

Interactive Redundant Robotics: Control of the Inverted Pendulum with Nullspace Motion

Günter Schreiber, Christian Ott, Gerd Hirzinger

German Aerospace Center - DLR
Institute for Robotics and Mechatronics
82230 Wessling, Germany

E-mail: {Guenther.Schreiber, Christian.Ott}@dlr.de

Abstract — In the growing field of service robotics the interaction between humans and robots is an important topic. In this paper the interactive features of the kinematically redundant DLR lightweight robot are presented. At the example of an inverted pendulum the “*interactive nullspace motion*” is introduced, where the user is able to modify the configuration as a subtask while balancing the pendulum as primary task. Different ways of nullspace interaction are shown, either contact-free by a teach-device or a position tracker, or by touching the robot, whereas the joint torque sensors measure the external touch.

1 Introduction

In the growing field of service robotics smart robots will be needed, which have enough sensorial capabilities to interact with their environment and a high ability to react to the uncertainty of their task specification and environment. The need for flexibility will be fulfilled best with a kinematically redundant robot. This paper focuses on the relationship between task execution by a kinematically redundant robot, sensors and users interaction. The robot has to balance an inverted pendulum [10, 11], while it should react to user interactions like pushing or avoiding obstacles with nullspace motions. Those nullspace motions, so called subtasks, are designed such that the user is able to modify the robots configuration without disturbing execution of the end-effector task. Several subtasks are defined, which utilize common teach devices, the joint torque sensors and position trackers as input devices. All experiments have been implemented on DLR’s new light-weight robot (figure 1)(<http://www.robotic.de/LBR/> [4]).

The paper is organized as follows: In the next section the control theory of a spherical inverted pendulum is shown, then the problem of redundant inverse kinematics and the additional subtasks is presented. After the theoretical discussion experimental results are shown.



Figure 1. DLR’s kinematically redundant seven-axis light-weight robot

2 Inverted Pendulum (Main Task)

As the main task we chose the stabilization of a spherical inverted pendulum. The pendulum is prevented from toppling due to the movements (accelerations) of the robot’s end-effector.

For the mathematical model of the pendulum a linear controller was designed which stabilizes the pendulum. This controller gives the desired accelerations of the robot’s end-effector as output signals. For the realization with the DLR’s light-weight robot it was also necessary to consider the robot’s cartesian dynamics by a proper excitation of the joint controllers.

2.1 Modeling of the Inverted Pendulum

The pendulum was modeled as a rigid body (with mass m_p and length $2l$) which stands at the robot’s end-effector (see fig. (2)).

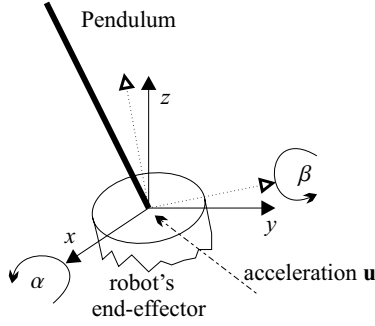


Figure 2. Generalized Coordinates for the Pendulum

The inclination of the pendulum can be described by two consecutive rotations with angles α and β about orthogonal axes (x - and y - axes of a right-handed coordinate systems).

The equations of motion for this body can be derived straight forward by utilizing Lagrange's equations with α and β as the generalized coordinates. So one gets two coupled nonlinear differential equations, which are dependent on the accelerations of the end-effector. To get a simpler model for the controller design, this nonlinear model was linearized at the unstable equilibrium point ($\alpha = 0$ and $\beta = 0$). Notice that our goal in this paper is only to stabilize the pendulum locally around the upright position and not to treat global stabilization techniques. This strategy results in two uncoupled linear models, which describe the movements of the pendulum in two vertical planes.

$$\begin{aligned}\ddot{\alpha} &= g \frac{l_{m_p}}{J} \alpha - \frac{l_{m_p}}{J} u_y \\ \ddot{\beta} &= g \frac{l_{m_p}}{J} \beta + \frac{l_{m_p}}{J} u_x\end{aligned}\quad (1)$$

Herein J denotes the inertia of the pendulum for a rotation around an axis perpendicular to the pendulum at a point of the end-effector. g is the gravitational acceleration. u_x and u_y are the accelerations of the robot's end-effector in x - and y -direction respectively.

These equations can now be interpreted as two one-dimensional inverted pendulums which can be controlled separately. Further on we will describe the controller design for the first of the two equations (for the angle α).

2.2 Controller Design

Our goal for the controller design is to reach the equilibrium point $\alpha = 0$. It is also necessary to prevent that the robot moves to the boundary of its (dextrous) workspace. Therefore the system has to be extended

by the deviation of the robot's end-effector from a desired equilibrium-position $r_{y,0}$.

$$d^2(r_y(t) - r_{y,0})/dt^2 = u_y(t) \quad (2)$$

It is assumed here, that the robot is able to follow the given (cartesian) acceleration signal $u_y(t)$ exactly without any dynamics.

This leads to the following system of ordinary differential equations

$$\dot{\mathbf{x}}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ g \frac{l_{m_p}}{J} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{x}_p + \begin{bmatrix} 0 \\ -\frac{l_{m_p}}{J} \\ 0 \\ 1 \end{bmatrix} \cdot u_y \quad (3)$$

with state

$$\mathbf{x}_p^T = \left[\alpha(t), \frac{d\alpha(t)}{dt}, r_y(t) - r_{y,0}, \frac{dr_y(t)}{dt} \right]$$

The upper left part of the systems describes the inverted pendulum and the lower right part represents the position of the end-effector.

A linear state controller structure was chosen for this plant. The controller has to be designed for the discretized version of system (3).

$$u_{y,d} = \mathbf{k}^T \mathbf{x}_{p,d}$$

For the design criterion the optimization of a quadratic cost function L_R has been chosen. This leads to the well known LQR controller (eg. [5]).

$$L_R = \sum_{k=0}^{\infty} \mathbf{x}_{d,k}^T \cdot \mathbf{Q}_x \cdot \mathbf{x}_{d,k} + q_u u_k^2 \quad (4)$$

$\mathbf{x}_{d,k}$ stands herein for the state $\mathbf{x}_p(t)$ at time $t = kT_a$. $T_a = 6ms$ is the used sample time.

For the realization of this controller it is necessary to know about the full state $\mathbf{x}_{d,k}$ at each time step. Measurement data of the inclination angle and the end-effector-position (from joint-angle measurements and a kinematical model of the manipulator) is available but noisy. The velocities have to be calculated from these measurements. Therefore linear observers (Kalman-filter) were implemented which give filtered measurement-signals additional to the needed velocities.

2.3 Consideration of the Robot Dynamics

As already mentioned before, the robot's dynamics are not considered when using equation (2). This can

only be justified by a proper excitation of the robot's joint controllers.

We are interested here in the robot's dynamics as it is revealed to the cartesian position interface. This dynamics result from the underlying joint controllers in addition with the used inverse kinematics, which is described in section 3 in detail. The used joint controllers can be regarded as a simplified implementation of the feedback linearization controller [1]. It is well known that the feedback linearization decouples the nonlinear joint dynamics and allows to define a desired cartesian behaviour. Thus, instead of modelling the nonlinear dynamics of the robot in detail, linear techniques are sufficient to get an appropriate cartesian model.

To get a model of the dynamic behavior of the robot when commanding a cartesian path, an identification procedure was performed. Starting from a point in the middle of the robot's workspace various circular paths were commanded. Due to the phase-differences between commanded and actual positions at different angular velocities, a frequency response was computed. Therefore it was assumed that the dynamics are of 4th order and can be written in the form

$$G_{assumed}(s) = \frac{1}{(s^2/\omega_r^2 + 2\xi_r/\omega_r s + 1)^2} = \frac{1}{\sum_{i=1}^4 b_i s^i} \quad (5)$$

This transfer function can be approximated by a simpler model for the relevant frequency domain by neglecting the higher order terms

$$G_{simpler}(s) = \frac{1}{\sum_{i=1}^2 b_i s^i} \quad (6)$$

An ideal compensation of this dynamics would require an inversion of the function $G_{simpler}(s)$. It is obvious that this could not be implemented in a causal matter. Instead a compensation of the form

$$G_{compensate}(s) = \frac{\sum_{i=1}^2 b_i s^i}{s^2/\omega_e^2 + 2\xi_e/\omega_e s + 1} \quad (7)$$

with proper chosen $\omega_e = 1.23\omega_r$ and $\xi_e = \xi_r$ was sufficient. Figure 3 shows the phase plots of this compensation.

3 The redundant inverse kinematic system

In the previous section it was described, how the desired end-effector trajectory for the balancing problem is obtained. In the sequel the problem of redundant inverse kinematics is reviewed and some sub-tasks for exploiting the self motion are defined. Various different authors dealt with the solution of the

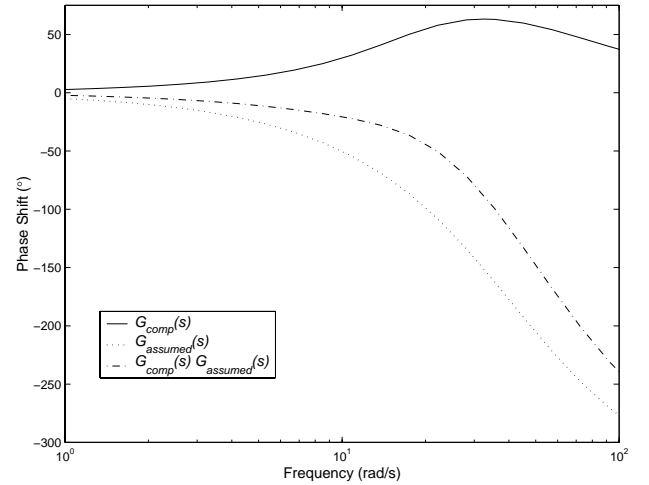


Figure 3. Compensation of the robot dynamics

redundant inverse kinematic problem (see [9] for an overview). Cartesian position and orientation \mathbf{x} of the end-effector can be described as a function of the vector of joint variables \mathbf{q} of the manipulator.

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \quad (8)$$

While equation (8) can be obtained easily, the inverse problem is crucial. In the redundant case it is generally not possible to find an inverse mapping \mathbf{f}^{-1} . Instead of constructing an inverse function $\mathbf{g}(\mathbf{x})$ with $\mathbf{f}(\mathbf{g}(\mathbf{x})) = \mathbf{x}$ analytically, the problem is often reformulated in the velocities utilizing the partial derivation of $\mathbf{f}(\mathbf{q})$.

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (9)$$

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \quad (10)$$

Due to the fact that the inverse of the non-square (analytical) Jacobian $\mathbf{J}(\mathbf{q})$ does not exist in the redundant case, the wellknown (weighted) generalized Moore–Penrose pseudo inverse \mathbf{J}^\dagger is utilized. This often proposed technique takes use of a special solution of equation (10). Optimization criteria for the redundant self motion can be added by e.g. Nullspace projection, which leads to the wellknown relation:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\dot{\mathbf{q}}_0 \quad (11)$$

The vector $\dot{\mathbf{q}}_0$ can be chosen arbitrary and is used to force a desired behavior for the null space motion.

In our current system, a linear least square based constraint optimization is utilized [8, 3, 6]. The major advantage on utilizing some constraint optimization

approach is, that physical constraints, like joint speed limits can be enforced in a natural and pretty easy way via inequality constraints. Using the following incremental formulation

$$\dot{\mathbf{q}} \approx \frac{\Delta \mathbf{q}}{T_a} \quad (12)$$

linear conditions can be chosen for the nullspace motion of the form

$$\mathbf{A} \Delta \mathbf{q} = \mathbf{a} \quad (13)$$

so the inverse kinematic problem can be formulated as an optimization problem

$$\min_{\Delta \mathbf{q}} L_{IK}(\Delta \mathbf{q}) \quad (14)$$

for the cost function

$$L_{IK}(\Delta \mathbf{q}) = \|\mathbf{A} \Delta \mathbf{q} - \mathbf{a}\|_2 \quad (15)$$

with the equality constraint as main task

$$\Delta \mathbf{x}_{EF} = \mathbf{J} \Delta \mathbf{q} \quad (16)$$

$$\Delta \mathbf{x}_{EF} = \dot{\mathbf{x}}_d T_a + k(\mathbf{x}_d - \mathbf{x}) \quad (17)$$

Herein the symbol $\|\cdot\|_2$ denotes the quadratic norm, \mathbf{x}_d the desired cartesian position and k some arbitrary positive constant. Notice that the main task forms in fact the constraint for a linear optimization problem here. Additional constraints in form of inequalities can be added to restrict the solutions for particular joints (e.g. restrictions due to hardware constraints or maximum joint movements per time step).

$$\Delta \mathbf{q} \in [\Delta \mathbf{q}_{min}, \Delta \mathbf{q}_{max}] \quad (18)$$

It should also be noted that this optimization problem leads to the pseudo inverse solution by choosing \mathbf{A} as the identity matrix and \mathbf{a} as a vector containing just zero elements.

Any particular cost function $h(\mathbf{q})$ can be minimized locally by utilizing the $n \times n$ -identity matrix $\mathbf{E}_{n \times n}$ and the gradient of the cost function, whereas n is the number of joints.

$$\mathbf{A} = \mathbf{E}_{n \times n}, \quad \mathbf{a} = -\frac{\partial h(\mathbf{q})}{\partial \mathbf{q}} \quad (19)$$

It is now possible to combine multiple behaviors for the null-space motion by choosing

$$\mathbf{A} = \begin{pmatrix} w_1 \mathbf{A}_1 \\ \vdots \\ w_l \mathbf{A}_l \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} w_1 \mathbf{a}_1 \\ \vdots \\ w_l \mathbf{a}_l \end{pmatrix} \quad (20)$$

The particular equations $\mathbf{A}_i \Delta \mathbf{q} = \mathbf{a}_i$ are weighted by the factors w_i , which define how much an error in the i^{th} equation affects the value of the cost function $L_{IK}(\Delta \mathbf{q})$. One very interesting feature of this formulation is that the main task can also be included into the cost function instead of constraining the optimization if the problem gets ill-posed, e.g. due to singular configurations [8].

3.1 Including the main task

The controller for the stabilization of the inverted pendulum gives the desired trajectories $r_{x,d}$ and $r_{y,d}$ for the end-effector movement in x - and y - direction respectively. Additional constraints have to be considered due to the mechanical mounting of pendulum (figure 4). The orientation can be described by a set of three Euler angles $\phi_1(t)$, $\phi_2(t)$ and $\phi_3(t)$, whose initial values at time $t = 0$ correspond to an orientation in which the mounting of the pendulum stands in an upright position. In order to keep the cartesian orientation of the end-effector constant, those angles have to be fixed at their initial values. So, the mounting of the pendulum will only move translational. As follows, only five degrees of freedom of the cartesian end-effector movement are needed to fulfill the stabilizing task, two for the horizontal movement and three for the end-effector-inclination. Thus, the partition of equation (17) which describes the vertical movements of the end-effector (the third row) can be removed from the constraints. So we get modified constraints to equation 14, which are stated in equation 21.

$$\Delta \mathbf{x}_{EF,mod} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{J} \Delta \mathbf{q} \quad (21)$$

$$\begin{aligned} \Delta \mathbf{x}_{EF,mod} &= \dot{\mathbf{x}}_{d,mod} T_a + k(\mathbf{x}_{d,mod} - \mathbf{x}_{mod}) \quad (22) \\ \mathbf{x}_{d,mod} &= (r_{x,d}, r_{y,d}, \phi_1(0), \phi_2(0), \phi_3(0))^T \\ \mathbf{x}_{mod} &= (r_x, r_y, \phi_1, \phi_2, \phi_3)^T \end{aligned}$$

Having $n = 7$ joints of the robot, we have two remaining (redundant) degrees of freedom which can now be utilized for additional subtasks.

3.2 Self motion by command

The redundant degrees of freedom can also be used for internal joint movements which are commanded by an external process (or the user). As stated in [7] the movements of a reference frame, which is fixed to a

particular joint, can be commanded by a 6 DOF teach device in this way without affecting the trajectory of the end-effector.

When $\Delta \mathbf{x}_m$ denotes the movements due to the teach device and \mathbf{J}_m the map, which describes the motions of the reference frame, the controlling of the redundant DOFs can be fulfilled using

$$\Delta \mathbf{x}_m = \mathbf{J}_m \Delta \mathbf{q} \quad (23)$$

$$\mathbf{A}_1 = w_1 \mathbf{J}_m, \quad \mathbf{a}_1 = w_1 \Delta \mathbf{x}_m \quad (24)$$

with a proper chosen weight w_1 . Note: The interactive subtasks can be easily converted for use with the Moore–Penrose \mathbf{J}^\dagger formalism.

3.3 Self motion by touch

By including additional sensory data it is possible to extend the behaviors. Using the robots joint torque sensors the nullspace motion can be used for reactions due to external forces.

From a model of inverse dynamics [2], the nominal joint torques $\boldsymbol{\tau}_{nom} = \boldsymbol{\tau}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ can be calculated. This is compared to the sensed torques $\boldsymbol{\tau}_{sens}$. So, the difference of this two values leads to external torques $\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{sens} - \boldsymbol{\tau}_{nom}$, which can be interpreted as user commands, while the user is touching the robots structure. This can be easily utilized for the subtasks formulation. Each of the joint movements can be weighted individually by c_i as well. The full formulation for a common weight w_2 is stated in equation (25).

$$\mathbf{A}_2 = w_2 \mathbf{E}_{n \times n}, \quad \mathbf{a}_2 = w_2 \begin{pmatrix} -c_1 \boldsymbol{\tau}_{ext,1} \\ \vdots \\ -c_n \boldsymbol{\tau}_{ext,n} \end{pmatrix} \quad (25)$$

3.4 Collision Avoidance

As a last example how to use the nullspace motion we chose the avoidance of collisions with an external object. Therefore an artificial potential field is attached to the considered external object.

Let d denote the minimal distance between the robot and the object. If this distance is smaller than a proper chosen value d_0 , then the robot should move away according to the potential field $V(d)$.

$$V(d) = \frac{1}{2}(d - d_0)^2 \quad (26)$$

For the nullspace motion an equation of the form (13) is needed. Assume, that the object moves sufficiently

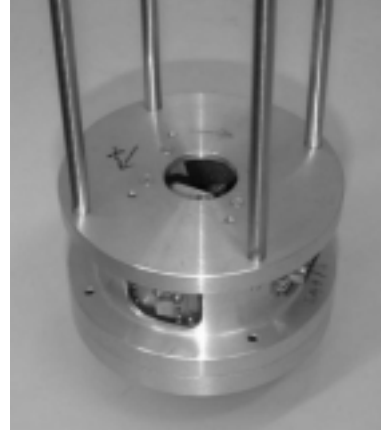


Figure 4. Mounting for the pendulum

slow, then the change of the value of $V(d)$ due to a joint movement can be formulated as follows.

$$\Delta V(d) = \frac{\partial V(d)}{\partial d} \frac{\partial d}{\partial \mathbf{q}} \Delta \mathbf{q} \quad (27)$$

Equation (27) can now be used for the formulation of another linear term in the inverse kinematics.

$$\mathbf{A}_3 = w_3 \frac{\partial V(d)}{\partial d} \frac{\partial d}{\partial \mathbf{q}}, \quad \mathbf{a}_3 = w_3 \Delta V(d) \quad (28)$$

It is obvious that collisions of the end-effector (and the pendulum as well) with the collision-object cannot be handled in this way, because only self motion has been considered herein. To avoid collisions of the end-effector too, it is necessary to allow a deviation of the desired trajectory, which can be calculated similar to (27). For interaction between the user and the collision system a position tracker is utilized.

4 Results

4.1 Experimental Setup

For the stabilizing task we used an aluminum rod pendulum of length $2l = 500mm$ and mass $m_p = 110g$. This was mounted on a sensor which measures the two inclination angles between the robots end-effector and the rod. The designed controller and inverse kinematic algorithm were implemented on a 300 MHz Power-PC processor with a sample rate of 6 msec. The tasks were performed utilizing the DLR light-weight robot (figure 1).

4.2 Stabilizing Task

The inclination of the pendulum for the starting procedure was approximately 5 degrees. This was re-

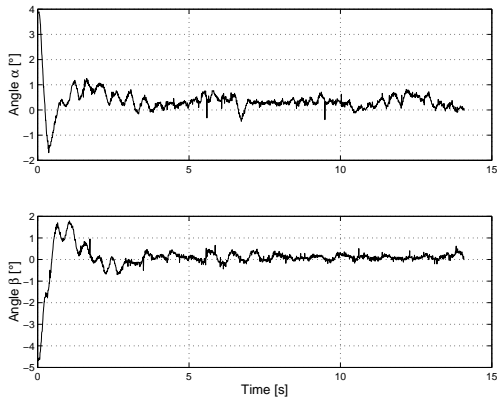


Figure 5. Inclination angles during the stabilization

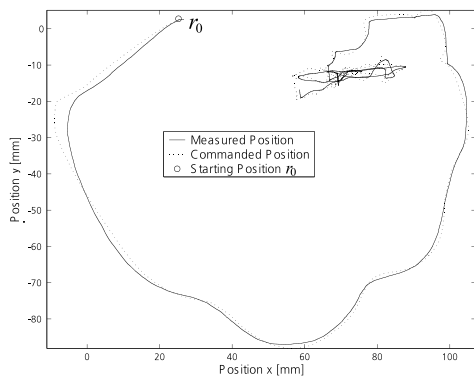


Figure 6. TCP - trajectory projected onto an horizontal plane

alized with a metal boundary-ring, on which the pendulum laid when the stabilizing task was started.

Figure 5 shows the measurements of the inclination angles α and β . As one can see, the designed controller was able to stabilize the inverted pendulum in the upright position.

In figure 6 the deviation of the position of the TCP from the equilibrium-pose r_0 is shown when being projected onto an horizontal plane. This deviation is kept bounded but is not brought to zero exactly because of the offset-errors in the measurements (Which can also been figured out in figure 5). When the given equilibrium-position was changed, the end-effector followed according to these changes while stabilizing the pendulum.

4.3 Nullspace Operations

Additional to the main task various subtasks were performed. All these subtasks were performed without



Figure 7. Nullspace Motion

affecting the stabilization task.

Figure 7 shows an online commanded (as described in section 3.2) nullspace-motion. The results from the interactive subtasks, which were described in section 3.3 and section 3.4, are presented in figure 8 and 9. In the case of null motion by touch (figure 8), the external torques are applied by the user. Also the collision avoidance is here only presented for pure nullspace movements. So only collisions of the robot's internal structure with the "collision object" (which is represented by the data glove at the user's right hand) are handled. The utilized position tracker was the Polhemus tracker. The videos showing all experiments are available online (<http://www.robotic.de/LBR/animationen.html>).

5 Conclusions

The feasibility of *interactive redundant robotics* and their usability for service robotics was shown. According to the task splitting concept in primary and secondary task, the concept was demonstrated on a real system. Interaction between the robot and a human was demonstrated, while performing end-effector tasks on the robot. The robot is able to react on users commands or touch while executing the primary task. This was shown experimentally with an inverted pendulum. This kind of robotics will be necessary in future service applications, where interactions between robots



Figure 8. Nullspace motion by touch through sensing external joint torques τ_{ext} applied by the user. Note: A human subject would seldomly succeed to balance a rod while somebody touches his elbow.

and humans are vital.

Acknowledgments

This work was granted by “BMBF–Leitprojekt MORPHA” Number ITL 0902 E.

References

- [1] A. Albu-Schäffer and G. Hirzinger. State feedback controller for flexible joint robots: A globally stable approach implemented on DLR’s light-weight robots. In *IROS 2000*, pages 1087–1093, 2000.
- [2] A. Albu-Schäffer and G. Hirzinger. Parameter identification and passivity based joint control for a 7DOF torque controlled light weight robot. In *ICRA May 2001, Seoul, Korea*, pages 2852–2858, 2001.
- [3] F.-T. Cheng, T. Chen, and Y.-Y. Sun. Resolving manipulator redundancy under inequality constraints. *IEEE Transaction on Robotics and Automation*, 10(1):65–71, Feb 1994.
- [4] G. Hirzinger, A. Albu-Schäffer, M. Hähle, I. Schaefer, and N. Sporer. A new generation of torque con-



Figure 9. Collision avoidance with nullspace motion, the colliding object is represented by the data glove with position tracker in the user’s right hand.

trolled light-weight robots. In *ICRA May 2001, Seoul, Korea*, pages 3356–3363, 2001.

- [5] K. Ogata. *Discrete-time control systems*. Prentice-Hall, Inc, 1987.
- [6] K. C. Park, P. H. Chang, and J. K. Salisbury. A unified approach for local resolution of kinematic redundancy with inequality constraints and its application to nuclear power plant. In *ICRA, April 1997, Albuquerque, New Mexico*, pages 766–773, 1997.
- [7] G. Schreiber and G. Hirzinger. An intuitive interface for nullspace teaching of redundant robots. In J. Lenarčič and M. Stanišič, editors, *Advances in Robot Kinematics*, pages 209–216, Portoroz/ Slovenia, 2000. Kluwer Academic Publishers, Dordrecht.
- [8] G. Schreiber, M. Otter, and G. Hirzinger. Solving the singularity problem of non-redundant manipulator by constraint optimization. In *IROS 1999, Kyongju, Korea*, pages 1482–1488, 1999.
- [9] B. Siciliano. Kinematic Control of Redundant Robot Manipulators: A Tutorial. *Journal of Intelligent and Robotic Systems*, 3:201–212, 1990.
- [10] K. P. Sondergeld. Balancieren eines Stabes mit einem digital gesteuerten Roboter. DFVLR-Forschungsbericht, Juli 1985.
- [11] B. Sprenger, L. Kucera, and S. Mourad. Balancing of an inverted Pendulum with a SCARA Robot. In *ICRA, April 1997, Albuquerque, New Mexico*, 1997.