

# Joint Trajectory Optimization Using All Solutions of Inverse Kinematics of General 6-R Robots

U. Kuenzer and M.L. Husty

**Abstract** Based on a paper [1] that introduced an algorithm which allows to transform all solutions of the inverse kinematics of a general 6-R robot into continuous joint trajectories we present in this paper methods to use the continuous joint trajectories for path optimization. For this purpose the motion is discretized and at each instant the inverse kinematic is computed using a fast algorithm developed in [2, 3]. In the set of resulting joint angles continuous paths are determined and the resulting sets of points are interpolated with quintic splines. Different possibilities for trajectory optimization are discussed.

**Keywords** Inverse kinematics · General 6R-robot · Trajectory planning · Trajectory optimization · Student paper

## 1 Introduction

The inverse kinematics (IK) of general serial 6-R robots was in the 1980s of the 20th century considered to be one of the most challenging problems in robot kinematics. The first solution to this problem was given by Lee and Liang [4]. Many papers followed, the most popular being [5], a solution that is mostly cited in textbooks when the inverse problem is discussed. An overview of the existing literature can be found in the thesis [2, 3], where a new approach to the inverse problem was presented. This solution algorithm uses the Study parametrization of the Euclidean displacement group  $SE(3)$  and needs much less equations than Raghavan's algorithm (eight compared to fourteen). Furthermore, the starting

---

U. Kuenzer (✉) · M.L. Husty  
Institute for Basic Sciences in Engineering, AB Geometry and CAD, University Innsbruck,  
Innsbruck, Austria  
e-mail: ulrich.kuenzer@student.uibk.ac.at

M.L. Husty  
e-mail: manfred.husty@uibk.ac.at

equations are so simple that they can be formulated completely general, i.e. without specifying the Denavit-Hartenberg parameters, making the algorithm applicable to any thinkable robot architecture without reformulation. Study parametrization is an algebraic parametrization of  $SE(3)$ , using eight parameters, that can be interpreted as homogeneous coordinates in a seven dimensional projective space  $P^7$ . To meet the dimensions of the Euclidean displacement group the coordinates have to fulfill a quadratic equation, corresponding to a quadric in  $P^7$ , the so called Study quadric. Detailed information on this parametrization and its use in kinematics can be found e.g. in [6].

Within a software developing project called Kinsoft this algorithm was implemented in C# and this package allows to compute all inverse kinematics solutions fast along a given motion trajectory at as many instances of the trajectory as specified. With motion trajectory (or equivalently motion curve) a curve on the Study quadric is meant. Note that a curve on the Study quadric corresponds to a one parameter motion in the Euclidean space. Mathematically the curve is represented by an eight dimensional vector function encoding position and orientation of the end effector (EE). At any instant the IK algorithm returns all solutions of the inverse kinematics. Having all solutions of the inverse kinematics at many instances the following problems arise:

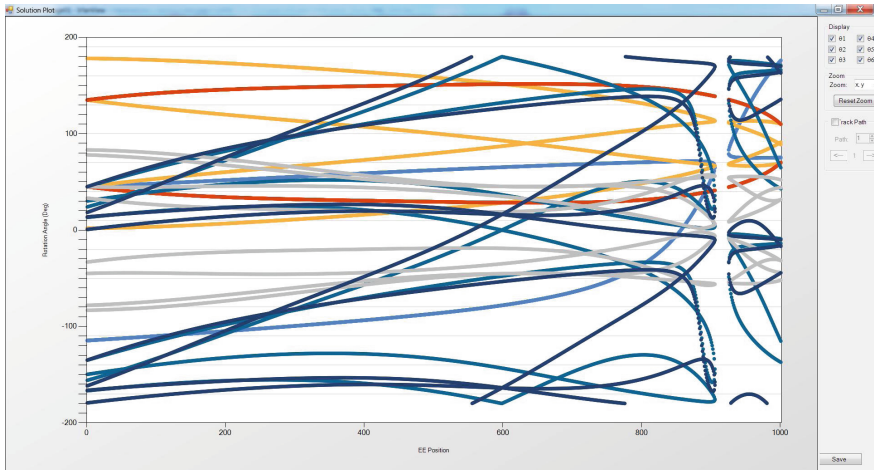
1. How can the solutions for different joint angles be separated such that a continuous path for each joint angle and each of its different solutions can be generated?
2. How can the discrete solution set be transformed into curves that have some desired properties, like stable behaviour at the end points for at least three orders of differentiability?
3. How can the data be used to distinguish the different solutions in order to find out which joint trajectory path is optimal according to some optimizing criteria.

The first two items are discussed in [1]. The results of this paper are recalled briefly in Sect. 2. In Sect. 3 the optimization strategies are presented. Section 4 shows some examples where one can clearly see that the presented optimization algorithm allows to choose between different solutions of the inverse kinematics to optimize the desired behaviour of the manipulator.

## 2 Separating the Joint Paths

In this section results from [1] are recalled to make the optimization strategies of Sect. 3 understandable.

When the Denavit-Hartenberg parameters of a 6-R robot are specified and a desired trajectory of the EE in position and orientation is given then the Kinsoft program [7] will return a text file that graphically processed yields an output as shown in Fig. 1. On the axis of abscissae of this plot one can see that 1,000 points on the EE trajectory have been used to compute the inverse kinematics. On the axis



**Fig. 1** Joint angle values resulting from inverse kinematics along an EE trajectory

of ordinates the corresponding joint angles are displayed in different colours in the range of  $-180^\circ \dots 180^\circ$ . Although it seems that there are continuous curves, the program can only return discrete sets of joint angles at every instant.

*Remark* The trajectory in the example of Fig. 1 had been chosen such that the manipulator hits its boundary which can be seen clearly because of the gap in all joint trajectories. The path generation algorithm follows the following steps:

1. Data import from the Kinsoft output and data storing in special arrays.
2. Separating the paths using the distance function.

**Definition 1** Let  $a, b \in [-180^\circ, 180]^\circ$  be two vectors. Then the difference between the two sets of angles is defined as

$$\|a - b\|_p^W := \left\| \begin{pmatrix} \min(|a_1 - b_1|, 360 - (|a_1 - b_1|)) \\ \min(|a_2 - b_2|, 360 - (|a_2 - b_2|)) \\ \min(|a_3 - b_3|, 360 - (|a_3 - b_3|)) \\ \min(|a_4 - b_4|, 360 - (|a_4 - b_4|)) \\ \min(|a_5 - b_5|, 360 - (|a_5 - b_5|)) \\ \min(|a_6 - b_6|, 360 - (|a_6 - b_6|)) \end{pmatrix} \right\|_p$$

where on the right side any  $p$ -Norm from  $\mathbb{R}^6$  can be used.

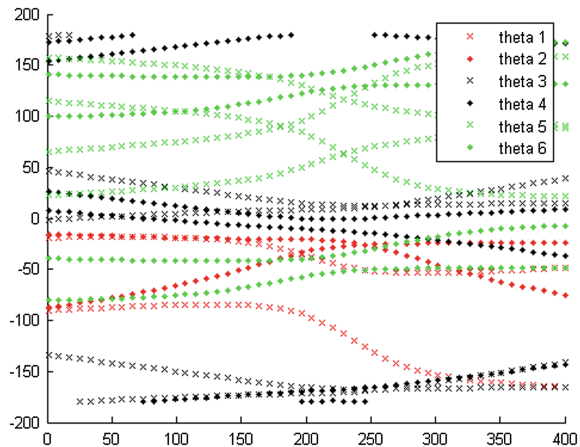
An upper bound of solutions is given by the number of solutions  $n_1$  in the first point of the motion curve. In order to find a continuation of the first solution of the first point  $\theta_1^1$ , the differences  $\|\theta_1^1 - \theta_i^2\|_2^W, i = 1, \dots, n_2$  are computed. If the minimum of these differences is smaller than an error bound  $\epsilon$ , then the solution is added to the path.

The error bound must depend on the number of points on the motion trajectory  $m$ , because with the number of points in which the inverse kinematics is computed the distance between the data points diminishes. Furthermore the error bound must be adapted to the used norm. In the algorithm different values of the error bound are tested to find out in which range the error bound has to be chosen such that all paths are found.

3. After path separation the joint paths are interpolated using quintic splines. Boundary conditions are given by the first two derivatives on the interval boundaries. The system of equations used is classical and can be found in any textbook on spline interpolation (see. e.g. [8] and in the thesis [9]). The derivatives at the boundary of the interval are not given explicitly, they have to be estimated. Discrete estimation using the first two points of the data did not yield satisfying results, the resulting splines showed unwanted oscillations in the derivatives of the curves near the boundaries. To obtain better results smoothness of the curves was used. The data which have to be interpolated result from the solution of a polynomial system of equations. If the solutions are not complex then they have to describe a smooth  $C^\infty$  curve in  $\mathbb{R}^6$ . Using this fact the derivatives in the boundary point were approximated. To do this the first six and the last six points of the data were used to construct a quintic polynomial. The derivatives of those two polynomials were used as boundary conditions for the interpolation of the path splines.

Figure 2 shows the result of the three steps explained above for a classical example from the literature (example nr.7 from Wampler-Morgan [10], abbreviated CW7 in the following). The Denavit-Hartenberg parameters of the manipulator are given in Table 1.

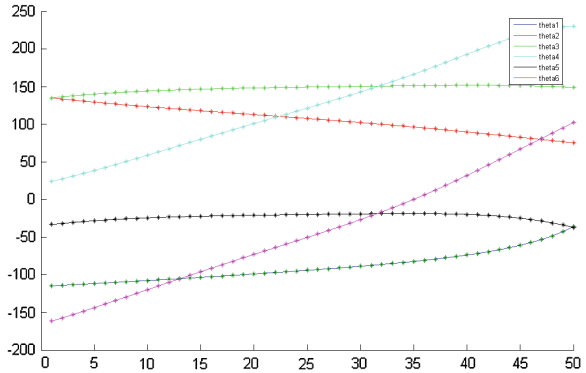
**Fig. 2** Joint angles of the four continuous paths in joint space of 6-R robot performing a continuous EE motion



**Table 1** DH-parameter of example CW7

$a$	$\frac{3}{10}$	1	0	$\frac{3}{2}$	0	0
$d$	0	0	$\frac{2}{10}$	0	0	0
$\alpha$	$\frac{\pi}{2}$	$\frac{\pi}{18}$	$\frac{\pi}{2}$	$\frac{\pi}{18}$	$\frac{\pi}{2}$	$\frac{\pi}{18}$

**Fig. 3** Interpolated joint angle paths of one solution of a 6-R robot performing a continuous EE motion



The motion which has been performed is given by the Euclidean motion matrix  $\mathbf{B}_C$ :

$$\mathbf{B}_C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1.140175 - t & -0.760117 & -0.641689 & 0.102262 \\ 0.133333 - t & 0.133333 & 0.991071 & 0 \\ \frac{t}{2} & -0.635959 & 0.766965 & 0.085558 \end{pmatrix}. \quad (1)$$

Note that European notation is used in  $\mathbf{B}_C$ , writing the translation part of the motion in the first column and the homogenizing coordinate at first place. In Fig. 2 the upper boundary and the lower boundary have to be identified, and some angles (e.g.  $\theta_5$ ) seem to have four paths and others only two (e.g.  $\theta_1$ ). This is due to the fact that in these angles two solutions coincide. Figure 3 shows a typical result of this procedure where one of the possible continuous solutions of the joint trajectories of a robot motion has been plotted after performing the quintic spline interpolation.

### 3 Joint Trajectory Optimization

After the first three steps of the algorithm quintic splines of the joint trajectories are given. Because of the fact that the inverse kinematics yields more than one solution more than one continuous solution in the joint trajectories will exist. Therefore one has the possibility to choose among the solutions according to some optimization criteria. As joint angles are computed in the inverse kinematics, the chosen criteria

should be linked to the joint angles along a given motion trajectory. In a first choice (type 1 optimization) we will minimize the overall change in the joint angles and in a second choice (type 2 optimization) we will ask for the solution which has a minimum change in joint angular velocities for all joints.

To perform this task a function  $f : [t_1, t_2] \mapsto \mathbb{R}$  is considered, which determines the joint angles and which is at least twice continuously differentiable. Then  $f'(t) = \frac{df}{dt}$  determines the change of joint angles or the joint velocity. The change of the velocity  $f''(t) = \frac{d^2f}{dt^2}$  yields the joint acceleration. The overall joint angle changes are given by the integral  $\int_{t_1}^{t_2} f'(t)dt$ . This integral has the disadvantage that positive and negative velocities cancel therefore we have taken the absolute values of the functions  $\int_{t_1}^{t_2} |f'(t)|dt$  and  $\int_{t_1}^{t_2} |f''(t)|dt$ . Integration of the absolute value of the velocity function yields information about the total length of the joint trajectory. The shorter the trajectory the less movement in the joint will be. Analogous consideration gives information that a smaller value for the integral  $\int_{t_1}^{t_2} |f''(t)|dt$  yields a smaller change of joint velocities in the interval  $[t_1, t_2]$ .

The interpolation curves are polynomials of degree five. Therefore it is possible to determine the integral exactly with help of quadrature rules. Because the curves are continuous and are piecewise polynomial, integration can be done separately for each piece of the curve. For computation of the integrals Gauss quadrature rule was used

$$\mathcal{I}_n := \sum_{k=1}^n \sigma_k f(\lambda_k) \text{ for } f \in [t_1, t_2], \tag{2}$$

with the nodes  $\lambda_1, \dots, \lambda_n \in \mathbb{R}$  and the weights  $\sigma_1, \dots, \sigma_n$ .

## 4 Examples

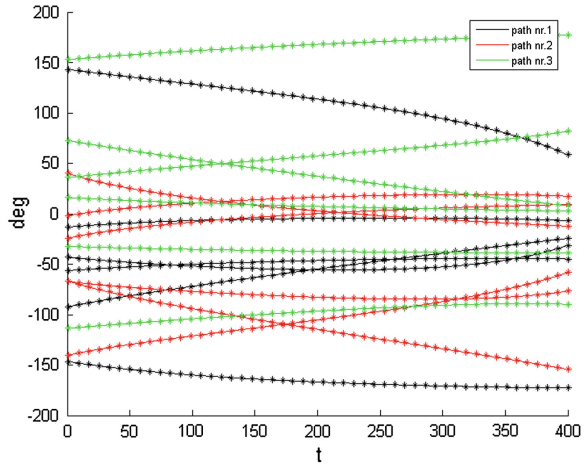
### 4.1 Example 1

In a first example the Denavit-Hartenberg parameters of the CW7 example in Table 1 were slightly modified as in (Table 2). The motion performed is the motion of Eq. (1) in the interval  $[-\frac{1}{2}, \frac{1}{2}]$ .

**Table 2** DH-parameter of Example 1

$a$	$\frac{3}{10}$	1	0	$\frac{3}{2}$	0	0
$d$	0	0	$\frac{2}{10}$	0	0	0
$\alpha$	$\frac{\pi}{2}$	$\frac{(\pi + 0.5)}{18}$	$\frac{\pi}{2}$	$\frac{\pi}{18}$	$\frac{(\pi - 0.5)}{2}$	$\frac{\pi}{18}$

**Fig. 4** Interpolated joint trajectories of Example 1



**Table 3** Result of the optimization

Optimization type	Path Nr.1	Path Nr.2	path Nr.3
1. Type	39.9009	50.5817	30.0131
2. Type	0.2598	0.3153	0.0824

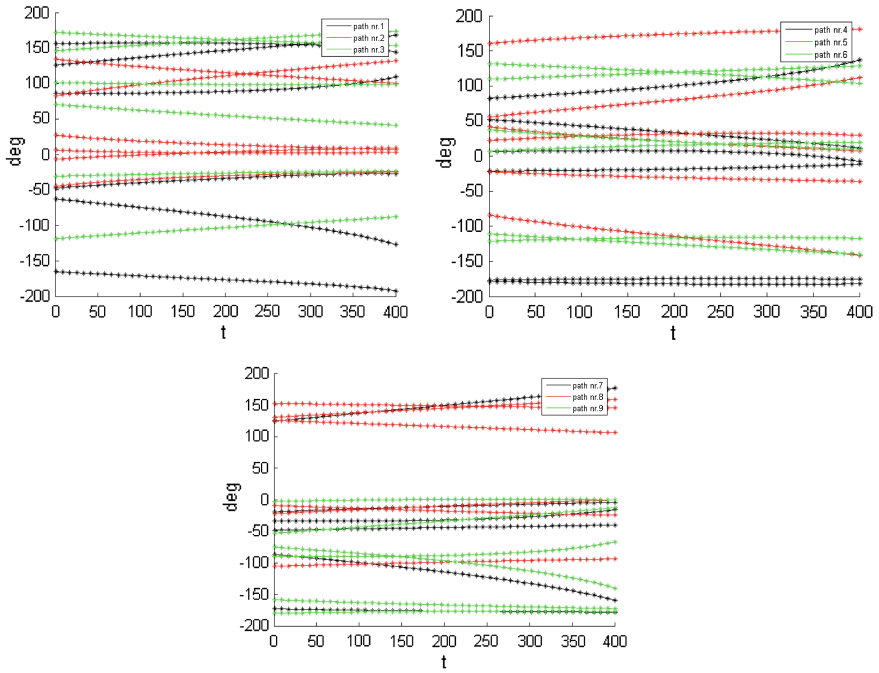
This example has three continuous paths (Fig. 4). The results of the optimization are shown in Table 3. The average of the six joint trajectory lengths is the smallest in the green path 3. Therefore this path is optimal when smallest overall change of joint parameters is desired. It can also be seen that the same joint trajectories are optimum when smallest overall change of joint velocities is desired. Once this result is obtained it is of course enough to put the manipulator in the starting values of path 3 and then it will stay on this path, because there is no singularity on this path. A boundary singularity would be seen in such a plot when a path ends, i.e. when two solutions meet in a point having a vertical tangent. This behaviour can be seen in Fig. 1 where the motion trajectory was chosen such that the manipulator hits its boundary and therefore no continuous path is possible. There is a gap in all joint trajectories.

### 4.2 Example 2

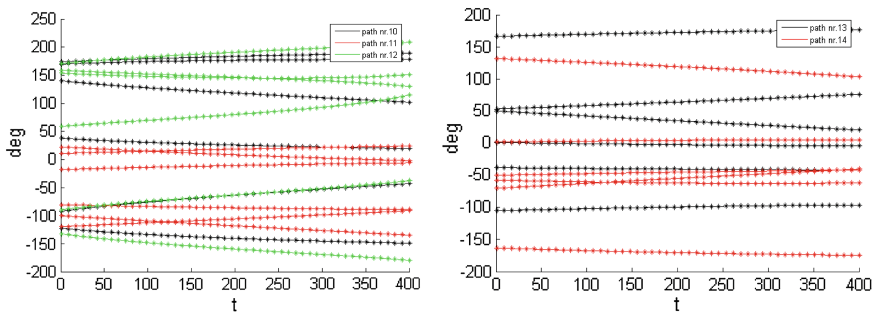
In this example the DH-parameters of CW7 are changed according to Table 4. The motion performed is the same as in Eq. (1) in the interval  $t \in [-\frac{1}{4}, \frac{1}{4}]$ . This example is especially interesting, because 14 continuous joint paths exist. The given task can be performed with 14 different configurations of the manipulator. It should be obvious that these 14 paths are not equivalent The 14 continuous joint trajectories are shown in Figs. 5 and 6.

**Table 4** DH-parameter of Example 2

$a$	$\frac{3}{10}$	$\frac{11}{10}$	0	$\frac{3}{2}$	0	0
$d$	0	0	$\frac{25}{100}$	0	0	0
$\alpha$	$\frac{\pi}{2}$	$\frac{(\pi + 0.2)}{18}$	$\frac{(\pi + 0.1)}{2}$	$\frac{\pi}{18}$	$\frac{(\pi - 0.3)}{2}$	$\frac{\pi}{18}$



**Fig. 5** Interpolated joint trajectories 1–9 of Example 2



**Fig. 6** Interpolated joint trajectories 10–14 of Example 2



**Table 5** Results of the optimization Example 2

Optimization type	Path Nr.1	Path Nr.2	Path Nr.3	Path Nr.4	Path Nr.5	Path Nr.6	Path Nr.7
1. Type	31.7504	24.1101	19.4176	21.5736	32.5666	20.5959	28.8555
2. Type	0.1897	0.0846	0.0213	0.1273	0.1165	0.0542	0.0954
	Path Nr.8	Path Nr.9	Path Nr.10	Path Nr.11	Path Nr.12	Path Nr.13	Path Nr.14
1. Type	17.2200	25.4676	26.2039	20.1783	39.7655	13.5023	14.7125
2. Type	0.0157	0.1321	0.0984	0.0251	0.2140	0.0163	0.0280

In the result of the optimization one can see big differences in the different trajectories (Table 5). Concerning the average joint motion change trajectory 13 is optimal whereas the path 12 is the worst path. One can see this also in the plot of Fig. 6 on the right side. Trajectories of path 13 (black) have almost horizontal curves, whereas the trajectories of path 12 have much more vertical deviation. Concerning the overall velocity change averaged over all joints (type 2) one can see that path 8 does slightly better than path 13.

*Remark* In a real robot joint limits would have to be taken into account. This has not been done in this example because the joint limits of the Wampler example are not specified. But it would be no additional effort to plot these limits and make a decision if the given motion can be performed without hitting a joint limit.

Summarizing the results of both optimization algorithms one can state clearly that it matters which starting configuration is chosen from the different possibilities offered by the different solutions of the inverse kinematics. The better choice of a starting can reduce the cycle time of a desired task considerably.

## 5 Conclusion

The inverse kinematics of a general 6-R manipulator yields up to 16 solutions for the joint angles when the end effector pose is given. Using the fast algorithm developed in [2, 3] the inverse kinematics can be computed along a given end effector motion efficiently. The algorithm presented in this paper detects in the set of joint angles continuous paths and the joint trajectories of these paths are interpolated with quintic splines. Having polynomial curves for joint trajectories, velocities and acceleration a optimization procedures were discussed to decide which of the possible solutions is optimal according to a given optimization criterion like e.g. minimum overall change of joint motion or minimum average overall change of joint velocities. This work can help to decide in which starting pose the manipulator should be brought to perform a given task in a shorter time or with overall smaller change of joint velocities.

## 6 Acknowledgments

The research presented in this paper was supported by the Translational research project KineControl of the Standortagentur Tirol, Austria. M. Husty acknowledges the support of FWF project P 23831-N13, Algebraic Methods in Collision Detection and Path Planning.

## References

1. Kuenzer U, Husty M (2014) Joint trajectory generation using all solutions of inverse kinematics of general 6-R robots. In: Proceedings of ARW (Austrian Robotics Workshop), 2014. accepted for publication and presentation
2. Pffurner M (2006) Analysis of spatial serial manipulators using kinematic mapping. PhD thesis, University of Innsbruck, October 2006
3. Husty M, Pffurner M, Schröcker H-P (2007) A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mech Mach Theory* 42(1):66–81
4. Lee H, Liang C (1988) Displacement analysis of the general 7-link 7R mechanism. *Mech Mach Theory* 23(3):219–226
5. Raghavan M, Roth B (1990) Inverse kinematics of the general 6R manipulator and related linkages. *Trans ASME J Mech Des* 115:228–235
6. Husty M, Schröcker H (2012) 21st century kinematics, ch. Kinematics Algebraic Geometry, pp 85–107. Springer, Berlin
7. Pffurner M, Husty ML (2010) Implementation of a new and efficient algorithm for the inverse kinematics of serial 6R chains. In: Pisla D, Husty ML, Corves B, Ceccarelli M (eds.) *New trends in mechanism science*, pp 91–98, Springer, New York
8. Späth H (1973) *Spline-algorithmen zur Konstruktion glatter Kurven und Flächen*. Oldenburg, München
9. Kuenzer U (2014) Bestimmung von glatten und optimierten Kurven im Raum der Gelenksvariablen eines allgemeinen 6-R Manipulators aus den Daten der inversen Kinematik. Master's thesis, Universität Innsbruck, 2014
10. Wampler C, Morgan A (1991) Solving the 6R inverse position problem using a generic-case solution methodology. *Mech Mach Theory* 26(1):91–106