Documentation for the program to find the inverse kinematics solutions for a general 6-R serial manipulator

Ambuj Shahi and Sandipan Bandyopadhyay
Department of Engineering Design
Indian Institute of Technology Madras
Chennai-600 036

1 Introduction

This article documents the use of the function `InverseKinematicsGeneral6R`, to compute the inverse kinematics solutions of the general 6-R serial manipulator. The structure of the directory in which the aforementioned function is located is as follows:

```
Parent directory (IK_Gen_6-R)
├─ Coeff_x0
├─ Coeff_x1
├─ Coeff_x2
├─ Doxygen_documentation
└─ Examples
   ├─ example1
   ├─ example2
   ├─ example3_Jaco
   ├─ IK
   ├─ Residue
   └─ Theta
```

The function can be located in the `IK` folder and its usage has been demonstrated through three examples, situated in the folders `example1`, `example2`, and `example3_Jaco`. In the folder `example1`, the numerical example taken by Raghavan and Roth [1] has been implemented, in folder `example2`, the numerical example taken by Liu and Zhu [2] has been implemented, and in folder `example3_Jaco`, the inverse kinematics algorithm has been implemented on the general 6-R robot developed by `Kinova technology`, **JACO**®[3].

1.1 Input files and other dependencies

The inputs to the function `InverseKinematicsGeneral6R` are the D-H parameters of the robot and its End-effector pose, which have been denoted by `dhmat` and `eemat` respectively, in the function. The inputs `dhmat` and `eemat` are being taken from separate `CSV` files. For the input `dhmat`, the corresponding `CSV` file should have 3 rows and 6 columns, the 1^{st} row representing the D-H parameter a of the robot with the six column entries of that row representing the values of a_1 to a_6 for the robot. The values of a_1 to a_6 should be in meters. The 2^{nd} row represents the D-H parameter α of the robot with the six column entries of that row representing the values of α_1 to α_6 for the robot. The values of α_1 to α_6 should be in radians. The 3^{rd} row represents the D-H parameter d of the robot with the six column entries of that row representing the values of d_1 to d_6 for the robot. The values of d_1 to d_6 should be in meters. In this algorithm, the classic D-H parameters [4] of the robot has been considered. So, if the robot has been modelled using modified D-H parameters, then that must be converted into classic D-H parameters, which can then be used as input to the `CSV` file in the way described above to create `dhmat`. In the folder `example1`, one such `CSV` file can be found named, `dhmat_Raghavan_Roth.csv`.

For the other input `eemat` to the function `InverseKinematicsGeneral6R`, the corresponding `CSV` file should have 4 rows and 3 columns, the 1^{st} row of which represents the 1^{st} column of the rotation matrix, representing the orientation of the end-effector of the robot w.r.t the base frame. The 2^{nd} row of this `CSV` file represents the 2^{nd} column of the aforementioned rotation matrix, the 3^{rd} row of the `CSV` file represents the 3^{rd} column of the aforementioned rotation matrix and the 4^{th} row of the `CSV` file represents the position vector (P_x, P_y, P_z) of the end-effector w.r.t. base frame, and values of the entries of the position vector should also be in meters. In the folder `example1`, one such `CSV` file can be found named, `eemat_Raghavan_Roth.csv`.

Following the papers by Raghavan and Roth and Manocha and Canny [1, 5] on Inverse kinematics of general 6-R robots, one can arrive at the 12×12 matrix, whose entries are quadratic polynomial in x_3, lets call it $\mathbf{\Sigma''}$. When the problem of root finding of the determinant of the matrix $\mathbf{\Sigma''}$ is reduced to an eigenvalue problem, the matrix $\mathbf{\Sigma''}$ can further be decomposed into $\mathbf{A}x_3^2 + \mathbf{B}x_3 + \mathbf{C}$, where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are 12×12 matrices and $x_3 = \tan\left(\dfrac{\theta_3}{2}\right)$. In the function `InverseKinematicsGeneral6R`, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ is represented by `cmatx2`, `cmatx1`, and `cmatx0` respectively, each of the entries of which are functions of the D-H parameters and the end-effector pose. The 12 x 12 coefficient matrices `cmatx2`, `cmatx1`, and `cmatx0` in the function `InverseKinematicsGeneral6R` has been formed by calling functions from the folders `Coeff_x0`, `Coeff_x1`, and `Coeff_x2` respectively. Each of the files in these folders has been formed out of the expressions generated in CAS Mathematica [6], after simplifying the entries of the corresponding coefficient matrices.

1.2 Example file

In the folder `example1` , there is a `C` file named `example1.c`, which demonstrates the usage of the function `InverseKinematicsGeneral6R`. Similar files are there in other example folders as well. This `example1.c` is compiled with the help of a `makefile`, which is there in that folder. For computing the inverse kinematics of any other general 6-R robot, the name of the CSV file containing the D-H parameters of that robot needs to be placed at the position of '`dhmat_Raghavan_Roth.csv`' in the program `example1.c`, and for any other end-effector pose of the robot, the name of the CSV file containing the end-effector pose needs to be placed at the position of '`eemat_Raghavan_Roth.csv`' in the program `example1.c`.

1.3 Results

The results from the numerical example-1, implemented in folder `example1` has been summarized below. The link parameters for the 6-R manipulator has been shown in the table below:

Table 1: D-H parameters of Raghavan and Roth's manipulator

$a_i(m)$	$d_i(m)$	α_i(deg)
0.8	0.9	20.0
1.2	3.7	31.0
0.33	1.0	45.0
1.8	0.5	81.0
0.6	2.1	12.0
2.2	0.63	100.0

The end-effector pose at the goal position was taken as:

$$\boldsymbol{A}_{hand} = \begin{bmatrix} 0.354937475307970 & 0.461639573991743 & -0.812962663562556 & 6.82151837150213 \\ 0.876709605247149 & 0.137616185817977 & 0.460914366741046 & 1.46146704002829 \\ 0.324653132880913 & -0.876327957516839 & -0.355878707125018 & 5.36950521368663 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Table 2: Sixteen solutions of IKP

Sol.	$\theta_1(^\circ)$	$\theta_2(^\circ)$	$\theta_3(^\circ)$	$\theta_4(^\circ)$	$\theta_5(^\circ)$	$\theta_6(^\circ)$	$e^*(10^{-4})$
1, 2	$-7.447 \pm 26.02i$	$10.73 \mp 77.84i$	$129.3 \pm 38.64i$	$-83.23 \pm 34.27i$	$17.74 \mp 84.77i$	$-16.51 \pm 66.89i$	0.061
3, 4	$46.78 \pm 60.09i$	$172.5 \mp 139.4i$	$-119.4 \pm 6.014i$	$-70.15 \pm 43.33i$	$91.52 \pm 149.4i$	$-29.14 \mp 125.0i$	0.247
5, 6	$53.64 \pm 6.493i$	$-96.18 \mp 20.01i$	$81.86 \pm 23.33i$	$18.04 \mp 15.90i$	$-32.86 \pm 48.10i$	$-2.663 \mp 37.06i$	0.044
7, 8	$30.07 \pm 61.98i$	$-43.49 \mp 226.3i$	$-147.5 \pm 113.7i$	$-39.25 \mp 9.875i$	$-68.98 \pm 236.0i$	$-2.945 \mp 145.6i$	1.92
9	$13.10 - 0.00i$	$50.99 - 0.00i$	$-72.04 + 0.00i$	$72.07 + 0.00i$	$-7.200 + 0.00i$	$-37.85 + 0.00i$	0.044
10	$14.00 - 0.00i$	$29.70 - 0.00i$	$-45.00 + 0.00i$	$71.00 + 0.00i$	$-63.00 + 0.00i$	$10.00 + 0.00i$	0.054
11,12	$135.8 \mp 32.33i$	$-166.9 \mp 20.43i$	$35.62 \pm 98.87i$	$20.72 \mp 54.02i$	$21.57 \pm 58.57i$	$-77.24 \mp 26.90i$	0.134
13, 14	$143.8 \pm 24.03i$	$-127.6 \mp 109.3i$	$-5.851 \pm 151.5i$	$150.4 \mp 79.83i$	$-117.2 \mp 166.3i$	$-30.59 \pm 133.8i$	0.984
15, 16	$166.7 \mp 130.3i$	$153.5 \mp 18.96i$	$-20.03 \pm 153.9i$	$36.40 \mp 156.2i$	$-164.3 \mp 89.82i$	$15.83 \pm 130.5i$	0.497

$^*e$ is the maximum among the absolute values of Res_1 to Res_6, which have been defined below.

The validity of the inverse kinematics solutions thus obtained is checked by calculating the six residues, denoted by Res_1 to Res_6, by back-substituting the solutions into Eq. (1-6), of Ref. [5], and the maximum values among these six residues, denoted by e has been shown in the table above. The `C` program file to calculate these residues can be found in the `Residue` folder, the location of which has been shown in the directory structure above.

1.4 Output file

The output file is also a `CSV` file, in which the solution tuple corresponding to all the real eigenvalues, x_3 [5] are stored. In the folder `example1`, one such `CSV` file can be located named, `example1_output.csv`. Similar files are there in the other example folders as well.

1.5 Failure of the algorithm

There can be three failure cases for this algorithm,

1. D-H parameter $a_1 = 0$, or
2. D-H parameter $\alpha_1 = 0$ or π, or

3. D-H parameter $\alpha_6 = 0$ or π,

For each of these cases, inverse kinematics algorithm needs to be developed separately, but there exists a workaround with the present algorithm through which we can find the inverse kinematics solutions even under these failure conditions. The idea is to choose a value of a_1 or α_1 or α_6 close to zero but not zero, when the failure condition corresponding to these D-H parameters is encountered. For example, for a robot with $a_1 = 0$ m, we can make this algorithm work by choosing $a_1 = 0.001$ m.

In the example with **JACO**®, this workaround has been implemented, as for **JACO**®, a_1=0 m. Taking a set of joint-angle tuple, within the joint-limits, $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ $= (-5^{\circ}, -40^{\circ}, 150^{\circ}, 2^{\circ}, -176^{\circ}, 106^{\circ})$, the final transformation matrix, representing the orientation and position of the end-effector of the robot with respect to the base frame, is determined. Using this matrix, the `CSV` file corresponding to the input `eemat` is formed.

Finally on applying this inverse kinematics algorithm on this robot with above calculated end-effector pose as input and with D-H parameter a_1=0.001 m, one of the joint-angle tuple that we obtain is:

$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = (-4.998^{\circ}, -40.029^{\circ}, 149.724^{\circ}, 2.337^{\circ}, -176.301^{\circ}, 105.838^{\circ})$.

The other inverse kinematics solutions can be found in the output file `example3_output.csv` in the folder `example3_Jaco`.

The specific inverse kinematics algorithm for each of these failure cases is still a work in progress.

1.6 Conclusion

In the `C` file `InverseKinematicsGeneral6R.c`, apart from the functions from `C standard library`, functions from GNU Scientific Library (GSL) [7] has also been used, for computing eigenvalues and eigenvectors of matrices. The configuration of the platform on which the inverse kinematics computations were performed are: Intel(R) Core(TM) i7-4790 processor, 3.6 GHz and 16 GB RAM. The computational time cost, taken as average over 2000 experiments, tested on the three aforementioned 6-R manipulators, are presented in the table below:

Table 3: Inverse Kinematics calculation time

Manipulators	Time (avg.) [ms]
RRM: Raghavan and Roth's Manipulator	1.12
LZM: Liu and Zhu's Manipulator	1.18
JACO®	1.21

References

[1] M. Raghavan and B. Roth, "Kinematic Analysis of the 6R Manipulator of General Geometry," *Proc. Fifth Int. Symposium on Robotics Research*, 1990.

[2] S. Liu and S. Zhu, "An Optimized Real Time Algorithm for the Inverse Kinematics of General 6R Robots," in *2007 IEEE International Conference on Control and Automation*, May 2007, pp. 2080–2084.

[3] JACO, "Kinova technology," 2013. [Online]. Available: http://www.kinovarobotics.com/innovation-robotics/products/robot-arms

[4] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control.* Wiley, 2005. [Online]. Available: https://books.google.co.in/books?id=wGapQAAACAAJ

[5] D. Manocha and J. F. Canny, "Real Time Inverse Kinematics for general 6R Manipulators," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation.* IEEE Comput. Soc. Press, 1992, pp. 383–389. [Online]. Available: http://ieeexplore.ieee.org/document/220309/

[6] W. R. Inc., "Mathematica, Version 10.4.1 and 11.0.1," champaign, IL, 2017.

[7] M. G. et al., "Gnu scientific library reference manual (3rd ed.)." [Online]. Available: http://www.gnu.org/software/gsl/