torques sensors to measure the applied forces. By using special control algorithms the needed compliance of the robot can be achieved. A detailed analysis of impedance control algorithms can be found in the work of Bruyninckx and DeSchutter (Bruyninckx and DeSchutter, 1996) and of Sciavicco and Siciliano (Sciavicco and Siciliano, 2001). The DLR/KUKA LWR described in chapter 2 integrates today's most advanced technologies regarding compliance behavior and is therefore perfectly suited for PbD.

Even though manual guidance is a necessary prerequisite for PbD it is not sufficient and other methods are needed to set up a PbD system. PbD has been used many years in research and to some limitations also in industry, but many of those setups are based on a simple record and playback functionality which is not sufficient for most applications. Additional concepts are needed to bring PbD a step forward. Examples are possibilities to modify recorded movements, to generalize recorded tasks, for sequencing of subtasks and identification of user intention, just to name a few. Those technologies are more general issues also useful for Programming by Imitation systems. Therefore, in our work we will concentrate on specific PbD issues regarding manual guidance. Two main technologies are especially interesting because they can be used also for training purposes and are basic technologies useful within the whole SKILLS project: Virtual fixtures and capturing/rendering of forces and compliance. Again the DLR/KUKA LWR is perfectly suited as a system to realize these two technologies.

Another focus of our work is the concept of defined and undefined constraints. This is a very simple abstraction regarding programming of robots but can be used to explain PbD and training systems. The basic idea behind this concept is that a PbD system has a set of fixed (already defined) constraints M_d and a set of yet undefined constraints M_u. The more elements are in M_u the more flexible the system is (see Fig. 2). In traditional robot programming one starts with a very flexible system A with almost no constraints. The only existing constraints are due to the hardware setup $M_{HW} \subseteq M_d$, which includes robot specific parameters like maximal workspace and velocity as well as singularities, but also peripheral issues, what kind of additional sensors and actors are available. After the system is programmed all constraints are defined and M_u is empty (system C).

In a PbD application the system is set up in such a way that specific constraints are still undefined and have to be defined interactively by the user. In actual systems the first step from system A to system B is done via traditional programming by a system integrator and "only" the second step from system B to the final running application (system C) is done with PbD methods. So one future aim is to bring as many PbD possibilities also into the first step and in parallel raise the level of the PbD systems by enlarging M_u.

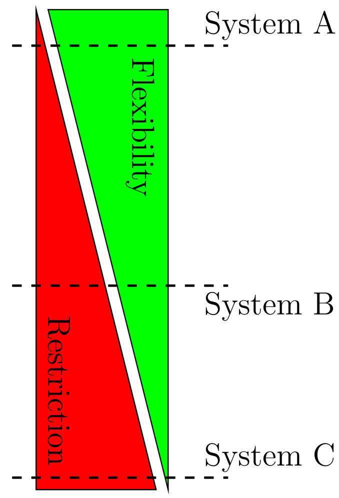


Figure 2: Overview of general methodology of robot programming

Another problem that can be easily explained with this concept is what kind of constraints are within M_u that can be defined with PbD and which kind of information has still to be provided explicitly. For example, nowadays, there are no industrial suitable concepts to define logical condition like "Wait until A is true" or "If A is true than ... else ..." via demonstration. Therefore we concentrate in a first step on the following subsets of definable constraints:

- $M_{GEO} \subset M_u$: Geometrical constraints, e.g. boxes, spheres, trajectories, ...
- $M_{MFC} \subset M_u$: Motion, Force and Compliance.
- $M_{COM} \subset M_u$: Commands regarding peripherals, e.g. Open Gripper, ...
- $M_{SEQ} \subset M_u$: Simple sequences of subtasks (without branches)

There is one big difference between the PbD training scenario and the other SKILLS demonstrator. That is, that the tasks that should be learned are still under development. We think that exactly this is a very strong approach. If the aim is to develop PbD systems that are easily to be learned, the training itself have to be studied in parallel. Only then it is possible to develop the right concepts.

4. Human Skills for Programming-by-Demonstration

The primary goal of Programming-by-Demonstration is to transfer a skill from a human to a machine. To be able to employ such systems, a human operator must have knowledge how to use a PbD system in general. Therefore, the required skills for human operators can be categorized into two groups, skills about the process (task skills) and skills about how to use a PbD system (system skills).

Task skills strongly depend on the field of application. For robotic systems, mainly employed for industrial applications, task skills typically describe characteristics of an assembly or manipulation process, to answer questions like "what has to be processed" and "how has this to be processed". This includes mainly knowledge about the concrete value

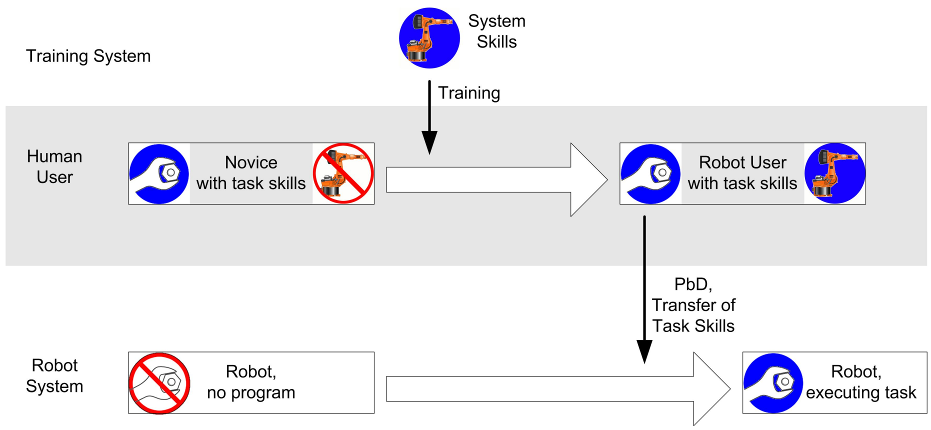


Figure 3: A novice in using robotic systems is trained and acquires system skills. With these skills he can use the PbD system and transfer some of his broad task skills to the robot. After being programmed the robot can execute a specific task.

of process parameters like “correct process speed/force for this kind of material”, “assembly order of a workpiece” or “kind of movement to achieve desired results”. Whereas the system skills are related to general aspects on how to define those parameters with a PbD system, e.g. how to define movements, forces or sequences.

There are two main problems regarding task skills and PbD systems. First, task skills could be different when using a robot instead of pure manual execution. One reason is the different dynamics of the human arm and the robot arm. This could for example be observed if a human is asked to perform a very well known task, like drawing a letter with the help of a manual guided robot. Due to the different dynamics of the system a untrained person has significant problems of performing the task. The second problem is that task skills often consist of implicit knowledge, e.g. the skilled worker could not explicitly describe what the essential properties of the performed task are. This makes transferring skills to a machine very complex, especially when the PbD system is intended to generalize the learned skills and not just perform a 1:1 replay. Task skills can be very rich in variety and are therefore not the main focus of the described work. However, the proposed method of using the robot system as a training platform is also very well suited to train task skills.

System skills enable a human to use a PbD system. These skills depend on the PbD system, not on the process itself. Hence, a human operator has to acquire system skills only once for a specific PbD system and can apply them for all processes of that PbD system. Regarding the concept of defined and undefined constraints the user has to learn basically two things: First, skills how to cope with existing constraints M_d and second, skills how to define the undefined ones M_u.

The main group of already defined constraints result from the robot itself and the equipped peripherals $\mathrm{M}_{HW} \in M_d$. Some follow from the mechanical design, some are necessare due to sfety reasons. The following list is an example of such constraints, which are the main constraints of a system A in Fig. 2:

- deadman functionality
- tool (e.g. gripper) functionality
- robot singularities
- joint limits
- dynamical limits

These constraints limit some important parameters which directly influence the interaction between the human user and the system, e.g. maximal workspace, maximal velocities, maximal acceleration, maximal/minimal compliance and others. These limitations are difficult to understand by just reading documentation; they have to be experienced. A good example is the task to guide the robot in minimal time from one place to another without the violation of the system limits, which would normally lead to an immediate stop.

Another group of already defined constraints are for example application dependent constraints. Within a concrete PbD setup several constraints are already defined (see System B in Fig. 2). Of course the user has to understand and learn how to cope with those constraints. Such application dependent constraints could also further limit the system constraints, like a smaller defined workspace or smaller maximal velocities. Examples for such existing constraints are:

- virtual fixtures, e.g. forbidden regions, preferred orientation, ...
- predefined sequence of actions to perform
- type of allowed actions

Besides skills on how to handle the existing constraints the user has to acquire skills on how to define constraints by his own. This includes as already mentioned a redefinition of already defined constraints, if this is possible. The following list is an example of constraints that can be defined by the user:

- size, form and behavior of virtual fixtures
- points and paths
- process parameters, like velocity, force, ...

5. Tools for Transfer of Human Skills

Transfer of human skills for PbD applications is a challenging aim. The focus will be a PbD application suitable for an industrial environment. As these operating conditions prohibit large scale capturing facilities, we will implement capturing devices that can cope with the restrictions of an industrial environment. Accounting for these needs, this section investigates compliant motion and virtual fixtures as tools for skill-transfer, focusing on haptic human-robot interaction. Subsequently it discusses a way of stating a PbD skill more precisely by means of an authoring-tool. As the demonstrator will work in an environment with limited capturing possibilities and other restrictions, it is anticipated that an expert needs a possibility to interact with the transferring system. The expert will be able to emphasize or deemphasize some features of the captured skills, give the transferred information a preferred order or connect it with logical operations. Especially logical correlations might be difficult to capture but will be easy to organize by using an authoring tool.

5.1 Compliant Motion Capturing

Compliant behavior of robots allows for intuitive and safe human-robot interaction. Combining compliant behavior with spatial trajectories that are commanded to robots, results in compliant motion. In other words, compliant motion describes a motion along an attracting trajectory. It belongs to the category of virtual fixtures which will be discussed detailed in the next subsection.

The following lines address capturing of compliant motion due to its importance for PbD. Capturing of compliant motions means recording the data that is necessary for rendering a compliant motion, i.e. recording concurrently position and force signals. This allows for two possibilities: (A) exploring an environment interacting with the robot and (B) capturing the compliance of a human operator moving the robot. In particular, the latter contains information about human skills. Yet, also the first option is of interest for PbD, for example to capture environmental properties, that can be imitated afterwards in the scope of virtual reality training setups.

5.1.1 Compliance of environment

Objects feature different compliances on their surface, depending on material, geometry and even on direction. A robot equipped with force/torque sensors is well suited for capturing the compliance of real objects. Capturing can be achieved by moving the robot around and “touching” objects. While moving, a force profile is to be captured, i.e. a 6DoF force/torque vector for each point of a gridded space. At the same time the environment can be explored and the geometry of objects can be determined. An exemplary scenario is to capture the compliant motion for turning a screw in a nut. For that task, the one DoF for motion around the axis of the screw has high compliance, whereas the other DoFs are blocked by the screw nut and thus have low compliance.

5.1.2 Compliance of human operator

Rendering of compliant motion can be performed using haptic devices or robotic arms Hogan (1989). Yet, capturing a trajectory with compliant data is much more challenging, because robots can not capture both, motion and compliance, at the same time; especially if the compliance of a human operator has to be captured, (Koeppe, 2001). It is very similar to the problem of capturing forces. If the user has a very high compliance, for example, almost no forces can be measured between the user and the environment.

Therefore, the following strategies are suggested for capturing compliant motion of human operators:

Offline adjustment A robot is first used to capture only motion data, for example an assembly path. Afterwards, values for compliance are set offline by a human operator for all parts of the previously recorded trajectory. This method can store compliant behavior most accurately, as an operator can directly define the values for compliance.

Standard deviation This method is sensible for skills in which compliance is directly related to accuracy. For example during assembly tasks, the accuracy must be high in direct proximity to fragile objects, thus the compliance must be very small. To capture such skills, the motion of one task (action) is recorded several times. From standard deviation between the recorded trajectories, a value for compliance can be determined.

Temporal dependency Research on the psychology of user interfaces has shown that users will slow down motion if they want to perform precise interaction. On the other hand unimportant parts of a trajectory can be performed faster. This observation can also be proven by Fitts' Law for movements on constricted trajectories (Fitts, 1954). This methodology is also used in robotic PbD scenarios to smoothen paths (Mayer et al., 2007). Similar to the second strategy the compliance can be set lower in those areas where high precision and therefore slow movements have been recorded.

Muscle activity This strategy also presumes a direct relation between compliance and accuracy. It is based on the fact that humans increase arm stiffness to achieve higher accuracy and precision of his arm. This is causing muscle activity, which can be measured by additional sensors, e.g. electromyographic (EMG) sensors.

Indirect Estimation One could also make some assumptions from the measured forces and motions. Low measured forces and bigger movements could lead to the assumption that the user has performed the task with high compliance. Whereas higher forces and restricted movements could be interpreted as high stiffness. If the amount of the external forces is known an even better estimation could be made.

Once human compliant motion is captured, it can be used for training of novice, by rendering compliant motion through a suited robot. Even more important is that the data on human compliant motion can be utilized to extract human skills. The amount of compliance is namely correlated to the required accuracy at a certain location (task skill). Furthermore, it gives a hint how to operate with a PbD system (system skill). Yet, the topic of using compliant motion for skills capturing is not completely solved and will be investigated inside the skills project.

5.2 Virtual Fixtures

Sensory guidance tools, also called virtual fixtures, show promising potential in facilitating skill aquisition (Zheng et al., 2006). These tools can be rendered with graphics (visual virtual fixtures) and/or with force features (haptic virtual fixtures). Haptic virtual fixtures are software-generated force and position signals applied to human operators. In most applications haptic virtual fixtures help humans perform robot-assisted manipulation tasks by mechanically limiting movement into restricted regions and/or influencing movement along desired paths. Visual virtual fixtures also transfer task or workspace information to the operator, but by optical signals. An analogy with a real physical fixture a ruler or a template is often used as an example for the functionality of a virtual fixture. The aid of a simple device like a ruler allows a task, which turns out to be difficult to solve by a human (e.g. drawing a straight line free handed), to be carried out fast and accurately. The usual purpose of virtual fixtures is to improve safety, accuracy and speed of robot assisted manipulation tasks (Abbott et al., 2007).

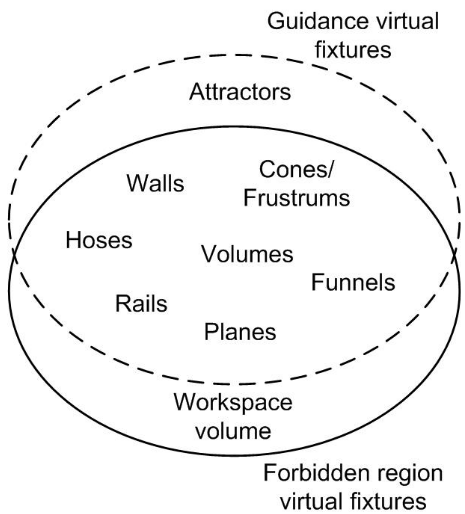


Figure 4: Examples of virtual fixtures

A common categorization of virtual fixtures is their functional effect: *Guidance virtual fixtures* and *forbidden-region virtual fixtures*. Guidance virtual fixtures assist the user e.g. in moving an end effector to and along desired paths or surfaces or forcing a preferred orientation. Forbidden-region virtual fixtures prevent or impede a part of the robot from entering into specified regions within the workspace. Some examples of virtual fixtures clearly fall into one category, like an attractive point in space or a workspace volume. But most examples can be implemented for both categories: Walls, planes, volumes, rails, cones, funnels, hoses can either guide or restrict actions (see figure 4). Further characteristics of virtual fixtures consider their area of influence and the consequence of entering this area. Attracting forces can e.g. be switched on and off or continuously increased according to the distance to an attractor. Entering a volume can affect the orientation or the position of the end effector or a combination of the six degrees of freedom.

As virtual fixtures proved to improve performance in robot assisted tasks it stands to reason to use such sensory guidance tools for PbD tasks. (Abbott et al., 2007)(Marayong et al., 2001) suggest that virtual fixtures can be a useful aid in fine manipulation. Furthermore (Zheng et al., 2006), (Kuang et al., 2004) and (Abbott et al., 2007) show that virtual fixtures can improve both time and accuracy of performing manipulation tasks while still allowing some independent user motion. The ability to give the user the important task specific hands-on information in any task state individually, while allowing the user to define the less important task data himself, makes virtual fixtures a feasible tool for transfer of skills.(Feygin et al., 2002) and (Zheng et al., 2006) evaluate the use of sensory guidance for skills training. (Feygin et al., 2002) found that visual training was better for teaching the trajectory shape while temporal aspects of the task were more effectively learned from haptic guidance. (Kuang et al.,

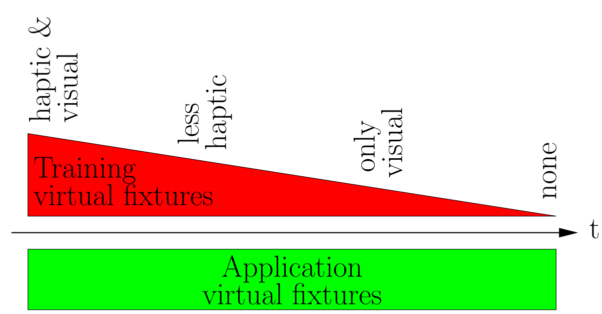


Figure 5: Training virtual fixtures and application virtual fixtures over training time. The amount of training virtual fixtures can be reduced after training.

2004) evaluated the application of virtual fixtures as an aid for guiding a user in a path navigation task. Results showed significant learning and transfer effects measured by performance time and path length. (Zheng et al., 2006) found that for skills transfer an optimal training environment should match the sensory feedback for the designed environment. In detail they found that subjects being trained only with visual virtual fixtures did better in transferring the learned to an unguided environment than subjects being trained with haptic virtual fixtures or even haptic and visual virtual fixtures at the same time. But the traning-transfer environments differed in such ways that the training environment provided virtual fixtures and the environment to transfer to did not have any virtual fixtures. This is a rather abrubt change in demand for the users guidance. We propose a scenario which differs in two aspects. On one hand the application to be trained for has some virtual fixtures (*application VFs*) which help the user in performing the PbD task. On the other hand effects of virtual fixtures added to the application for training reasons (*training VFs*) can be gradually changed. In loosening the contraints over the training period, the user can step by step adapt to the demands of the application (see figure 5).

5.3 Authoring

We anticipate two general circumstances for a human interacting with the machine. On one hand an expert wants to transfer his skills to the system (capturing), on the other hand a novice wants to acquire skills by using the system (rendering). Because of the nature of the demonstrator, the capturing system is almost identical to the rendering system, the setups for experts' use and for novices use will intermingle. Even though capturing of skills from expert and rendering of skills to novice can be treated separately, it is important to state, that in both modes motion/force capturing and motion/force rendering are simultaneously involved. That means, even during the expert capturing phase there will be application dependent motion/force rendering involved and during the novice learning phase the motion/force capturing is needed to "Program by Demonstration".

The expert has full access to all the given interaction and tracking devices and their data (e.g. positions, forces and trajectories of the robot, positions of the tracked hand, detected actions on the machine, like pressed buttons, etc.). In addition to a semi autonomous detection of the relevant skills he has access to an authoring tool (see figure 6). This tool enables the expert to manipulate and extend the captured data (e.g. with AR-objects, virtual fixtures, tolerances, sounds, obstacles) to create a so-called "story". The story stores the relevant data for transferring certain skills to a novice user. It combines captured data with additional input from the authoring tool and stores it in a format usable by the middleware for later use when interacting with a novice user. This way the expert has still full control of the contents being stored, e.g. he is able to emphasize certain features, minimize unimportant information, set or release constraints. Because of the capturing and rendering system being identical, the expert can already access and check the renderable features while creating the story, in order to verify it. Forces are fed back through the robot; supplementary information is fed back through the available multi media devices.

A novice user should ideally be able to use the whole setup without the supervision of an expert. He has, in analogy to the expert, access to all means of interaction the demonstrator provides (see figure 7) but is not confronted with the underlying data nor an authoring tool.

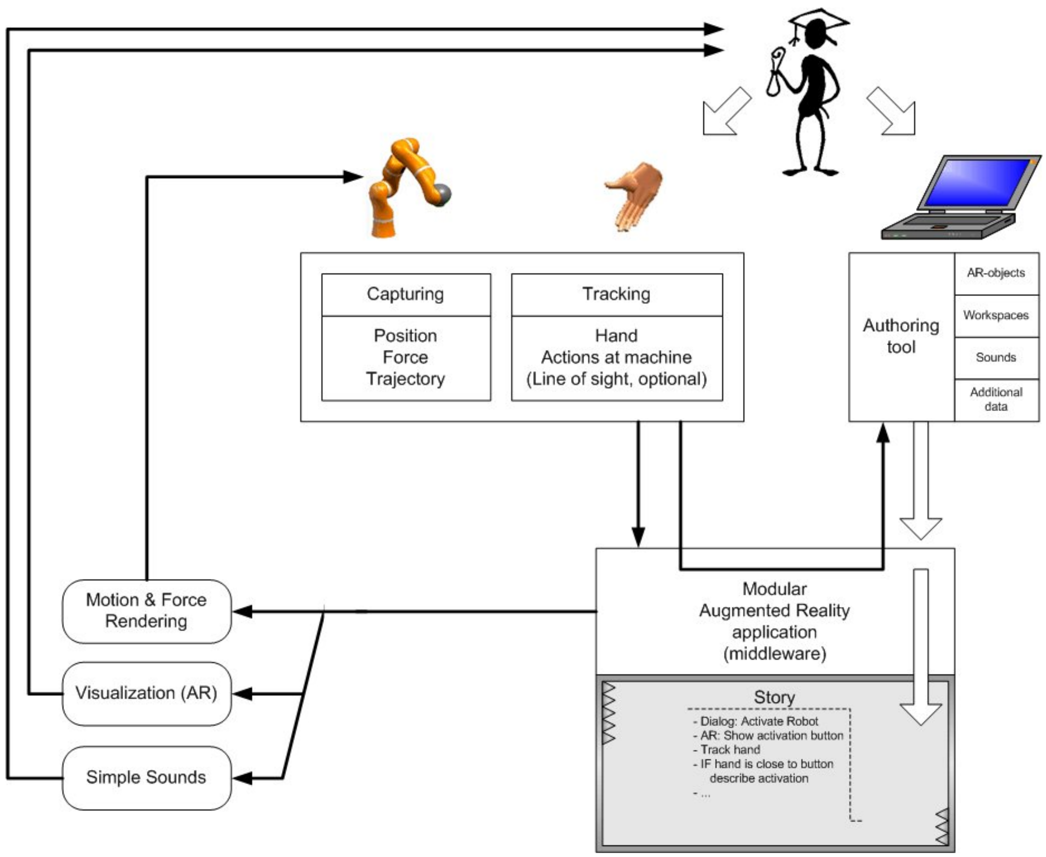


Figure 6: MMI concept for capturing the experts skill

The user's possible actions are limited by the borders set by the expert. Initially the user will use a very limited system with a lot of guidance-feedback and restrictions. He will basically experience a kind of replay of the previously recorded story, which introduces all important parts of the setup (robot, screen, speakers, tracking systems, buttons, panels, objects, etc.). After the user is made familiar with the setup, the demonstrator tries to teach the user skills e.g. by describing a sample task and helping the user in solving it or by describing probable maloperation and potential solutions. According to the story, constraints can be loosened as the user improves. This way the user is slowly guided to use the machine with its full potential and solve problems with it independently.

In this scenario we anticipate a system that proves line-of-sight detection of human users to be very useful. The expert can provide information connected to a certain location. The user can recall object, background or interrelation information just by looking at a correlated area. A line-of-sight sensor can provide this functionality in order to enhance the user's context understanding and allow a better overview in any situation. This information does not need to be directly included in the story but just added as supplementary information and therefore makes the work for the expert a lot easier. The additional hardware needs to be compatible to the authoring tool.

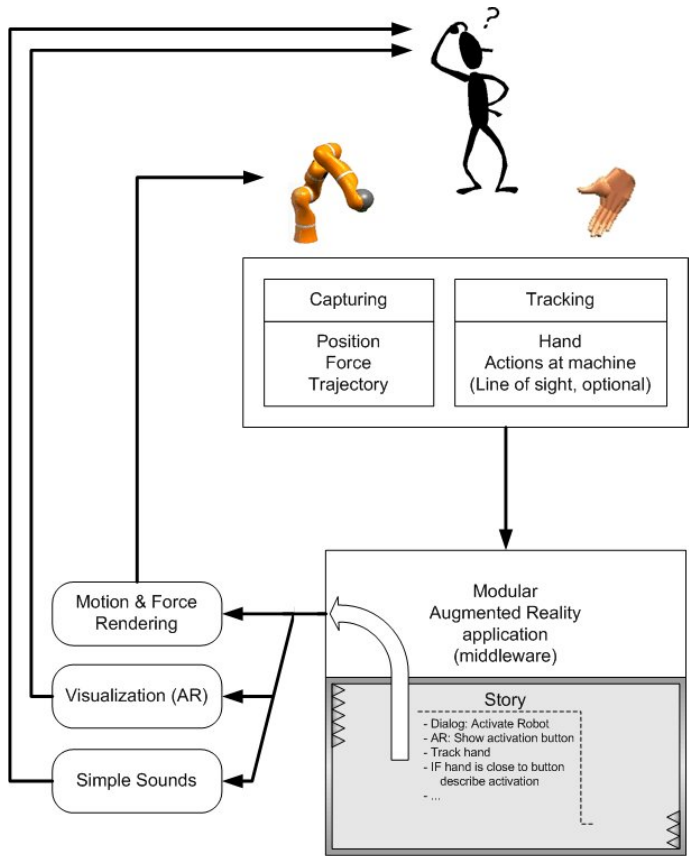


Figure 7: MMI concept for teaching novice users

6. Demonstrator Setups

With its light-weight robot KUKA has introduced a new robot generation that can lift its own weight and shows potential to be easily programmed.

It is based on the DLR LWRIII technology and offers new possibilities in automation for branches and applications, in which robots are not used today. Prospective users in these branches are seldom robotics experts, and hence there is a need for simple and intuitive robot programming and operation methods.

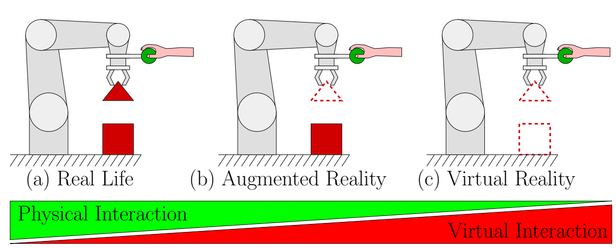


Figure 8: Schematical presentation of (a) real life, (b) augmented reality and (c) virtual reality. There is a smooth transition between these setups with increasing amount of virtual reality means from left to right.

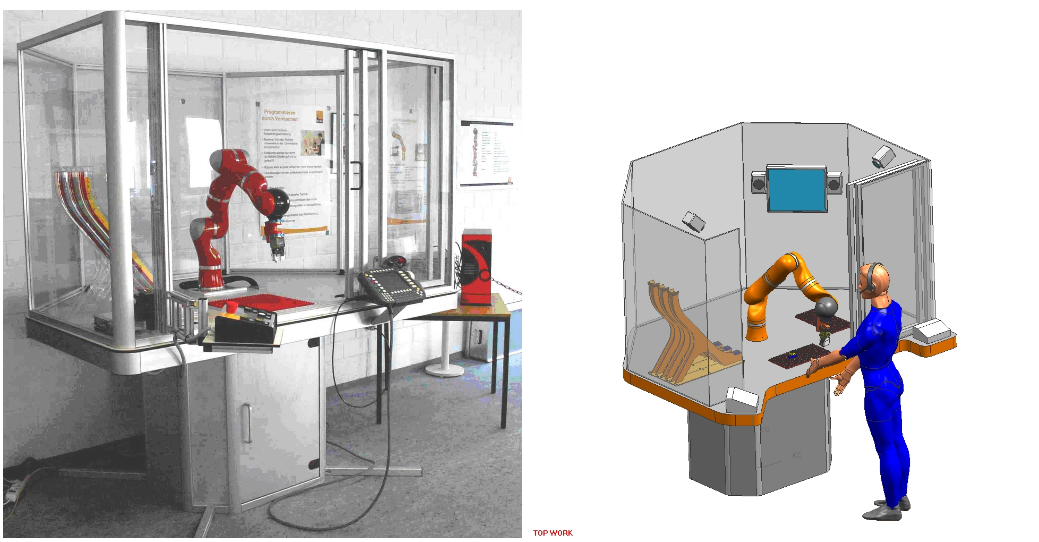
The KUKA/DLR LWR is similar to the human arm with seven degrees of freedom, which results in advanced flexibility in comparison to normal industrial robots. Each of the LWR's joints has a motor position sensor and sensors for joint position and joint torque. Together with the KUKA robot controller the robot can be operated position-, velocity- and torque-controlled. By processing the joint data at high rate and making use of the torque control the robot can compensate or respond to external forces applied. As such the robot can be used as an advanced manipulator for force-torque controlled assembly of goods, or as a haptic input device, e.g., in human-robot collaboration scenarios.

Knowing the model of the robot and the attached tool in terms of their masses and inertia, the torques applied to each joint can be controlled in such a way that the robot is keeping its current pose (so called "gravity compensation mode"). Externally applied forces along the whole robot structure can be detected. To move the robot within its workspace the user simply has to push, pull or turn the tool, links or joints.

Based on this robot, two demonstrator prototypes have been set up that differ in the type of interaction. One demonstrator employs techniques of augmented reality, the other is simulating a scene by means of virtual reality (compare Fig. 8).

6.1 Augmented Reality Setup

A first Programming-by-Demonstration test-bed was set up by KUKA. This test bed is the so-called "LEGO-Cell" which has already been presented on several conferences and fairs. It consists of one LWR with a KUKA control system, a gripper for LEGO bricks, a panel for operating the gripper, a LEGO baseplate and three LEGO brick dispensers (see Fig. 9(a)).



(a) LEGO demonstrator in the KUKA labs

(b) Setup of LEGO demonstrator with additional features for augmentation

Figure 9: Setup for Augmented Reality Demonstrator

As described above a user can move the robot within the workspace just by applying a force to the structure. This way he can guide the robot, pick up LEGO bricks from the dispensers and place them on the baseplate. By placing several bricks on the baseplate a

LEGO structure is built up. The actions of the user are simultaneously recorded, enabling the user to program the robot without writing a single line of code. After the user has finished assembling the LEGO structure, the robot can reproduce the structure several times. The LEGO-Cell is a first attempt to enable even robotic laymen to teach industrial robots. With the exhibited KUKA LEGO-Cell any visitor can easily program a simple assembly task by manually guiding the robot. It has shown, that even young kids, never seen a real robot before, are very successful in using the system.

The assembly task is simple, it is a composition of a limited number of the well known LEGO-brick, and hence the task can be considered a priori known, at least to a certain degree. But even though the task is simple, users still need to be instructed by a continuously supervising expert, showing that skill transfer in this and similar areas is very important. In the current state the system records positions of interest of the assembly task and processes them in a program developed beforehand. The presented system is used as a test-bed for research and development in manual guidance and PbD scenarios.

The existing demonstrator is extended with additional equipment for augmentation of information for skill transfer. Figure 9(b) shows the demonstrator described above with cameras for acquiring workspace view and tracking users hands and an AR screen with speakers on the rear wall to provide the user with augmented reality information. The user can also wear a head mounted see-through display.

The focus of this demonstrator is on an application in real life with additional support from augmented reality. A novice user will be able to learn how to activate the robotic system in order to use it for PbD. After that the main idea of PbD is communicated to the user. In a third step the user learns how to manually guide the robot and make use of the system in order to program a specific process just by showing the system the necessary actions. Furthermore additional system information is transferred to the user: Safety instructions, limitations of the setup, options of the setup, nature of the given specific process.

This demonstrator will be a test bed and validation platform for new approaches and technologies for transferring skills and knowledge to a novice user. Certainly it has not yet direct industrial relevance. Therefore it is anticipated to transfer successful concepts and experiences from this demonstrator to an industrial application.

6.2 Virtual Reality Setup

The second demonstrator focuses on virtual reality (VR). Again the LWR is used as haptic device to interact in a virtual world. As virtual scenario a virtual model of the "LEGO-Cell" test bed, used already for the first demonstrator of previous section, is employed, see Fig. 10.

This demonstrator should illustrate skill transfer from expert to novice. Two different groups of skills shall be transferred, task skills and system skill, which are described in Sect. 4. Similar to the augmented reality demonstrator this platform will be used for capturing and rendering of skills.

The use of virtual models instead of real ones yields several advantages. The VR demonstrator is not in need of real objects, as they are replaced by their virtual representations. This is especially interesting for scenarios with extremely expensive or rarely available parts.

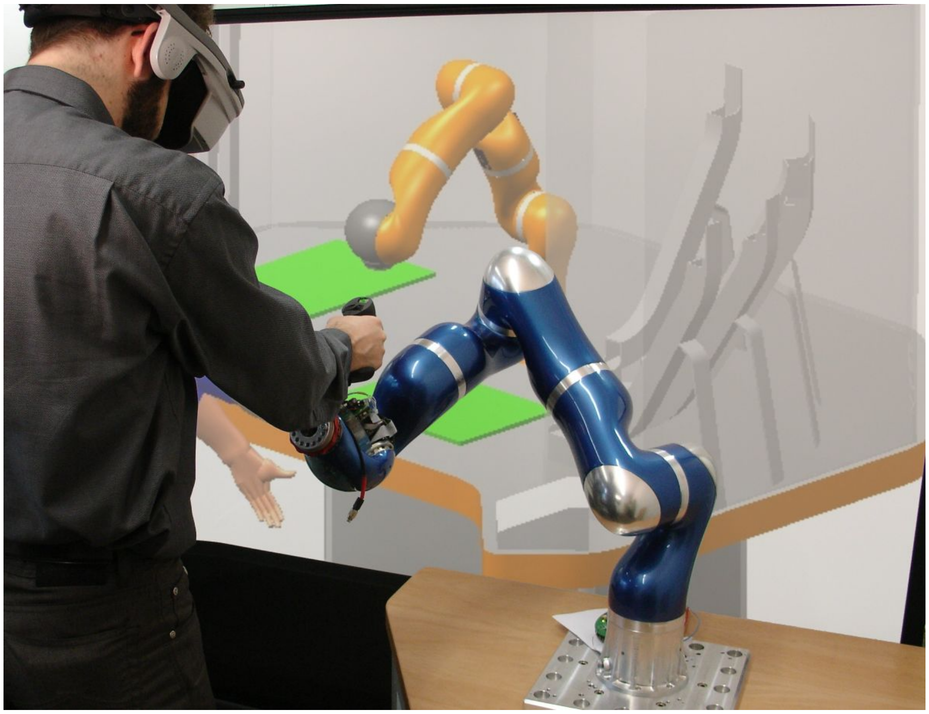


Figure 10: Virtual reality setup of LEGO demonstrator with Light Weight Robot as haptic device.

Another advantage is that manipulation of hazardous objects is harmless if performed in virtual world. Moreover, using means of VR allows for higher flexibility in modifying objects or the whole scenario. Also the preparation time of simulations can be reduced.

Concerning transfer of skills, the VR demonstrator increases portability and reproductivity of captured skills that have to be distributed. Particular a distribution of skills among international institutions is made possible.

To increase immersion of the virtual simulation, several extensions are expedient. Adding haptic stimuli to other parts of a human, in addition to force feedback to the human hand, would help the human operator to orientate himself in a virtual environment. One possibility would be a vibrotactile devices for human arms, that gives haptic information about collisions of the arm with virtual objects or virtual guidances.

7. Conclusion

“Programming-by-Demonstration” for robotic systems is a novel way to intuitively programm robots for a certain task by direct human robot interaction. This method has a huge potential to speed up and reduce costs for programming robots. But nevertheless human users still have to learn how to use such systems. On the one side skills how to cope with existing constraints, e.g. different dynamics, have to be aquired. On the other side skills how to define the missing parameters also need training. Because for PbD sensorimotor skills and hand-eye-coordination are very important new training methods have to be used.

This paper proposes to use the PbD system itself as a training system to teach novices the needed skills. This seems to be a promising approach, because the main training tools are based on virtual fixtures and compliance motion which are also useful concepts for PbD systems. The DLR/KUKA Light Weight Robot with its compliance control is well suited for both tasks. Based on this robot two training setups have been described. One real life training system using augmented reality concepts and one virtual training system using virtual reality.

Acknowledgments

We would like to acknowledge support for this project from the European Commission for the funding of the Integrated Project IP-SKILLS-35005.

References

J. J. Abbott, P. Marayong, and A. M. Okamura. Haptic virtual fixtures for robot-assisted manipulation. In S. Thrun, R. Brooks, and H. Durrant-Whyte, editors, *Robotics Research: Results of the 12th International Symposium ISRR*, pages 49–64. Springer, 2007.

A. Albu-Schäffer and G. Hirzinger. Cartesian impedance control techniques for torque controlled light-weight robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 657–663, 2002.

G. Biggs and B. MacDonald. A survey of robot programming systems. In *Proceedings. Australasian Conference on Robotics and Automation*, 2003.

A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Springer Handbook on Robotics*, chapter 59. Springer, 2008.

R. Bischoff, A. Kazi, and M. Seyfarth. The morpha styleguide for icon-based programming. In *Proceedings. 11th International Workshop on Robot and Human Interactive Communication, RO-MAN 2002*, pages 482–487, 2002.

H. Bruyninckx and J. DeSchutter. Specification of force-controlled actions in the task frame formalism: a survey. *IEEE Transactions on Robotics and Automation*, 12(5):581–589, 1996.

J.R. Chen and B.J. McCarragher. Programming by demonstration - constructing task level plans in a hybrid dynamic framework. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1402–1407, San Francisco, CA, USA, April 2000.

A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *Int. Conf. on Intelligent Robots and Systems*, pages 1623–1630, Oct. 2006.

K. Diethers, T. Firley, T. Kröger, and U. Thomas. A new framework for task oriented sensor based robot programming and verification. In *Proceedings. IEEE International Conference on Advanced Robotics, ICAR 2003*, pages 1208–1214, 2003.

D. Feygin, M. Keehner, and R. Tendick. Haptic guidance: experimental evaluation of a haptic trainingmethod for a perceptual motor skill. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems HAPTICS2002*, pages 40–47, 2002.

P. Fitts. The information capacity of the human motor system in controlling the amplitude of movements. *Journal of Experimental Psychology*, 47:381–391, 1954.

G. Grunwald, G. Schreiber, A. Albu-Schäffer, and G. Hirzinger. Programming by touch: the different way of human-robot interaction. *IEEE Transactions on Industrial Electronics*, 50(4):659–666, Aug. 2003.

G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl. Dlr's torque-controlled light weight robot iii - are we reaching the technological limits now? In *IEEE Int. Conf. on Robotics and Automation*, pages 1710–1716, Washington D.C., USA, 2002.

N. Hogan. Controlling Impedance at the Man/machine Interface. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1626–1631, Scottsdale, AZ, May 1989.

R. Koeppe. *Robot Compliant Motion based on Human Skill*. PhD thesis, ETH Zürich, 2001. URL `http://elib.dlr.de/11791`.

A.B. Kuang, S. Payandeh, Bin Zheng, and F. Henigmanand C.L. MacKenzie. Assembling virtual fixtures for guidance in training environments. In *Proceedings. 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2004*, pages 367–374, 2004.

Y. Maeda, N. Ishido, H. Kikuchi, and T. Arai. Teaching of grasp/graspless manipulation for industrial robots by human demonstration. In *Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2002*, pages 1523–1528, 2002.

P. Marayong, Ming Li, A.M. Okamura, and G.D. Hager. Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In *Proceedings IEEE International Conference on Robotics and Automation, ICRA2003*, pages 1954–1959, 2001.

H. Mayer, I. Nagy, A. Knoll, E.U. Braun, R. Lange, and R. Bauernschmitt. Adaptive control for human-robot skilltransfer: Trajectory planning based on fluid dynamics. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007.

P. McGuire, J. Fritsch, J. J. Steil, F. Röthling, G. A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. In *Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2002*, pages 1082–1089, 2002.

M. Pardowitz, B. Glaser, and R. Dillmann. Learning repetitive robot programs from demonstrations using version space algebra. In *Proceedings. The 13th IASTED International Conference on Robotics and Applications, RA 2007*, 2007.

C. Preusche, R. Koeppe, A. Albu-Schäffer, M. Hähnle, N. Sporer, and G. Hirzinger. Design and haptic control of a 6 dof force-feedback device. In *Workshop: Advances in Interactive Multimodal Telepresence Systems*, München, March 2001.

L. Sciavicco and B. Siciliano. A survey on robot control technology. In *Proceedings. 4th Jordanian International Electrical and Electronics Engineering Conference*, pages 395–400, 2001.

B. Zheng, A. Kuang, F. Henigman, S. Payandeh, A. Lomax, L. Swanström, and C.L. MacKenzie. Effects of assembling virtual fixtures on learning a navigation task. *Studies in Health Technology and Informatics*, pages 586–591, January 2006.