Inverse Kinematics for Human Fingers

Lu Guowei (6374395) Jajodia Rahul (6544088)

October 29, 2018

Abstract

In this paper, we applied a kinematic approach to study the human finger for both constrained and unconstrained joint angles. We used forward kinematics to solve an unconstrained index finger of the human and inverse kinematics to design three different solvers for a constrained index finger. The three solvers were implemented using: Psuedo-inverse method, Psuedo-inverse method with Optimization Derivation and Extended Jacobian method; We also compared the performance of the solvers.

1 Introduction

There exists two widely-known approaches for animating articulated structures like the human fingers. The first approach is called **Forward Kinematics**, using which we can compute the position of the end-effector from inputted values for the joint parameters. The second approach is called **Inverse Kinematics**, using which we attempt to determine the joint parameters for a given position of the end-effector.
If we define the coordinates of a manipulator as the $n-dimensional$ vector of joint angles $\theta \in R^n$, and the position and orientation of the manipulator's end-effector as the $m-dimensional$ vector $x \in R^m$, the forward kinematics function can be defined as:

$$f(\theta) = x \tag{1}$$

while the inverse kinematics function is defined as:

$$f^{-1}(x) = \theta \tag{2}$$

A first-order Taylor expansion uses the Jacobian matrix $(m \times n)$

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} \tag{3}$$

to relate the change of status vector to a change of the end-effector. The relation can be defined as:

$$dx = J(\theta)d\theta \tag{4}$$

Equation 4 can be rewritten to solve for a desired(or known) dx and unknown $d\theta$ by taking the inverse of $J(\theta)$:

$$d\theta = J^{-1}(\theta)dx \tag{5}$$

If $J(\theta)$ is a square $(m = n)$ and non-singular, $J^{-1}(\theta)$ in Equation 5 can be easily computed. However, articulated structures are usually redundant, i.e., $n > m$. For these cases, there are several known alternatives of Equation 5. In the experiment, we provide three solvers:

1. **Pseudo Inverse**[1.1],
2. **Pseudo Inverse with Optimization Derivation** [1.2], and
3. **EJM**[1.3].

1.1 Pseudo Inverse

For any $m \times n$ matrix A such that $m < n$, the inverse of Equation 4 can be computed by using the Moore-Penrose right inverse:

$$A^+ = A^T(AA^T)^{-1} \tag{6}$$

which can then be used to rewrite Equation 5 as:

$$d\theta = J^+(\theta)dx = J^T(JJ^T)^{-1}dx \tag{7}$$

1.2 Pseudo Inverse with Optimization Derivation

Equation 7 is computationally fast but not robust as *this solution to the inverse kinematics problem does not generate joint angle trajectories which avoid singular configurations in any practical sense* [1]. An advantage of using the pseudo-inverse is the option of adding a 'null-space vector'. If A is an $m \times n$ matrix such that $m < n$ and A^+ $(n \times m)$ is the Moore-Penrose right inverse such that

$$I = A^+A \tag{8}$$

then $I - A^+A$ is the orthogonal projection of R^n onto the null space of A. Using this, we can modify Equation 7 to define joint angle trajectories without influencing the results as:

$$d\theta = J^+(\theta)dx + \alpha(I - J^+J)\frac{\partial g}{\partial \theta} \tag{9}$$

where α is a positive scalar constant and $g(\theta)$ an optimization criterion [more on this in subsection 1.4] that is used to calculate gradient $\frac{\partial g}{\partial \theta}$ and project it to the null-space of the matrix J. If function g does not encounter any singular configurations, then there exists a continuous $\frac{\partial g}{\partial \theta}$ which generates θ via Equation 9.

1.3 Extended Jacobian Method (EJM)

Pseudo-inverse methods are not conservative, and if there is constrain such as joint limits, which can generate other unavoidable singular configuration. The Extended Jacobian Method was suggested by Baillieul as a solution to achieve a conservative resolved motion rate control (RMRC) solution and avoid singularities. $r = n - m$ rows are added to J with a goal to zero the gradient of the optimization criterion $(\frac{\partial g}{\partial \theta})$ in the null-space of the Jacobian[2]. We can get the SVD of the Jacobian matrix and get $V_N = [\eta_1, ..., \eta_{n-m}]$ to be the orthonormal set of vectors that span the null space of J. The EJM can then be defined as:

$$J_{ext} = \begin{bmatrix} J \\ \cdots \\ \frac{\partial G_1}{\partial \theta} \\ \vdots \\ \frac{\partial G_{n-m}}{\partial \theta} \end{bmatrix} \tag{10}$$

where $G = \frac{\partial g^T}{\partial \theta}V_N$. The inverse kinematic equation then used is:

$$\begin{pmatrix} d\theta \\ 0 \end{pmatrix} = \alpha J_{ext}^{-1}(\theta) \begin{pmatrix} dx \\ 0 \end{pmatrix} \tag{11}$$

where α is a positive scalar constant.

1.4 Optimization function g

The optimization function g is used to maximize a given criterion, θ. In this paper, we use two different optimization functions:

- Manipulatability index, w: The manipulatability of a robot in a given configuration θ was first introduced by Yoshikawa[4]. It is defined as:

$$w = \sqrt{\det(J(\theta)^T J(\theta))} \tag{12}$$

  w is the product of eigenvectors of J, which means that when $w = 0$, J is singular. Hence, this function helps in avoiding singularities.

- Gradient descent function, $H(\theta)$: This function is needed to avoid the joint limits as human fingers have constrained angles.

2 Experiments: Setup and Results

To make the experiment similar to a possible scenario in the real world, we chose the index finger as our manipulator. In all experiments, all joint axes were aligned parallel to the z-axis. Thus, all motions of the finger take place in the x-y plane. The space of the finger contained a single unbounded object $O = \{(x, y, z) \in R \mid y + 2 \leq 0\}$. A step-size of 0.1 radians was used for θ_M, θ_P and θ_D. We also assume $\alpha = 1$ for simplicity. Table 1 provides the length of the phalanges we used for our experiments.

Proximal Phalanx (PP)	Intermediate Phalanx (IP)	Distal Phalanx (DP)
39.8	22.4	15.8

Table 1: Average lengths (in mm) of the phalanges of the index finger

2.1 Experiment 1: Forward kinematics

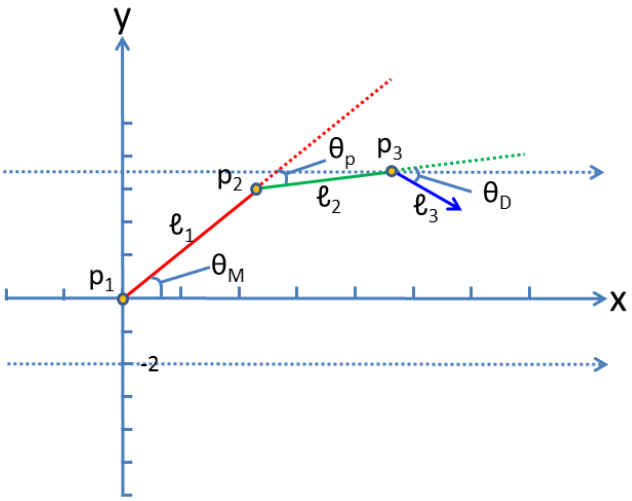

Figure 1: The index finger as 3R planar manipulator

The first experiment aims at forward kinematics of the index finger. As Figure 1 shows, we chose the right-handed coordinate system. l_1 (red line) is the PP, l_2 (green line) is the IP and l_3 (blue

line) is the DP. The arrow of l_3 denotes the orientation of the fingertip. p_1, p_2 and p_3 are the joints of the index finger depicted as points and θ_M, θ_P and θ_D are the joint angles. We neglect potential intersections of the phalanges and assume these constraints:

$$\left\{ \begin{array}{l} -\pi/3 \leq \theta_M \leq \pi/3 \\ -2\pi/3 \leq \theta_P \leq 0 \\ -2\pi/3 \leq \theta_D \leq 0 \end{array} \right\} \tag{13}$$

Using this information, we derived the forward kinematics equation for the system and explored a set of reachable positions of the fingertip (and the orientation of the distal phalanx) ignoring the obstacle O. We also explored the set of reachable positions of the fingertip when we place the DP vertically, or the PP horizontally.

Results

The coordinates of the Cartesian position is first used to represent the planar manipulator as:

$$\left\{ \begin{array}{l} X = l_1 cos(\theta_M) + l_2 cos(\theta_M + \theta_P) + l_3 cos(\theta_M + \theta_P + \theta_D) \\ Y = l_1 sin(\theta_M) + l_2 sin(\theta_M + \theta_P) + l_3 sin(\theta_M + \theta_P + \theta_D) \\ \Theta_{Orientation} = \theta_M + \theta_P + \theta_D \end{array} \right\} \tag{14}$$

Using the constraints [from Equation 13], we can simulate the set of reachable positions of the fingertip [See Figure 2]. The pentagram marks the mean of all such positions.

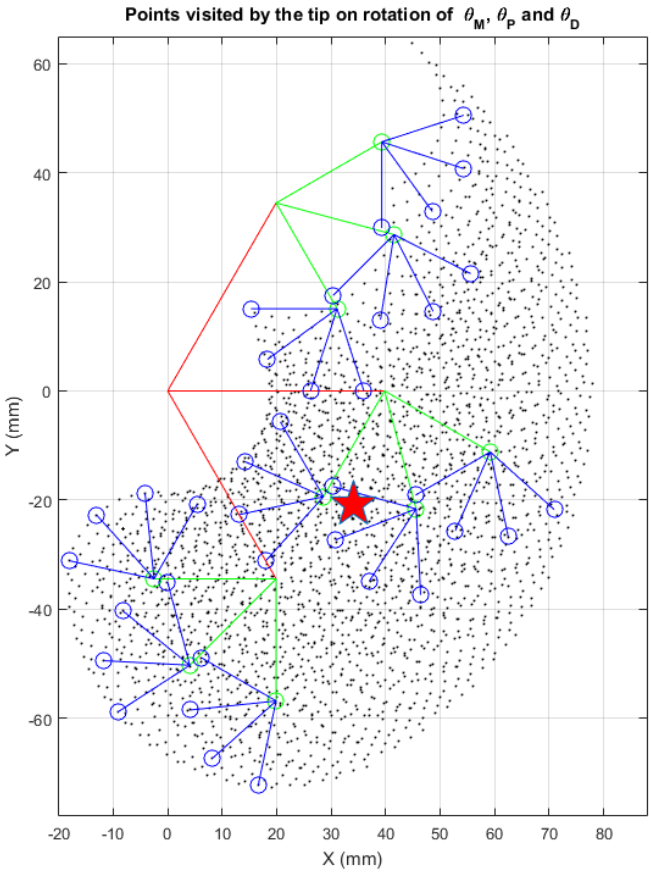

Figure 2: Set of reachable positions of the fingertip

For the PP to be horizontal, $\theta_M = 0$ is a necessary condition. This helps in simplifying Equation 14 to

$$\left.\begin{aligned} X - l_1 &= l_2 cos(\theta_P) + l_3 cos(\theta_P + \theta_D) \\ Y \quad\quad &= l_2 sin(\theta_P) + l_3 sin(\theta_P + \theta_D) \\ \Theta_{Orientation} &= \theta_P + \theta_D \end{aligned}\right\} \tag{15}$$

Let $A = X - l_1$ and $B = Y$. We can now use Equation 15 to solve for θ_P and θ_D.

$$\left.\begin{aligned} \theta_M &= 0 \\ \theta_P &= atan2\frac{B}{A} - atan2\frac{l_3 \sin\theta_D}{l_2 + l_3 \cos\theta_D} \\ \theta_D &= \pm 2atan2\sqrt{\frac{(l_2 + l_3)^2 - (A^2 + B^2)}{(A^2 + B^2) - (l_2 - l_3)^2}} \end{aligned}\right\} \tag{16}$$

Similarly, for the DP to be vertical, $\theta_M + \theta_P + \theta_D = -\pi/2$ must be satisfied. We can use this information to simplify Equation 14 to

$$\left.\begin{aligned} X \quad\quad &= l_1 cos(\theta_M) + l_2 cos(\theta_M + \theta_P) \\ Y + l_3 &= l_1 sin(\theta_M) + l_2 sin(\theta_M + \theta_P) \\ \Theta_{Orientation} &= -\pi/2 \end{aligned}\right\} \tag{17}$$

Finally, Equation 15 and Equation 17 are used to compute the set of reachable positions by the fingertip under these two conditional requirements [See Figure 3].

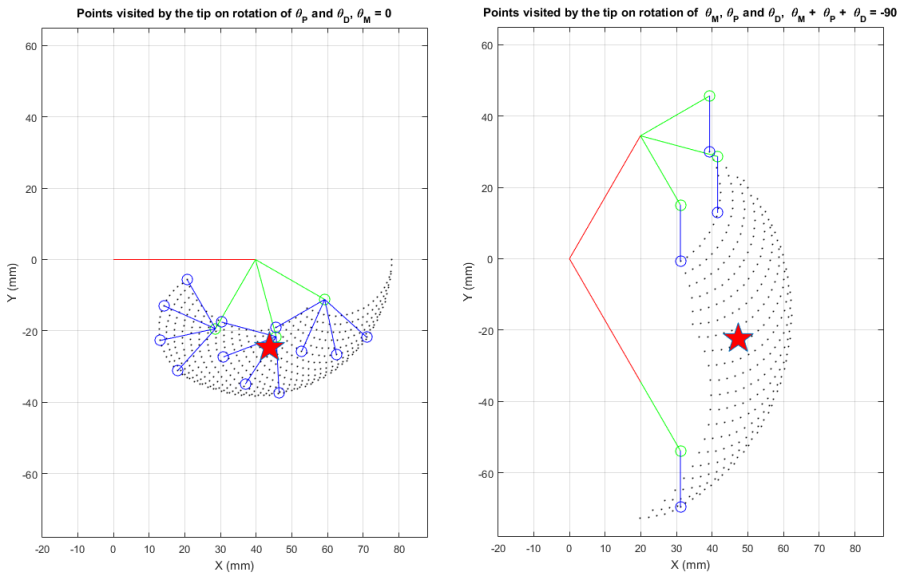

Figure 3: the set of positions of the fingertips under conditional requirements (left = horizontal PP, right = vertical DP)

2.2 Experiment 2: Inverse kinematics with constrained joint angles

For this experiment, we use the same parameters as experiment 1 (subsection 2.1), but instead use inverse kinematics to develop an iterative solver to approximate the joint angles given the desired position of the end-effector. Biologically, rotations of the PP and DP are not independent. Thus, another constraint, $\theta_D = \frac{2}{3}\theta_p$ was added to Equation 13 to accommodate for this. We also know that

- The proximal phalanx is horizontal when $\theta_M = 0$
- The distal phalanx is vertical when $\theta_M + \theta_P + \theta_D = -\pi/2$

Results We chose three Jacobian solvers:

- Pseudo-Inverse (Equation 7)
- Pseudo-Inverse with OD (Equation 9)
- EJM (Equation 11)

and created two gradient descent function for the joint limits from Equation 13

$$H(\theta) = \left\{ \begin{array}{l} \frac{1}{3}\sum_{i=1}^{3}(\frac{\theta_i - a_i}{a_i - \theta_{iMax}})^2 \ldots (i) \\ \frac{1}{6}\sum_{i=1}^{3}(\frac{\theta_{iMax} - \theta_{imin}}{(\theta_{iMax} - \theta_i)(\theta_i - \theta_{imin})}) \ldots (ii) \end{array} \right\} \tag{18}$$

where $a_i = (\theta_{iMax} + \theta_{imin})/2$. Figure 4 shows these two functions. The blue line represents equation (i) (retrieved from [5]) where we see that the gradient descent is not so significant but the computation is fast. The red line represents equation (ii) (retrieved from [4]) where the gradient descent is indeed significant but the computation is heavy thus slow. This function provides an elastic line to stretch the joint angles under the average of the joint limits.

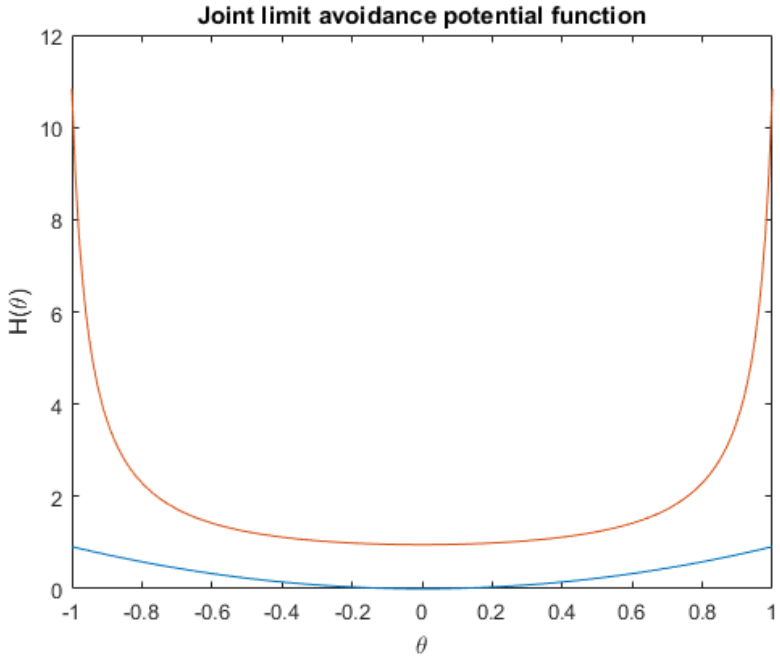

Figure 4: The potential function rises when approaching joint limits.

After that preparation, we can get three different Jacobian Matrices:

$$J = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ G_1 & G_2 & G_3 \end{bmatrix} \tag{19}$$

where

$f_{11} = -l_1 sin(\theta_M) - l_2 sin(\theta_M + \theta_P) - l_3 sin(\theta_M + \theta_P + \theta_D)$
$f_{12} = -l_2 sin(\theta_M + \theta_P) - l_3 sin(\theta_M + \theta_P + \theta_D)$
$f_{13} = -l_3 sin(\theta_M + \theta_P + \theta_D)$
$f_{21} = l_1 cos(\theta_M) + l_2 cos(\theta_M + \theta_P) + l_3 cos(\theta_M + \theta_P + \theta_D)$
$f_{22} = l_2 cos(\theta_M + \theta_P) + l_3 cos(\theta_M + \theta_P + \theta_D)$
$f_{23} = l_3 cos(\theta_M + \theta_P + \theta_D)$

Table 2: coefficients of the Jacobian matrix

Using these we can develop the iterative inverse kinematics algorithms (see algorithm 1). A different J^{-1} is computed for each solver.

Result: Find three joint angles to place the fingertip at the desired point

Set target as the desired point;
Set q as the guessed joint angles ;
if *nTime = 1:1:50 (at most 50 iterations)* **then**
  $\Delta Distance$ = target - Position(q) ;
  if $\Delta Distance \geq$ *Tolerance* **then**
    $q = J^{-1} * \delta Distance$;
    adjustByConstraint(q);
  else
    return q ;
  end
end

Algorithm 1: The iterative inverse kinematics solver

This algorithm was then applied to the EJM to find the joint angles which can place the fingertip at that point under different conditional requirements within a tolerance of $1mm$.

index	θ_M	θ_P	θ_D	X	Y	Steps	Condition
1	-41.3442	-112.8990	-100.1748	5.4	-20.7	6	extreme negative angle
2	18.904	-16.1462	-89.0251	61.1925	-1.94756	3	common position
3	3.94493	-3.35635	-11.2536	78	0	4	all angles equal 0
4	0	-9.78677	-86.2075	60	-20	4	proximal phalanx is horizontal
5	0	-82.9396	-78.2209	25	-30	13	proximal phalanx is horizontal
6	25.36731	-101.7524	-13.61487	40.69977	-17.8173	3	distal phalanx is vertical
7	-13.1946	-75.5033	-1.30215	39.748	-47.6426	3	distal phalanx is vertical
8	-16.4127	-56.1869	-37.4579	39.748	-47.6426	7	$\theta_D = \frac{2\theta_P}{3}$
9	-21.5717	-24.9438	-16.6292	61.1925	-45.8281	4	$\theta_D = \frac{2\theta_P}{3}$
10	-14.5945	-56.4507	-37.6338	39.748	-47.6426	2	distal vertical with guessed position

Table 3: Iterative counts of the EJM solver under 4 conditional requirements

We choose two special positions: when $index = 0$, all joint angles must be the minimum angle, when $index = 3$, all joint angles are zero. The solver is robust to cope with these situations. Meanwhile, when a conditional requirement is applied, the solver can find one solution to touch this point. Finally, with a good guess, the steps needed can be reduced. In order to find a good guess, we created a table to store the sampling positions from Figure 2. A total of 343 points were stored and the **nearest position in the table** was chosen as the best guess.

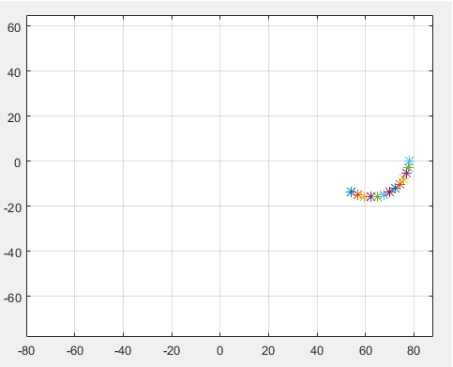

Figure 5: A sequence of points at regular small distances on the surface of O

A sequence of points at regular small distances on the surface of O (see Figure 5) were chosen and the solver was made to touch these points one by one in an anti-clockwise sequence. The position of the previous point was used as the best guess. Table 4 shows the number of steps our solver needed for each point in this sequence, the tolerance is $1mm$. When the manipulator is to be placed horizontally ($index = 14$), it needs more steps. This is caused due to the gradient descent function we chose, as for horizontal placements θ_P and θ_D both approach 0 which is the maximum value of these joint limits.

index	θ_M	θ_P	θ_D	X	Y	Steps
1	20.4523	-51.8871	-66.3252	54.3	-13.6832	3
2	17.8444	-47.7891	-61.9997	56.7961	-14.8471	1
3	15.6035	-43.3093	-57.7676	59.4564	-15.56	1
4	13.4907	-38.5321	-53.4878	62.2	-15.8	1
5	11.4899	-33.5092	-49.1801	64.9436	-15.56	1
6	9.57881	-28.28	-44.8649	67.6039	-14.8471	1
7	7.73391	-22.8658	-40.5646	70.1	-13.6832	1
8	5.92813	-17.2681	-36.3036	72.356	-12.1035	1
9	4.12534	-11.4598	-32.1077	74.3035	-10.156	1
10	2.26448	-5.35671	-27.9986	75.8832	-7.9	1
11	0.200336	1.28867	-23.9617	77.0471	-5.40392	1
12	0.200336	1.28867	-23.9617	77.0471	-5.40392	1
13	-2.65512	9.52646	-19.7116	77.76	-2.74364	1
14	3.55843	-0.523552	-16.1844	78	0	6

Table 4: Steps needed to touch a sequence of points in the EJM solver

Finally, we compared the difference between two solvers. We choose a point (5.4,-20.7) and developed one solver using Pseudo-inverse method and another using EJM. As seen in Figure 6, the EJM solver needed less steps to find the joint angles and the change in the joint angles was more stable and smooth.

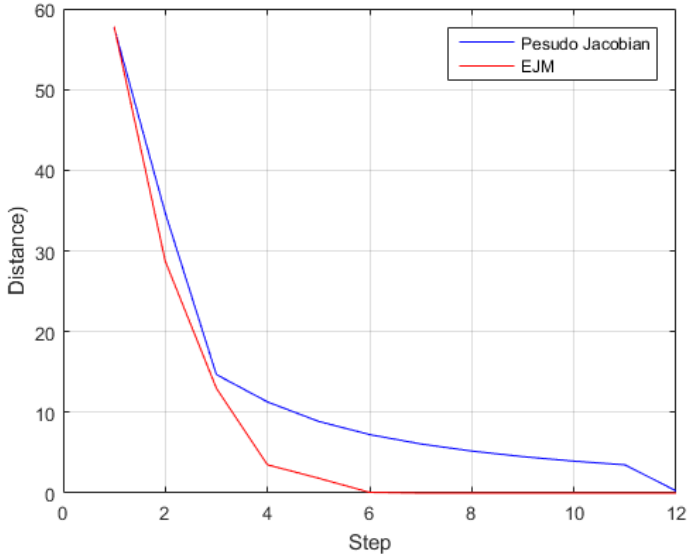

Figure 6: The distance of each step in two solvers

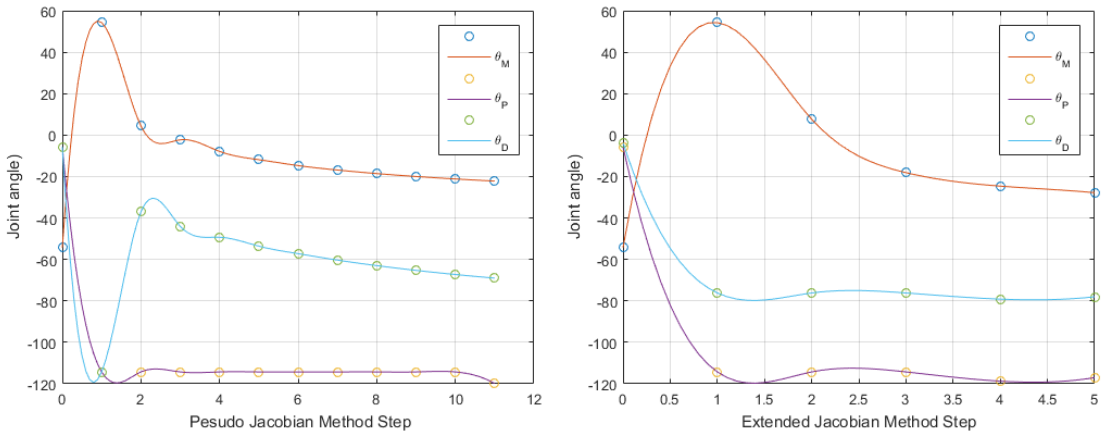

Figure 7: The joint angles of each step in two solvers

3 Evaluation and discussion

The primary goal of this paper was to find an inverse kinematics solver for the human finger. Based on the results, we can conclude that: **The Extended Jacobian Method is robust and conservative, and with a suitable gradient projection needs less steps to find the joint angles.**

In common situations, the Pseudo-inverse Jacobian Method (PJM) suffices and is computationally fast than EJM, but in extreme situations such as when joint angles are horizontal or near the joint limits, it becomes unstable. Theoretically, it could cause singularity problems by making the Jacobian matrix singular.

Perhaps the biggest surprise for us was the Pseudo-inverse Jacobian Method with Optimization Derivation (OD). It is quite unstable. Mathematically, it should be similar to PJM as both are not conservative where PJM with OD provides control of constraints. On testing the solver for

PJM with OD, we noticed that sometimes it is as efficient EJM but sometimes it is even worse than PJM. The reason for this is uncertain but a possible explanation could be the α we chose for the experiments and the gradient descent function itself.

However, in this experiment, the influence of cost function was not studied. To touch a sequence of points the influence of gradient function is significant, which makes it harder to touch certain points. We also did not find the value of manipulatability index as the optimization function. So, there are still many unsolved problems in this research.

4 Conclusion

In this paper we explored forward kinematics and inverse kinematics. We designed three solvers using different Jacobian methods and tested their performance under different conditional requirements. From the analysis of the result, we concluded that EJM is the better solver in this experiment. Additionally, in our experiments the impact of the solver implemented using PJM with OD is not significant which was unexpected. For future work we could find a better cost function and optimized the computational performance of EJM further.

References

[1] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985, pp. 722-728. doi: 10.1109/ROBOT.1985.1087234 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1087234&isnumber=23642

[2] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 294-299 vol.1. doi: 10.1109/ROBOT.2000.844073 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=844073&isnumber=18235

[3] Šoch, M., Lórencz, R. (2005). Solving inverse kinematics—a new approach to the extended Jacobian technique. Acta Polytechnica, 45, 21–26. https://ojs.cvut.cz/ojs/index.php/ap/article/view/680/512

[4] COLOME, A., Smooth Inverse Kinematics Algorithms for Serial Redundant Robots, Master Thesis, Institut de Robotica i Informatica Industrial (IRI), Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, Sep. 2011. https://pdfs.semanticscholar.org/673d/1b06d968216933d298e0eecff9fa20a2514b.pdf

[5] Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Mu—tibody Mechanisms," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-7, No. 12, December, 1977. https://ci.nii.ac.jp/naid/80014975771/