New Inverse Kinematics Algorithms Combining Closed Form Solutions With Nonlinear Optimization for Highly Redundant Robotic Systems

Rainer Konietschke and Gerd Hirzinger

Institute of Robotics and Mechatronics

German Aerospace Center (DLR)

82234 Wessling, Germany

`rainer.konietschke@dlr.de`

Abstract— This paper presents inverse position kinematics algorithms with real time capability for Justin, a robotic system with high redundancy and many degrees of freedom. The combination of closed form solutions for parts of the kinematic chain embedded in a nonlinear equation solver is shown to be advantageous. The algorithms are evaluated with the DLR service robot Justin both in simulation and reality. Calculation times of 1 ms are achieved, including various optimization criteria for redundancy resolution. In case only a single arm with 7 **DoF** is considered, fast calculation time of $250 \, \mu s$ is reached. With inclusion of an iterative step, reachability can be shown in more than $99\,\%$ of the calculations regardless of the initial guess. The problem of weighting in multi-criteria optimization problems remains, though in the chosen approach the tool tip position is never compromised by other criteria due to the partially closed form solution. The presented algorithm can be applied to inverse position kinematics for all manipulators with serial or tree structure and redundant joints in case closed form solutions are available for parts of the kinematic chain.

Fig. 1. The *table-mounted Justin*. The joints $q_{1..18}$ relevant to the described inverse kinematics algorithm are shown.

I. INTRODUCTION

Inverse position kinematics for robotic systems with many degrees of freedom (DoF) and high redundancy that provide the joint angles to position one or more end effectors in free space are still an open issue. Namely, computation time in control applications is often too high to reach reasonable rates, and convergence in case initial guesses are not available is often weak [3], [8], [9]. This paper presents a new method that combines a nonlinear optimization algorithm with closed form solutions for parts of the kinematic chain. This way, the joints to reach given TCP positions and orientations are solved exactly while additional criteria are combined by weighting factors and optimized using the robot redundancy. Test bed of the derived algorithms is the upper body of the DLR Justin robot, an experimental system for two-handed manipulation shown in Fig. 1 and Fig. 2.

The algorithms are compared with a standard method for inverse position kinematics that uses a nonlinear optimization algorithm and weighted optimization criteria. Comparison criteria are the computation time, the inclusion of criteria for redundancy resolution and the ability to provide valid solutions in absence of initial guesses.

The paper is organized as follows: Sect. II presents the developed methods, and performance experiments are shown in Sect. III. Section IV concludes the paper and points out further research directions.
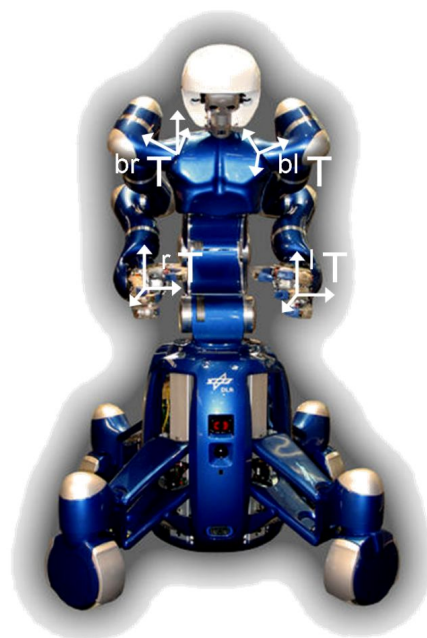

Fig. 2. The *mobile Justin* is mounted to a mobile platform. The upper body is kinematically similar to the *table mounted Justin*. In the Figure, the base and tool tip frames of left and right arm are depicted.

II. COMBINED CLOSED FORM SOLUTION WITH NULLSPACE OPTIMIZATION

This Section describes the developed inverse kinematics algorithm. The relevant kinematics of the Justin robot are described, and the closed form solutions are explained in detail. The general structure of the inverse kinematics algorithm is then presented, and the considered optimization criteria are depicted.

A. Relevant kinematics of the Justin experimental system

The experimental system Justin is shown in Fig. 1 (*table-mounted Justin*) [1] and in Fig. 2 (*mobile Justin*). The upper body consists of the torso with 3 DoF $\mathbf{q}_{\text{Torso}} = q_{1...3}$, two DLR light weight arms in right and left configuration with 7 DoF each (Joints $\mathbf{q}_{\text{RightArm}} = q_{5...11}$ and $\mathbf{q}_{\text{LeftArm}} = q_{12...18}$), two DLR hands, and a 3D-Modeller as head. See Fig. 1 for the joint notation and Fig. 2 for the tool center points (TCP) relevant for the presented inverse kinematics algorithm. The joint

$$q_4 = g(\mathbf{q}) \tag{1}$$

is a passive joint, depending primarily on joints q_2 and q_3, but also on the tendon tensions that vary with Justin's pose and thus with all joints. Goal of the inverse position kinematics algorithm in scope of this paper is to find the joint configuration

$$\mathbf{q} = \left[\mathbf{q}_{\text{Torso}}, \mathbf{q}_{\text{RightArm}}, \mathbf{q}_{\text{LeftArm}}\right] \in R^{17}$$

with a given initial configuration $\mathbf{q}_{\text{init}}$ such that the left and right TCPs are reached in the task space:

$$f(\mathbf{q}) = \left[{}^{\text{r}}\mathbf{T}, {}^{\text{l}}\mathbf{T}\right] ,$$

and the passive joint condition (1) is fulfilled. The task nullspace is used for optimization of further criteria such as the distance from joint limits, singularities, and collisions. The taskspace is 12-dimensional (positions and orientations of the two TCPs), and the joint space is 17-dimensional, leaving thus 5 dimensions for the task nullspace.

B. Closed form solution for the Justin arms

For the Justin arms, three axes intersect both in the shoulder and in the wrist. With Pieper's method [10] it is possible to calculate six of the seven joint angles in closed form, while the remaining joint is held fixed, denoted q_{fix} hereafter. Different algorithms s_i are possible according to which joint i is held fix. If a joint is held fixed during calculation of the closed form solution, additional algorithmic singularities are encountered: using the formula of Cauchy-Binet, a singular configuration of a 7-DoF robot occurs if

$$\left|\mathbf{J}\mathbf{J}^{\mathbf{T}}\right| = \sum_{i=1}^{7} \left|\mathbf{J}^i\right|^2 = 0,$$

with $\mathbf{J}$ the robot Jacobian and $\mathbf{J}^i$ the i-th minor of the Jacobian, obtained by omitting column i of the matrix $\mathbf{J}$. Hence the singular configurations of a 7-DoF robot are similar to the conjunction of all singular configurations obtained with one of the joints i fixed. Therefore the number of singular configurations is obviously lower for the 7-DoF robot than for any of the 6-DoF robots corresponding to the Jacobians $\mathbf{J}^i$. Thus the closed form solution that is not singular has to be chosen according to the initial joint angles q_{init}. Furthermore, a solution does not exist for all values of q_{fix}. This is shown in Fig. 3 (top): for a complete nullspace motion disregarding the joint limits, the joint $q_{\text{fix}} = q_3$ remains in the intervals

$$q_{\text{fix}} \in [68.76°, 111.24°] \cup [-111.24°, -68.76°] .$$

This is reflected in the closed form calculation for joint q_2:

$$q_2 = -\frac{s_3 s_4 d_5 \pm \sqrt{k}}{z + c_4 d_5 + d_3}, \tag{2}$$

with $s_i = \sin(q_i)$, $c_i = \cos(q_i)$, d_i segment lengths, z the z-direction of the right TCP ${}^{\text{r}}\mathbf{T}$ in coordinates of the right base ${}^{\text{br}}\mathbf{T}$, and

$$k = s_4^2 s_3^2 d_5{}^2 - z^2 + 2 d_5 c_4 d_3 + d_5{}^2 c_4^2 + d_3{}^2. \tag{3}$$

The term k is plotted in Fig. 3 (bottom) as a function of q_3 during a nullspace motion. Since from (2) the condition $k > 0$ needs to be fulfilled, not all values for q_{fix} are possible. This is also shown in Fig. 3. Similar conditions exist for all algorithms s_i. These conditions and the individual singularities of each algorithm s_i lead to criteria for the choice of the algorithm s_i such that (a) no algorithmic singularities occur and (b) the joint angle q_{fix} is within the range of the nullspace.

The calculation speed for the closed form solution without nullspace optimization is very fast with less than $80\,\mu s$. In the next Subsection, the complete inverse kinematics algorithm is described. For the Justin system, the arm angles $\mathbf{q}_{\text{RightArm}\backslash q_{\text{fix}}}$ and $\mathbf{q}_{\text{LeftArm}\backslash q_{\text{fix}}}$ are solved in closed form.

C. Algorithm overview

The developed inverse kinematics algorithm is depicted in Fig. 4. In the initialization step, the closed form solution algorithms s_i are chosen according to the initial joint angles $\mathbf{q}_{\text{init}}$. This way, the parameters available for nullspace optimization $\mathbf{q}_{\text{opt}}$ are determined.[1] For the Justin system,

$$\mathbf{q}_{\text{opt}} = \left[q_1, q_2, q_3, q_{\text{fixRight}}, q_{\text{fixLeft}}\right] .$$

A Levenberg-Marquardt optimization [6] with included closed form solutions is then performed: unless the terminating condition is reached, the following loop is executed:

1) *Update optimization parameters:* The Levenberg-Marquardt algorithm selects new values for the optimization parameters $\mathbf{q}_{\text{opt}}$.[2]

---

[1]The notation $\mathbf{q}_{\text{opt}}$ is chosen since all optimization parameters are joint angles.

[2]In the first step, $\mathbf{q}_{\text{opt}}$ is extracted from $\mathbf{q}_{\text{init}}$.
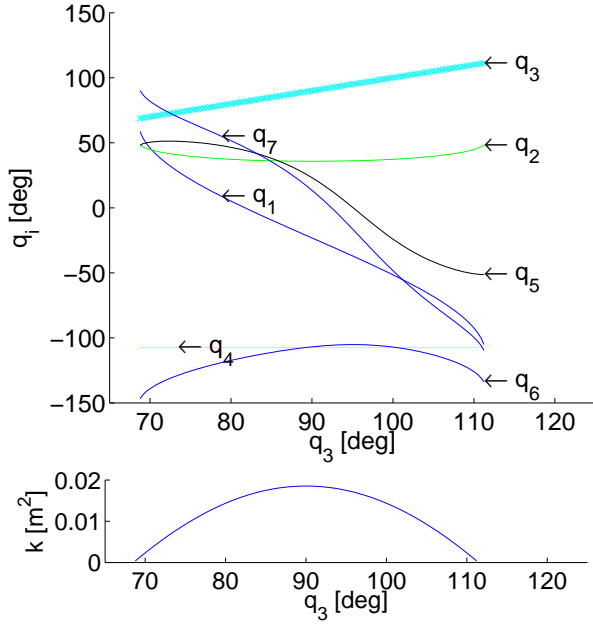
Fig. 3. Joint angles $\mathbf{q}_{\text{RightArm}}$ of the right Justin arm during a nullspace motion with torso joints $\mathbf{q}_{\text{Torso}}$ held constant as a function of q_3, $q_3 > 0$ shown at the top. Values for the term k are shown at the bottom. The root $\sqrt{k}$ needs to be calculated for the inverse kinematics algorithm s_4 with $q_{\text{fix}} = q_4$.

2) *Update passive joints:* All passive joints (in case of the experimental system Justin, only q_4 is a passive joint) are updated according to the initial joints $\mathbf{q}_{\text{init}}$ and the new optimization parameters $\mathbf{q}_{\text{opt}}$.

3) *Update forward kinematics:* Starting from the robot base, the kinematic chain is updated using the forward kinematics. This way, the bases ${}^{br}\mathbf{T}$ and ${}^{bl}\mathbf{T}$ are determined

4) *Update closed form solutions:* The closed form solutions are calculated. This way all sought joint are found.

5) *Update variable joint limits:* In case of the Justin systems, the joint limits of q_2 are a function of q_1 and need to be updated in every step.

6) *Evaluate optimization criteria:* The optimization critera are evaluated, and the Levenberg-Marquardt algorithm continues with step 1 in the next iteration step.

The optimization is stopped if either all criteria are minimized to zero, or the maximum number of iterations is reached, or the decrease in the values of the optimization criteria functions is below a certain threshold from one step to another. The next Subsection describes the included optimization criteria.

D. Optimization criteria

The following optimization criteria are considered:

- c_1 *Preference position of the arms' elbows:* the distance between each elbow and a fixed point in space is minimized.
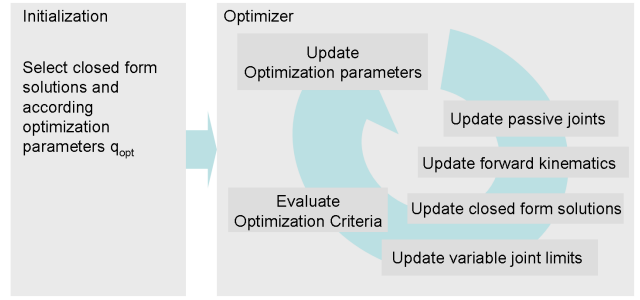

Fig. 4. Inverse kinematics algorithm for combined closed form solution with nullspace optimization.

- c_2 *Variable joint limits (position dependant):* the joint limits are updated in every cycle. In case a minimum distance of 15° from the joint limits is reached, a penalty function is assigned.
- c_3 *Low joint speeds:* The joint speeds are minimized.
- c_4 *Low joint speeds, scaled with TCP motion:* the joint speeds are variably minimized. If the TCP is moving fast, also fast Nullspace motions are allowed, whereas the robot does not move in Nullspace if it does not move in Cartesian space.
- c_5 *Singularity avoidance:* All arm singularities can be calculated in closed form according to [5]. Similar to the joint limits, a penalty function is assigned if a minimum distance of 15° from the singularities is reached.

III. EXPERIMENT

The inverse kinematics algorithms are currently tested for the Justin system. First results are shown in the following. The Levenberg-Marquardt algorithm of [7] is chosen for the implementation. The next Subsection presents experiments on reachability and calculation times, then realtime experiments are depicted.

A. Reachability and Calculation time

Reachibility is an important issue e.g. if a goal joint configuration $\mathbf{q}_d$ for path planning needs to be generated from a given goal in task space. In this case, the initial joint angles correspond to a configuration that is usually far away from the goal configuration, thus a good initial guess for the inverse kinematics algorithm is not available. Many algorithms do not converge in this case and hence fail. Goal of this work is to propose an algorithm that practically always finds a solution.

Figure 5 and Fig. 6 show the performance of an implementation of the *Closed solution subchains* algorithm as described in Section II with varying maximum number of iterations n. As optimization criteria, c_1, c_2, c_4 and c_5 are considered. Analytic derivatives can not be calculated in this case, since calculation of the nullspace projection is too time consuming and would be necessary due to the closed form solutions. To calculate the error rate and the calculation speed, a set of 5000 reachable tool tip frames was generated

and the solution of the inverse kinematics was calculated. To evaluate the influence of the deviation of the initial solution from the sought solution, the *feasible angles deviation ε* of the initial angles $\mathbf{q}_{\mathrm{init}}$ is plotted on the horizontal axis, defined as follows:

$$\mathbf{q}_{\mathrm{init}} = \mathbf{q}_{\mathrm{valid}} + \mathrm{rand} \cdot \varepsilon, \qquad (4)$$

with $\mathbf{q}_{\mathrm{valid}}$ a joint angles vector leading to the sought tool tip frame, and $\mathrm{rand} \in [-1, ..., 1]$ a random number. An error is stated if either the calculated joint angles do not correspond to the sought TCPs or if joint limits are violated.
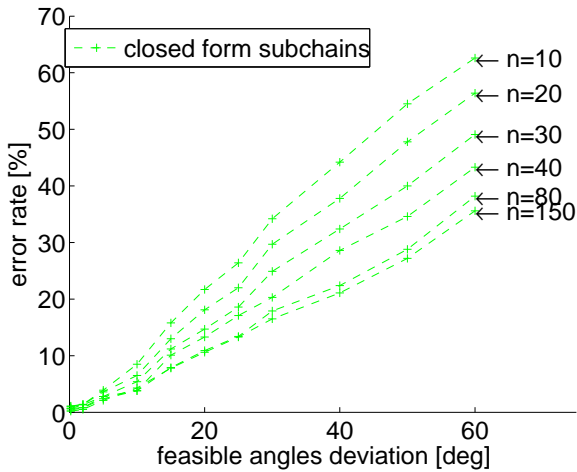

Fig. 5. Influence of the *feasible angles deviation* and the maximum number of iterations on the error rate for the *closed form subchains* algorithm.

From Fig. 5, it can be seen that the error rate is decreasing with increasing maximum number of iterations n, however a value below 30% can not be reached even with very high values for n. For *feasible angles deviations* below $10°$, a maximum number of iterations of $n = 10$ is sufficient. The calculation time as shown in Fig. 6 is basically proportional to n.
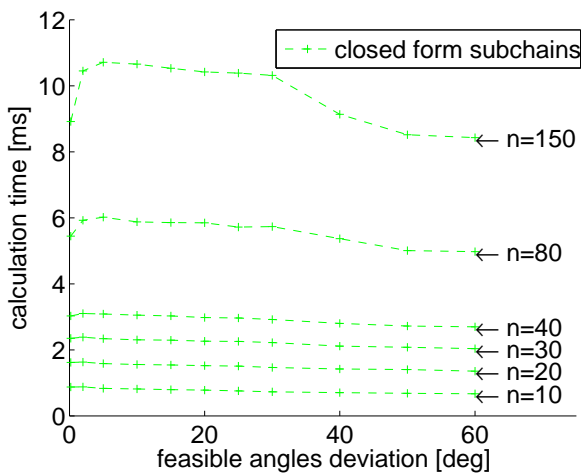

Fig. 6. Calculation time for the *closed form subchains* algorithm.

In order to achieve better reachability, the *Closed form subchains iterative* algorithm is implemented, similar to the *Closed solution subchains* algorithm, but with $n = 150$ and additional 100 iterations over the initial values for the optimization parameters.

Furthermore, for comparison purpose, a *Nonlinear optimization* algorithm is considered that implements a pure nonlinear optimization using Levenberg-Marquardt. For this algorithm, the joints $\mathbf{q} = [\mathbf{q}_{\mathrm{Torso}}\ q_{4,}\ \mathbf{q}_{\mathrm{RightArm}}\ \mathbf{q}_{\mathrm{LeftArm}}] \in R^{18}$ are subject to optimization, while the passive joint condition 1 is included as optimization criteria. Also the condition to reach the TCPs is formulated as an optimization criterion, and thus it will be influenced by the other optimization criteria[3]. Furthermore, criteria c_2, c_3, and c_5 are considered. Analytic derivatives are given for the optimization criteria to speed up the Levenberg-Marquardt optimization.

Other standard algorithms for inverse kinematics calculation based e.g. on the Jacobian inverse were shown to have low performance with respect to reachability [4].

Results of the algorithm performance are given in Fig. 7 and Fig. 8. The *Nonlinear optimization* algorithm has a higher error rate than the *Closed form subchains* algorithm, while its calculation time is faster. The *Closed form subchains iterative* algorithm has a very low error rate regardless of the *feasible angles deviation* and thus achieves the goal of independance for initial joint angles. However the calculation time is very high as can be seen in Fig. 8.
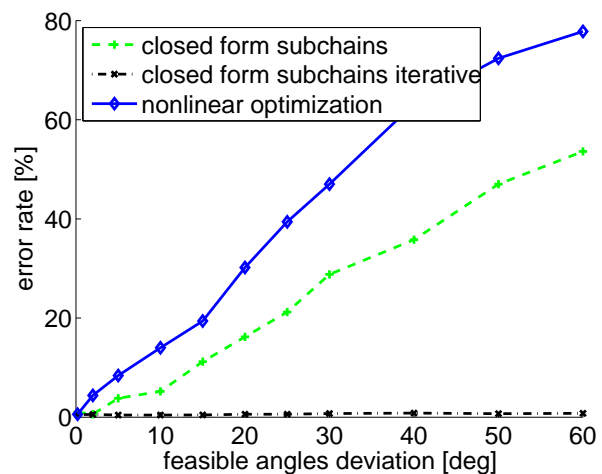

Fig. 7. Error rates as a function of the *feasible angles deviation* .

With variation of n, the calculation time and error rate of the *closed form subchains iterative* algorithm can be tuned. As shown in Fig. 9 and Fig. 10, reasonable values for path planning and reachability checks are $n \in [60, ..., 150]$.

---

[3]Note that this algorithm requires that all optimization criteria may be minimized to zero, otherwise the TCP position is never reached exactly by the algorithm. This is a drawback of the used optimization method. However it is considerably faster than other standard methods like e.g. SQP methods [11] which would allow for constrained optimization and was therefore chosen in this paper.
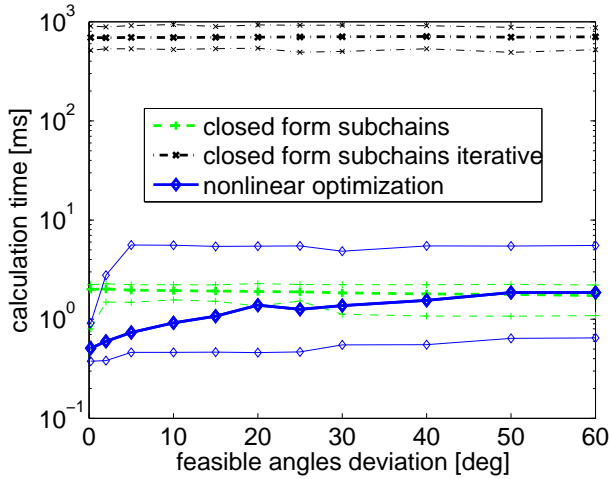
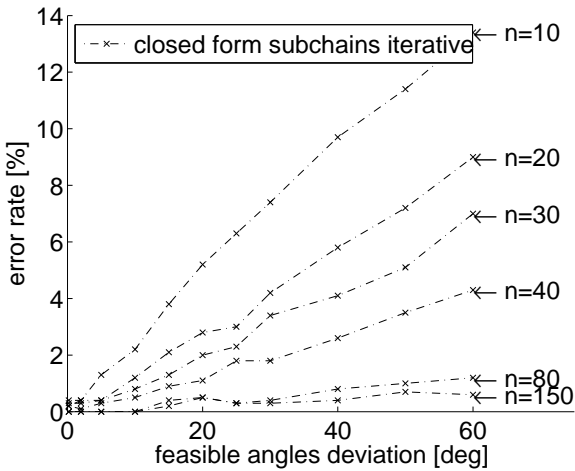Fig. 8. Calculation time for different inverse kinematics algorithms.

Fig. 9. Influence of the maximum number of iterations on the error rate for the *closed form subchains iterative* algorithm.
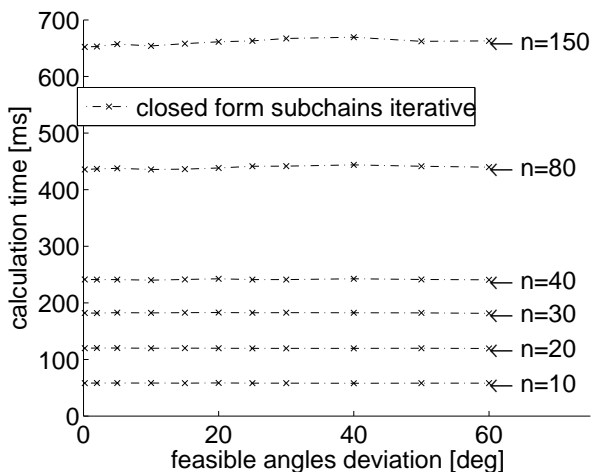

Fig. 10. Calculation time for the *closed form subchains iterative* algorithm with variation of the maximum number of iterations.

B. Real time performance

Real time performance is shown in the attached video. The inverse kinematics algorithm is implemented in C++ and uses an s-function as interface to simulink. It runs in a multirate model with a sampling rate of $100\,\mathrm{Hz}$, while the whole model runs at $1\,\mathrm{kHz}$. In the video, the *mobile Justin* is shown. The task is to hold a tray in a fixed position while the platform is moving along a circle.

IV. CONCLUSION

This paper presents two new inverse kinematics algorithms that combine closed form solutions with nonlinear optimization for highly redundant robotic systems. The right choice of the closed form solution such that algorithmic singularities are avoided is detailed, and the complete algorithm is depicted. From this, two new algorithms are presented in the experiment: the *closed form subchains* algorithm is suitable for realtime use with calculation times of 1 ms, while the *closed form subchains iterative* algorithm achieves a very high reachability above 99% throughout the workspace with calculation times of 1 s. Realtime capabilities are presented for the *closed form subchains* algorithm as it is applied to the *mobile Justin* shown in the attached video.

The presented methods are also valid for other serial kinematic chains. Note however that in each branch of a kinematic tree structure only one closed form solution may be present. The algorithms will be also tested with DLR's minimally invasive robotic surgery system [2]: here, three robots with altogether 25 active joints are considered. Two robots are manipulating forceps, while the third robot guides an endoscope. Currently the inverse kinematics are solved for each robot independantly. A common inverse kinematics to all robots could allow for better performance of global optimization criteria such as collision avoidance between the robots.

REFERENCES

[1] Christoph Borst, Christian Ott, Thomas Wimbock, Bernhard Brunner, Franziska Zacharias, Berthold Bäuml, Ulrich Hillenbrand, Sami Haddadin, Alin Albu-Schäffer, and Gerd Hirzinger. A Humanoid Upper Body System for Two-Handed Manipulation. In *ICRA*, pages 2766–2767. IEEE, 2007.

[2] Ulrich Hagn, Mathias Nickl, Stefan Jörg, Andreas Tobergte, Bernhard Kübler, Georg Passig, Martin Gröger, Ulrich Seibold, Rainer Konietschke, Tobias Ortmaier, and Gerd Hirzinger. DLR MiroSurge project – An approach towards versatility in surgical robotics. In Volkmar Falk and Oliver Burgert, editors, *Proceedings of the Annual Convention of Deutsche Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, Leipzig, Germany, September 2008.

[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. *IEEE J RA*, 12:566–580, August 1996.

[4] Rainer Konietschke, Silvia Frumento, Tobias Ortmaier, Ulrich Hagn, and Gerd Hirzinger. Kinematic Design Optimization of an Actuated Carrier for the DLR Multi-Arm Surgical System. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2006*, Beijing, China, 2006.

[5] Rainer Konietschke, Gerd Hirzinger, and Yuling Yan. All Singularities of the 9-DoF DLR Medical Robot Setup for Minimally Invasive Applications. In *Advances in Robot Kinematics*, Ljubljana, Slovenia, 2006.

[6] K. Levenberg. A Method for the Solution of Certain Problems in Least Squares. 2:164–168, 1944.

[7] Manolis Lourakis. levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms in C/C++. [web page] http://www.ics.forth.gr/ lourakis/levmar/, July 2004. [Accessed on 31 Jan. 2006.].

[8] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[9] F.B. Ouezdou and RŽegnier Mavroidis. Kinematic Synthesis of Manipulators Using a Distributed Optimization Method. 121, 1999.

[10] D. L. Pieper. *The Kinematics of Manipulators Under Computer Control*. PhD thesis, Stanford University, Department of Mechanical Engineering, 1968.

[11] Klaus Schittowski. NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems. *Annals of Operations Research*, 5:485–500, 1985.