Submitted to 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '01)

CleaningAssistant - A Service Robot Designed for Cleaning Tasks

F. Marrone, M. Strobel,
Research Institute for Applied Knowledge Processing FAW
Helmholtzstr. 16
89081 Ulm, Germany
{marrone,mstrobel}@faw.uni-ulm.de

Abstract— **A new service robot designed for cleaning tasks in home environments is introduced. Robot systems will work directly with people in there areas, thus placing a central importance on making interactions between people and machines as natural as possible. The main focus of this paper is twofold: First, an introduction to the system's design and to an intuitive programming approach which allows the robot to be easily used by non-experts is given. The approach is based on human gesture recognition and context sensitive interpretation. Second, a closed form solution of the robot's inverse kinematics is derived. The reduncancy of the manipulator is therein used to optimize the inverse kinematics solution in terms of manipulability.**
Keywords: service robot, man-machine-interface, inverse kinematics, manipulability.

I. Introduction

Housekeeping robot assistants focus on the employment of assistive robot systems in everyday domestic settings. There are different motivating factors for the employment of robots at home: on one side, comfort factors and a changing societal framework favor the employment of man-made personnel; on the other side, an increasing number of households include inhabitants that require physical support in day-to-day life due to sickness or age. Robot systems will work directly with people in this area, thus placing a central importance on making interactions between people and machines as natural as possible.

The robot assistant in the home should work together with the user to perform simple housework. In addition to fetch-and-carry duties, this includes tasks such as setting the table or performing cleaning tasks. A new service robot which is able to perform cleaning tasks is currently being built at the FAW. In Chapter II of this article the hardware, especially the kinematics design and the sensor setup of this cleaning assistant robot is envisaged.

Human interaction with the robot assistance system will have the commanding and teaching of the robot as its purpose, but it also offers interesting possibilities of augmenting the performance of the entire system. Speech and gestures are a human being's most natural communication channels. Interaction with the robot may further occur over versatile (multimodal), portable control devices. Our approach to an intuitive programming and interaction is highlighted in chapter III. We decided to use communication techniques based on hand gestures and optional speech which can be expected to be natural and convenient as they don't require the use of any external equipment.

The robot system should already possess a sufficient base of high-level and specialized skills to avoid expendable and tedious instruction and programming of rudimentary skills. E.g. considering the task of cleaning serveral surfaces in a kitchen it should be sufficient to show the robot which surfaces it has to clean and not the whole cleaning trajectory. Let us consider e.g. the high-level skills of the cleaning robot which are required to plan and execute a cleaning task: After the gesture based selection of the surfaces to be cleaned, *Coverage path planning* in workspace is needed to determine a path that guarantees that e.g. a tool like a sponge will pass over every point within a given area. Choset and Pignon [1] introduced a complete coverage path planning algorithm which is based on a exact cellular decomposition of the target area. Next the complete coverage path calculated in workspace has to be transformed into a valid path in the configuration space of the manipulator system. Therefore it is usefull to have a closed form solution of the inverse kinematics problem, wich is finding a valid robot configuration for a given posture (position and orientation) in workspace. Chapter IV shows the forward kinematics of the manipulator and discusses our closed form solution of the inverse kinematics problem. The reduncancy of the manipulator is therein used to optimize the inverse kinematics solution in terms

Fig. 1. *CleaningAssistant* for home environments.

of manipulability.

Navigation of the robot's mobile base is not discussed in the scope of this paper. A detailed treatment of our approach to mobile robot navigation can be found in Strobel [14].

II. Design of the *CleaningAssistant*

The *CleaningAssistant* robot, shown in Figure 1 consists of a mobile base and a manipulator on top of it. The manipulator joints as well as the mobile base are built of modular drive components. Two independently controlled drive components are used to built a differential drive system for the mobile base. The differential drive system is a cheap and robust drive system, but it is underlying a so called non-holonomic constraint which complicates motion planning.

A 7 DOF manipulator is based on a vertical linear axis with the goal to enhance the vertical workspace of the system. Next a SCARA-like chain of revolute joints are mounted on the linear axis. An additional degree of freedom is used to switch between the horizontal and vertical arrangement of the SCARA-like chain. Intermediate configurations are also allowed. The advantage of this arrangement is that it allows to use the advantages of the horizontally mounted chain (low energy consumption and high dynamics for horizontal arm movements) without being limited to only horizontal movements. The end-effector is either a two-finger gripper or a special end-effector, e.g. for cleaning non-textile surfaces like working places in kitchens or sinks.

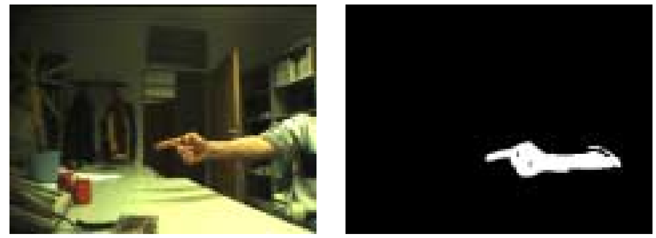
Sensory feedback is provided by a) a new kind of compliant force-torque-sensor, developed at the German Aerospace Center (DLR) (see Meusel and Hirzinger [9]) mounted between the wrist and the end-effector of the manipulator, b) a 2D range laser-scanner (Sick LMS 200) is used for position estimation as well as for obstacle detection and avoidance while navigating the mobile base, c) a trinocular stereo-vision system (Triclops) for gesture and object recognition and localization, d) a touchscreen (Elotouch) for additional gesture input, e.g. for the qualitative path specification or object selection in a displayed scene representation.

The central control unit is based on an industrial PC running the operating system LINUX and the software framework *SmartSoft* [12] which is especially designed for the communication between sensors, actors and planning components in the field of robotics. SmartSoft is based on a client-server architecture and a few simple communication primitives which provide unified interfaces. Thus modules can be combined in a flexible way to implement complex operations.

III. Intuitive Programming Approach

According to our conviction, natural instruction, interaction and programming of robots will play a decisive role in the entering of such systems in home environments. Therefore we decided to use techniques based on hand gestures and optional speech which can be expected to be natural and convenient as they don't require to use any external equipment.

In order to avoid expendable and tedious instruction and programming of rudimentary skills, the robot system should already possess a sufficient base of high-level and specialized skills. E.g. considering the task of cleaning serveral surfaces in a kitchen it should be sufficient to show the *CleaningAssistant* which surfaces it has to clean and maybe give information about how to do it, but not demonstrate the whole cleaning trajectory.



(a) (b)

Fig. 2. a) captured color image b) extracted human gesture

Let us consider now the task of showing a *CleaningAssistant* which surfaces it has to clean e.g. in a kitchen. Our approach for this purpose is based on the recognition and context-sensitive interpretation of human pointing gestures as follows: First a human is pointing on the surfaces the robot has to clean or shows the robot a specific surface area by means of a sequence of pointing gestures. We concentrated on visual recognition of hand gestures because they are preferable to speech in a number of situations, for example in noisy environments or when communicating quantitative information and spatial relations. To distinguish a hand region from the background a combined color- and disparity-based approach is used. Figure 2a shows a captured gesture image. A subsequent color classification stage checks whether each of the color segments are skin colored using a lookup table defined on hue and saturation of the color values. An additional figure-ground separation stage uses disparity information gained by trinocular stereo imaging. The disparity image is segmented based on a-priori information about the expected disparity range of the person. Finally color- and disparity segmentation are combined to provide regions of interest for hand regions. After some simple pre-processing, blob-labelling and validation of candidate regions the view aspect is normalized with respect to distance and orientation of the gesture. Contours of the normalized hand regions are extracted and approximated (see Figure 2b). To classify a gesture the hand region is compared to a set of gesture models using second and third order normalized moments of the extracted regions [5] and a Hausdorff-distance based contour matching [6]. The classifier uses a nearest-neighbor-criteria to select the most similar gesture model. If the difference between model and extracted gesture is too large the gesture is rejected.

Interpretation of a pointing gesture is done context-sensitive. Considering the example of selecting a surface to clean the system "knows" that it has to find a corresponding surface to the detected pointing gesture. Position and orientation of the pointing hand gesture are calculated as described above. The corresponding surface is determined with the help of the spacial pointing information, an internal three dimensional representation of the scene (especially the geometrical environment description) and algorithms known from computer graphics [4]. In a first stage, it is assumed that a priori information about the environment (here: house and arrangement of the furniture) is available. This restriction can be relaxed in a further stage, in which the system will learn interactively the object distribution in the environment and its coherence. So far, we are using polyhedrons as fundamental geometrical modelling primitive. A polyhedron is considered as single convex solid. Based on polyhedrons we are using a hierarchical descriptions, called *polytrees* here, for convex or nonconvex geometric objects. The depth of a polytree and the branching factor of the individual nodes are arbitrary. The leaves of a polytree corresponds to the underlying convex pieces of solid.

In place of showing items, positions, surfaces etc. in real world, communication techniques based on gesture input via touchscreen are considered too. This will favorably be used when showing a path in an internal representation of the environment, e.g. in a two dimensional floor plan, or when selecting e.g. items in pictures from remote locations. We consider the problem of transforming a given input sequence into a valid path on a underlying graph structure. This could be e.g. a topological graph of the environment in which vertices are representing locations (rooms etc.) and edges are representing connections between them which are possible to traverse by the *CleaningAssistant*. The transformation we developed (see Kämpke [8]) uses a so called *Graph Voronoi Partitioning* technique together with an appropriate matching of touch points to edges and vertices of the graph. Disontinious edge/vertex seqences have to be further processed in order to get a valid path in the graph. This sequence transformations are supposed to support issues of man-machine interaction which implies the lack of an ultimate formal design objective.

IV. Manipulator Kinematics

In this chapter we describe the development of a closed-form solution of the manipulator's inverse kinematics problem. First, the forward kinematics parameterization using Denavit-Hartenberg convention [3] is given in chapter IV-A. The result will be used in chapter IV-B in which our algorithm for the solution of the inverse kinematics problem is explained. Finally chapter IV-C deals with the optimization method employed, using the redundancy of the manipulator to increase the systems performance in terms of manipulability.

A. Forward Kinematics

We describe the forward kinematics of the manipulator in terms of Denavit-Hartenberg notation [3] because of the portability of this method which is well known both in the academic and in the industrial field. The Denavit-Hartenberg parameterization of the manipulator is shown in Table I.

joint	ϕ_i [deg]	α_i [deg]	a_i [mm]	d_i [mm]	type
1	-90	-90	175	185	prismatic
2	0	90	0	260	revolute
3	90	0	288	0	revolute
4	-90	-90	0	-160	revolute
5	-90	-90	0	400	revolute
6	0	90	0	0	revolute
7	0	0	0	180	revolute

TABLE I
Denavit-Hartenberg parameters

The 7 DOF (PRRRRRR) manipulator of the *CleaningAssistant* was designed with the intention to use the advantages of the horizontally mounted chain achieved by joints 3 and 4 but with the ability, given by joints 1 and 2, of enhancing the vertical workspace. Finally we added three revolute joints on the top of the kinematics chain in order to get a spherical wrist. This spherical wrist is important in our derivation of a closed-form solution for the inverse kinematics problem. In fact for our purpose it is useful to divide the manipulator in a part called *arm* which consists of the first four joints (PRRR) and a part called *wrist* which consists of the last three joints (RRR).

The joint space Q is defined as the Cartesian product between each individual joint space and $\theta \in Q$ is called a joint configuration. SE(3) is defined as the special Eucledean group (see e.g. Murray, Li, Sastry [10]) which members are a combination of a three-dimensional translation and rotation. The mapping ${}^0_7\mathbf{T}(\theta) : Q \to SE(3)$ which describes the forward kinematics of the whole manipulator can be subdivided as follows:

$$ {}^0_7\mathbf{T}(\theta) = {}^0_4\mathbf{T}(\theta_1, \theta_2, \theta_3, \theta_4) \cdot {}^4_7\mathbf{T}(\theta_5, \theta_6, \theta_7) \quad (1) $$

with the configuration vector $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7]^T$ and where

$$ {}^0_4\mathbf{T} = \begin{bmatrix} s_{34} & 0 & c_{34} & d_2 + a_3 s_3 \\ -c_2 c_{34} & s_2 & c_2 s_{34} & -s_2 d_4 - a_3 c_2 c_3 - a_1 \\ -s_2 c_{34} & -c_2 & s_2 s_{34} & \theta_1 - a_3 s_2 c_3 + c_2 d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

$$ {}^4_7\mathbf{T} = \begin{bmatrix} c_5 c_6 c_7 - s_5 s_7 & -c_5 c_6 s_7 - s_5 c_7 & c_5 s_6 & c_5 s_6 d_7 \\ s_5 c_6 c_7 + c_5 s_7 & -s_5 c_6 s_7 + c_5 c_7 & s_5 c_6 & s_5 s_6 d_7 \\ -s_6 c_7 & s_6 s_7 & c_6 & d_5 + c_6 d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

are the homogeneous transformation matrix of the wrist as

$$ {}^{\mathbf{i}}_{\mathbf{j}}\mathbf{T} = \left[\begin{array}{c|c} {}^{\mathbf{i}}_{\mathbf{j}}\mathbf{R} & {}^{\mathbf{i}}_{\mathbf{j}}\mathbf{p} \\ \hline \mathbf{0} & 1 \end{array} \right] $$

describing respectively the forward kinematics of the arm. ${}^{\mathbf{i}}_{\mathbf{j}}\mathbf{R}$ is the rotation matrix and ${}^{\mathbf{i}}_{\mathbf{j}}\mathbf{p}$ is the position of the jth frame in terms of the ith frame coordinates. For a compact notation we use here and in the following the notation c_i and s_i instead of $\cos(\theta_i)$ and $\sin(\theta_i)$ and c_{ij} and s_{ij} instead of $\cos(\theta_i + \theta_j)$ and $\sin(\theta_i + \theta_j)$.

B. Closed Form Solution of the Inverse Kinematics

The inverse kinematics problem of a serial manipulator is a very significant and important problem for computer controlled robots and in fact in the literature can be found many attempts to solve it. Our method starts from the idea of Pieper [11] who simplifies the problem by subdividing it into two simpler subproblems, but it takes a different way for realizing this idea and especially for handling the redundancy.

In chapter IV-A it was shown how the kinematic chain can be subdivided into *arm* and *wrist* to reach this aim. In fact the spherical wrist has the very important characteristic to have three axis of rotation intersecting in only one point. This point, which we call $\mathbf{p^w}$ in the following, is the keystone of our solution: Given the desired position and orientation of the end-effector, $\mathbf{p^w}$ can be calculated by geometrical consideration only. This means that we must use the first four joints only to reach this point, disregarding the orientation of the endeffector which will be independently obtained later using the 3 DOF of the spherical wrist.

From this idea we establish the following algorithm which also explains how the redundancy of the manipulator is treated.

Step 1: Let $p_{des} = [p_{x,des}\, p_{y,des}\, p_{z,des}]^T$ be the desired position and

$$ \mathbf{R_{des}} = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} $$

the desired orientation of the end-effector.

Step 2: Calculate the position of the wrist $\mathbf{p^w_{des}}$ using the equation:

$$ \mathbf{p^w_{des}} = \mathbf{p_{des}} - d_7 \cdot [a_x\; a_y\; a_z]^T \tag{4} $$

In fact, $\mathbf{p^w}$ is the difference between the vector $\mathbf{p}$ and the vector which has direction $\mathbf{a}$ and length equal to d_7 as shown in Figure 3.

Step 3: Solve the inverse kinematics of the arm using the equation $\mathbf{p^w} = \mathbf{p^w_{des}}$ and taking into account that $\mathbf{p^w} \equiv \mathbf{O_5}$, where $\mathbf{O_5}$ is the origin of the fifth link frame which by definition is the last column of ${}^{\mathbf{0}}_{\mathbf{5}}\mathbf{T}$ and therefore we can calculate it as

$$ \mathbf{p^w} = \mathbf{O_5} = {}^{\mathbf{0}}_{\mathbf{4}}\mathbf{T} \cdot {}^{\mathbf{4}}_{\mathbf{5}}\mathbf{p} $$

where ${}^{\mathbf{4}}_{\mathbf{5}}\mathbf{p} = [0\, 0\, d_5\,]^T$ is part of the last column of ${}^{\mathbf{4}}_{\mathbf{5}}\mathbf{T}$. It's obvious that $\mathbf{p^w}$ depends on θ_1, θ_2, θ_3 and θ_4. So we have to

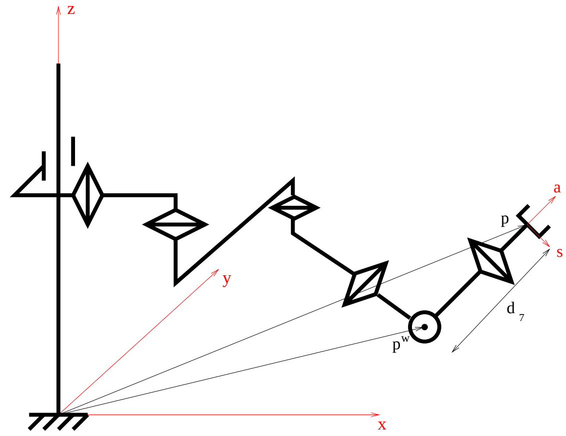


Fig. 3. Manipulator configuration

solve the following set of three equations in four unknowns, this means that an infinite set of solutions is expected:

$$ d_5(c_3c_4 - s_3s_4) + a_3s_3 + d_2 = p^w_{x,des} \tag{5} $$
$$ d_5c_2(c_3s_4 + s_3c_4) - s_2d_4 - a_1 - a_3c_3c_2 = p^w_{y,des} \tag{6} $$
$$ d_5s_2(c_3s_4 + s_3c_4) + c_2d_4 - a_3c_3s_2 + \theta_1 = p^w_{z,des} \tag{7} $$

We can express eq. 5-7 as follows:

$$ (a_3 - d_5s_4)s_3 + (d_5c_4)c_3 + (d_2 - p^w_{x,des}) = 0 \tag{8} $$
$$ (-d_4)s_2 + (d_5s_{34} - a_3c_3)c_2 + (-a_1 - p^w_{y,des}) = 0 \tag{9} $$
$$ p^w_{z,des} + a_3c_3s_2 - c_2d_4 - d_5s_2s_{34} = \theta_1 \tag{10} $$

It's obvious that in eq. 5, where there are only two unknowns (θ_3 and θ_4), it's possible to express one as a in function of the each other. Our choice was to express $\theta_3 = \theta_3(\theta_4)$ and we obtained eq. 8. So we can solve this equation using the $atan2$ function, and then use this result to solve the next equation (eq. 9) by the same function. Finally we use the results of eq. 8 and eq. 9 to solve eq. 10. Which results in:

$$ \theta_3 = 2atan2(d_5s_4 - a_3 \pm \sqrt{\Delta_3}, d_2 - d_5c_4 - p^w_{x,des}) \tag{11} $$
$$ \theta_2 = 2atan2(d_4 \pm \sqrt{\Delta_2}, a_3c_3 - d_5s_{34} - a_1 - p^w_{y,des}) \tag{12} $$
$$ \theta_1 = p^w_{z,des} + a_3c_3s_2 - c_2d_4 - d_5s_2s_{34} \tag{13} $$

where $\Delta_3 = a_3^2 + d_5^2 - 2a_3d_5s_4 - (d_2 - p^w_{x,des})^2$ and $\Delta_2 = d_4^2 + (d_5s_{34} - a_3c_3)^2 - (a_1 + p^w_{y,des})^2$. Notice that to obtain real solutions Δ_3 and Δ_2 must be greater or equal to zero. These two constraints will be constraints for θ_4 and θ_3. In fact $\Delta_3 \geq 0$ means $a_3^2 + d_5^2 - 2a_3d_5s_4 - (d_2 - p^w_{x,des})^2 \geq 0$ and hence

$$ s_4 \leq \underbrace{\frac{a_3^2 + d_5^2 - (d_2 - p^w_{x,des})^2}{2a_3d_5}}_{s_{4lim}} $$

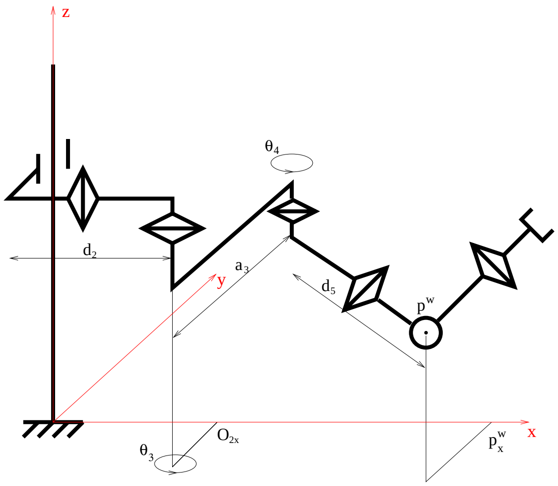


Fig. 4. Constraint along x-axis

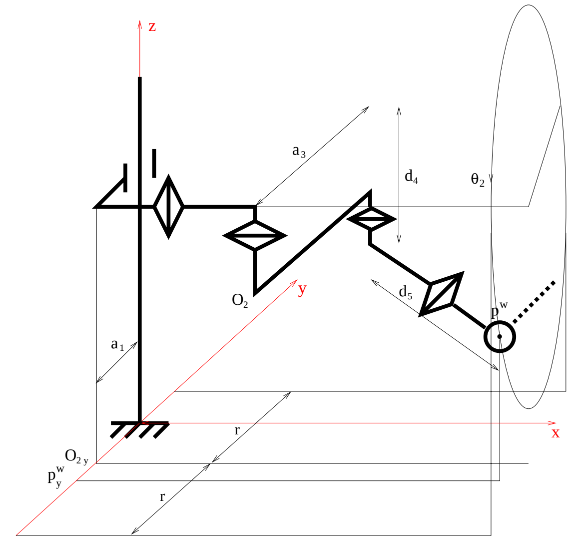


Fig. 5. Constraint along y-axis

Therefore, we are free to choose θ_4 according to the constraints

$$\theta_4 \begin{cases} \in [0, 2\pi] & \text{if } s_{4lim} \geq 1 \\ \in [-\pi - \theta_{4lim}, \theta_{4lim}] & \text{if } -1 \leq s_{4lim} < 1 \\ \in \emptyset \text{ (no solution)} & \text{if } s_{4lim} < -1 \end{cases} \quad (14)$$

where $\theta_{4lim} = \arcsin s_{4lim}$.

Similarly, θ_3 must satisfy the constraint $\Delta_2 \geq 0$ which yields:

$$d_4^2 + (d_5 s_{34} - a_3 c_3)^2 - (a_1 + p^w_{y,des})^2 \geq 0 \quad (15)$$

To understand the physical meaning of each constraint we consider limit conditions. There is no solution if $s_{4lim} < -1$ or if $\Delta_2 < 0$. Those with simple algebraic computation can be expressed as:

$$(a_3 + d_5)^2 < (d_2 - p_x^w)^2 \quad (16)$$

$$d_4^2 + (d_5 s_{34} - a_3 c_3)^2 < (a_1 + p_y^w)^2 \quad (17)$$

The meaning of eq. 16 is that p_x^w cannot be greater than the maximum extension of the arm along the x-axis ($a_3 + d_5 + d_2$) in fact this extension depends only on θ_3 and θ_4 that means that O_{2x} is fixed to d_2 and therefore we can consider the distances referring to it. (see Figure 4). The meaning of eq. 17 is that the distance of p^w from O_{2y} along y-axis ($(a_1 + p_y^w)^2$) cannot be greater than the maximum extension of the arm along the y-axis that is, when θ_3 and θ_4 are fixed, the radius r of rotation of the arm due to θ_2 ($d_4^2 + (d_5 s_{34} - a_3 c_3)^2$) (see Figure 5).
Notice that the constraint along the z-axis depends only on the physical limitation of θ_1. In conclusion the solution of the inverse kinematics of the arm is made of these steps:
For every θ_4 chosen according to eq. 14 do:

- Calculate the values of θ_3 by eq. 11;
- Consider only those values which satisfy eq. 15;
- Calculate the values of θ_2 by eq. 12;
- Calculate the values of θ_1 by eq. 13.

Step 4: Calculate ${}^0_4\mathbf{R}$ inserting the joint-values found (θ_1, θ_2, θ_3, θ_4) in eq. 2;
Step 5: Calculate ${}^4_7\mathbf{R_{des}}$.
As known

$$\mathbf{R} = {}^0_4\mathbf{R} \cdot {}^4_7\mathbf{R}$$

hence

$${}^4_7\mathbf{R} = ({}^0_4\mathbf{R})^{-1} \cdot \mathbf{R}$$

but $({}^0_4\mathbf{R})^{-1} = ({}^0_4\mathbf{R})^{\mathbf{T}} = {}^4_0\mathbf{R}$, which yields

$${}^4_7\mathbf{R_{des}} = {}^4_0\mathbf{R} \cdot \mathbf{R_{des}} \quad (18)$$

Step 6: Solve the inverse kinematics of the wrist by the equation:

$${}^4_7\mathbf{R}(\theta_5, \theta_6, \theta_7) = {}^4_7\mathbf{R_{des}} \quad (19)$$

finding the last three joint-values.
In fact the rotation matrix of this spherical wrist ${}^4_7\mathbf{R}$ coincides with the Euler Z-Y-Z matrix, therefore the solution is known (See e.g. Sciavicco-Siciliano [13]): Given

$${}^4_7\mathbf{R} = \begin{bmatrix} n_x^4 & s_x^4 & a_x^4 \\ n_y^4 & s_y^4 & a_y^4 \\ n_z^4 & s_z^4 & a_z^4 \end{bmatrix}$$

we obtain the following two solutions. For $\theta_6 \in [0, \pi]$

$$\begin{aligned} \theta_5 &= atan2(a_y^4,\ a_x^4) \\ \theta_6 &= atan2(\sqrt{(a_x^4)^2 + (a_y^4)^2},\ a_z^4) \\ \theta_7 &= atan2(s_z^4,\ -n_z^4) \end{aligned} \quad (20)$$

and for $\theta_6 \in [-\pi, 0]$

$$\begin{aligned} \theta_5 &= atan2(-a_y^4,\ -a_x^4) \\ \theta_6 &= atan2(-\sqrt{(a_x^4)^2 + (a_y^4)^2},\ a_z^4) \\ \theta_7 &= atan2(-s_z^4,\ n_z^4) \end{aligned} \quad (21)$$

Step 7: Choose the *"optimum"* $\theta \in \mathbf{Q}$ according to an *optimization function* (see next chapter IV-C).

We can notice that the steps 1-3 use the 4 DOF of the arm to reach the goal position p_{des}, the following steps 4-6 use the 3 DOF of the wrist to reach the desired orientation $\mathbf{R_{des}}$ and finally the last step 7 an optimization algorithm.

C. Optimization for Increased Smoothness and Manipulability

The reduncancy of the manipulator (here especially the choice of θ_4) is used to optimize the inverse kinematics solution in terms of manipulability. In a first approach, we choose the goal function

$$f_{cp}(\theta) = \sum_{i=1}^{7} w_i(\theta_i - \theta_{i,act})^2 \tag{22}$$

for the optimization with w_i is the weight and $\theta_{i,act}$ is the actual value of the i^{th} joint. The function, which we called *closer-posture* function is minimizing the joint movement allows us to decide which joint must be used more than the others. This is achieved by changing the weight of the appropriate joint to enhance particular aims as e.g. reducing the movement of the prismatic joint or increasing the movement of θ_3 and θ_4 to have a SCARA-type behavior.

We extended the above mentioned optimization law by including the *manipulability* theory of Yoshikawa [16] and his manipulability measure w defined (see Yoshikawa [15]) as

$$w = \sqrt{\det J(\theta) J^T(\theta)}$$

where $J(\theta)$ is the Jacobian of the manipulator when the position and orientation of the end-effector are taken as task vector. Thus $w(\theta)$ is a quantitative measure of the "mobility" of the manipulator in fact $w = 0$ only when the manipulator is in a singular state therefore maximizing w we increase the distance from the singular states. The new goal function is the sum of the closer-posture function and the inverse of the manipulability

$$f(\theta) = \frac{f_{cp}}{f_{cp,max}} + \frac{1}{w} \tag{23}$$

.

but to give more emphasis to the last one we normalized the first obtaining that the closer-posture is minimized only when we are in a region with high manipulability. Instead of when we are in a region with low manipulability the manipulability is mainly increased. The manipulability plays an important role and with this optimization function we found good results (smooth trajectory for each joint, see Figure 6) in solving the inverse kinematics problem for a trajectory given in workspace.

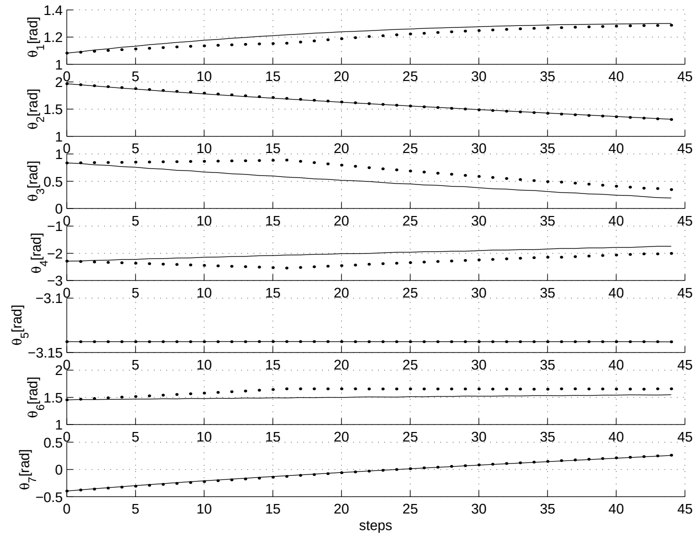


Fig. 6. Joints value trajectory

Acknowledgments

This work was supported by the German Department for Education and Research (BMB+F) under grant no. 01 IL 902 F6 as part of the project MORPHA and by the LEONARDO DA VINCI European Community Programme.

References

[1] Choset, H., Pignon, P., *Coverage Path Planning: The Boustrophedon Cellular Decomposition*, Proceeding of the Field and Service Robotics Conference FSR'97, 1997.
[2] Comaniciu D., Meer, P., *Robust Analysis of Feature Spaces: Color Image Segmentation*, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 750-755, 1997.
[3] Denavit, J., Hartenberg, R.S., *A Kinematic Notation for Lower-Pair Mechanism Based on Matrices*, Journal of Applied Mechanics, pp.215-221, June 1955.
[4] Foley, van Dam, Feiner, Huges, *Computer Graphics - Principles and Practice*, Addision- Wesley Publishing Company, 1996.
[5] Hu, M., Visual pattern recognition by moment invariants, IRE Trans. Information Theory, Vol. IT-8, Num. 2, 1962.
[6] Huttenlocher, D., Noa, J., Rucklidge, W., *Tracking Non-Rigid Objects in Complex Scenes*, Cornell University Computer Science Dept. Tech Report, CUCS TR-92-1320, 1992.
[7] Jain, R., Kasturi, R., Schunk, B., *Machine Vision*, McGraw-Hill Inc., New York, 1998.
[8] Kämpke, T., *Interfacing graphs*, to appear, 2000.
[9] Meusel,P., Hirzinger, G., *Compliant Force-Torque-Sensor for sensory-based Assembly*, Information Sheet at (http://www.robotic.dlr.de/COMPLIANCE/), 1999.
[10] Murray, R., Li, Z., Sastry, S., *A mathematical introduction to robotic manipulation*, CRC press, 1994.
[11] Pieper, D.L.,*The Kinematics of Manipulators Under Computer Control*, memo. AIM 72, Stanford Artificial Intelligence Laboratory, 1968.
[12] Schlegel, C., Wörz, R., *The Software Framework SmartSoft for Implementing Sensorimotor Systems* , IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS '99, pp. 1610-1616, 1999.
[13] Sciavicco, L., Siciliano, B., *Robotica Industriale*, McGraw-Hill, 1995.
[14] Strobel, M., *Navigation in Partially Unknown, Narrow, Cluttered Space*, IEEE International Conference on Robotics and Automation, Detroit, USA, May 1999.
[15] Yoshikawa, T., *Analysis and Control of Robot Manipulators with Redundancy*, Preprints 1^{st} Int. Symp. Robotics Res., Bretton Woods, N.H., Aug.28-Sept.2, 1983.
[16] Yoshikawa, T., *Manipulability of Robotic Mechanisms*, International Journal of Robotics Research, Vol.4, No. 2, pp. 3-9, 1985.