Analytical Inverse Kinematics for Franka Emika Panda – a Geometrical Solver for 7-DOF Manipulators with Unconventional Design

Yanhao He
Institute of Control Systems
University of Kaiserslautern
Kaiserslautern, Germany
yanhao@eit.uni-kl.de

Steven Liu, *Member, IEEE*
Institute of Control Systems
University of Kaiserslautern
Kaiserslautern, Germany
sliu@eit.uni-kl.de

***Abstract*—This paper presents an analytical inverse kinematic solution for the 7-DOF Franka Emika Panda robot with joint offsets. Contrary to the common belief that this problem has been thoroughly studied, there is no published work that discusses the existence of multiple solutions from a pure geometrical perspective. By investigating the geometry of the robot, the proposed solution reveals 8 different joint configurations for a given end effector pose and fixed redundant parameter, and a non-iterative algorithm is designed to consistently obtain one of them during motion control without random switching. The solver is tested in terms of accuracy and computational speed, and data show that accurate results can be obtained within a few microseconds, making it highly preferable for online trajectory planning and optimisation. The presented methodology can be extended to other 7-DOF anthropomorphic manipulators.**

***Index Terms*—Robotics, Inverse Kinematics, Redundant Manipulator**

I. INTRODUCTION

Inverse Kinematics (IK) is an important task in industrial robots for allowing motion planning and control in the position level. However unlike forward kinematics, it is a more difficult problem, because there may or may not be a solution, and when solution exists, there can be multiple ones. As pointed out by [1] and [2], a 6-Degree-Of-Freedom (DOF) robot arm can have up to 16 different joint configurations leading to the same end effector pose. For robots with even more DOFs, i.e. redundant manipulators, even infinite solutions may exist and none of them would have closed form. Another critical problem is that analytical IK derivation often varies with robot designs because each type of kinematic structure may need a different decoupling technique.

Franka Emika Panda robot manipulator is a relatively new 7-DOF product introduced in 2017, appearing more and more often in papers and tutorials. Some fundamental research about this robot including dynamic parameter identification [3] has been conducted, however its analytical IK has not been discussed yet. Its built-in IK algorithm is unfortunately not open to user, and for safety reason it can only tackle an end effector pose that is very close to the actual one.

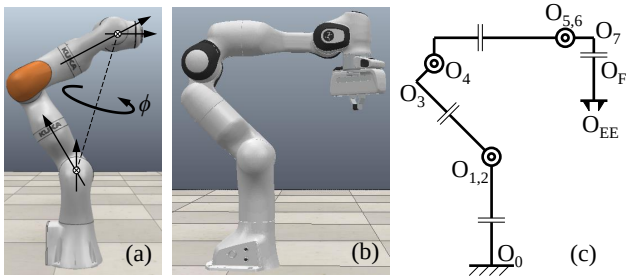

Fig. 1: The difference between a conventional 7-DOF robot arm and Franka robot arm

Most existing 7-DOF robots have an Euler shoulder and Euler wrist, as shown in Fig. 1.a, where a so-called "self-motion" about the line connecting the shoulder and the wrist can be characterised, and an analytical solution can thereby be obtained [4]. The Franka manipulator, on the other hand, has an offset wrist and offset elbow, as shown in Fig. 1.b and 1.c. This small kinematic difference makes the conventional parametrisation infeasible. Similar design can be found on a few new 7-DOF products, e.g. Kassow [5] and OB7 [6]. The authors of [7]–[9] claim to solve such IK problems analytically, but they have ignored multiple solutions and mathematical singularity. By fixing one joint angle as the redundant parameter, the IK problem can be reduced to 6-dimensional and some well-known methods e.g. in [10], [11] could be used, but their results are obtained by solving a set of complicated and coupled nonlinear equations without much geometrical insight. Similar problems exist for computer-derived analytical solvers like *IKFast* [12]. Iterative numerical methods [13] are often applied in practice to arbitrary robot types, but their common issue is the inability to return all possible solutions, and they tend to be slower. For example in [14] the motion planner based on numerical IK cost 200ms to compute for 24 waypoints with a high-end GPU.

In this paper a novel analytical IK solver for Franka Emika Panda will be derived completely based on robot geometry, i.e. all intermediate equations are geometrically interpretable.

The existence of multiple solutions are explicitly analysed. More importantly, the methodology is generalisable, leading to common rules for obtaining all solutions of other 7-DOF anthropomorphic robots with offset joints. At the time of writing, no published work is found dealing with this problem completely from a geometrical perspective. Furthermore, as a practical contribution, a "case-consistent" solver algorithm preventing the robot from randomly switching between multiple solutions will be designed.

II. FORWARD KINEMATICS

The kinematic model of Franka Emika Panda provided in its official documentation is based on the modified Denavit-Hartenberg convention, as shown in Fig. 2. The length parameters are denoted as a_i, d_i. By calculating the 4-by-4 homogeneous transformation matrices

$$ {}^{j}\boldsymbol{A}_i = \begin{bmatrix} {}^{j}\boldsymbol{R}_i & {}^{j}\boldsymbol{p}_i \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{1} $$

where ${}^{j}\boldsymbol{R}_i$ denotes their relative orientation and ${}^{j}\boldsymbol{p}_i$ the relative position, the Cartesian pose of the end effector can then be obtained by a chained multiplication:

$$ \boldsymbol{T}_{\mathrm{EE}} = \boldsymbol{A}_1 \cdot {}^{1}\boldsymbol{A}_2 \cdot {}^{2}\boldsymbol{A}_3 \cdots {}^{6}\boldsymbol{A}_7 \cdot {}^{7}\boldsymbol{A}_{\mathrm{F}} \cdot {}^{\mathrm{F}}\boldsymbol{A}_{\mathrm{EE}} \tag{2} $$

Frame	a (m)	d (m)	α (rad)	θ (rad)
Joint 1	0	0.333	0	q_1
Joint 2	0	0	-π/2	q_2
Joint 3	0	0.316	π/2	q_3
Joint 4	0.0825	0	π/2	q_4
Joint 5	-0.0825	0.384	-π/2	q_5
Joint 6	0	0	π/2	q_6
Joint 7	0.088	0	π/2	q_7
Flange (F)	0	0.107	0	0
End effector (EE)	0	0.1034	0	π/4

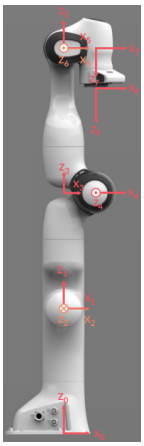

Fig. 2: The DH frames and parameters from Franka official documentation [15], with q_i being the input joint angle; the joint limits can be found on the same page in reference

In the remaining part of this paper, $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ will represent the unit vector for the X, Y, Z axes of a frame. O refers to the origin of a frame, as shown in Fig. 1. A right subscript will be used to indicate which frame the variable belongs to. Left superscript indicates the reference frame of the vector/rotation. E.g. ${}^{j}\boldsymbol{R}_i = [{}^{j}\boldsymbol{x}_i \ {}^{j}\boldsymbol{y}_i \ {}^{j}\boldsymbol{z}_i]$ is the orientation of Frame i represented in Frame j. When the reference frame is the base frame, i.e. Frame 0, the left superscript is omitted.

III. GEOMETRICAL INVERSE KINEMATICS

A. Fixing q_7 as redundancy parameter

As mentioned in Section I, to obtain a unique IK solution for a 7-DOF manipulator, a redundancy parameter must be specified first. When the Cartesian pose of the end effector is fixed, the offset on Link 6 makes it impossible to move the last joint without inducing translational motion of Frame 6, therefore the last three joints are no longer equivalent to a spherical joint and the commonly used technique that parametrises the "self-motion" fails. In this paper, the calculation starts from the end of the kinematic chain, hence the last joint angle q_7 is a natural alternative choice as the redundancy parameter. Its value will be treated as known in the remaining steps.

B. Calculating q_4

To solve q_4, triangle $\triangle \mathrm{O_2O_4O_6}$ is to be investigated. Since $\overline{\mathrm{O_2O_4}}$ and $\overline{\mathrm{O_4O_6}}$ are constant, which can be calculated with

$$ \overline{\mathrm{O_2O_4}} = \sqrt{d_3^2 + a_4^2} \tag{3} $$

$$ \overline{\mathrm{O_4O_6}} = \sqrt{d_5^2 + a_5^2} \tag{4} $$

and $\boldsymbol{p}_2$, the position of $\mathrm{O_2}$, is fixed due to the robot design, the first task in this part is to compute $\boldsymbol{p}_6$.

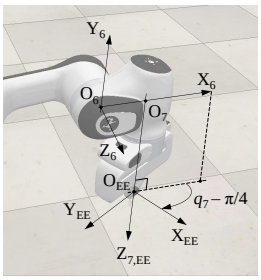

Fig. 3: Calculating the position of Frame 6

As shown in Fig. 3, the position of Frame 7 can be first calculated

$$ \begin{aligned} \boldsymbol{p}_7 &= \boldsymbol{p}_{\mathrm{EE}} + \overrightarrow{\mathrm{O_{EE}O_7}} \\ &= \boldsymbol{p}_{\mathrm{EE}} - (d_{\mathrm{F}} + d_{\mathrm{EE}})\, \boldsymbol{z}_{\mathrm{EE}} \end{aligned} \tag{5} $$

By the manufacturer definition, there is a fixed 45° offset between the flange and the end effector frames. Based on Fig. 3 it can be seen that $\boldsymbol{x}_{\mathrm{EE}}$ is obtained by rotating $\boldsymbol{x}_6$ about $\boldsymbol{z}_{\mathrm{EE}}$ by $q_7' = q_7 - \pi/4$, therefore $\boldsymbol{x}_6$ can be represented in the end effector frame as

$$ {}^{\mathrm{EE}}\boldsymbol{x}_6 = [\, \cos(-q_7') \quad \sin(-q_7') \quad 0 \,]^{\mathrm{T}} \tag{6} $$

And its representation in the world frame is

$$ \boldsymbol{x}_6 = \boldsymbol{R}_{\mathrm{EE}} \cdot {}^{\mathrm{EE}}\boldsymbol{x}_6 \tag{7} $$

The position of Frame 6 and its distance to Frame 2 can then be obtained

$$ \boldsymbol{p}_6 = \boldsymbol{p}_7 + \overrightarrow{\mathrm{O_7O_6}} = \boldsymbol{p}_7 - a_7 \boldsymbol{x}_6 \tag{8} $$

$$ \overline{\mathrm{O_2O_6}} = ||\overrightarrow{\mathrm{O_2O_6}}|| = ||\boldsymbol{p}_6 - \boldsymbol{p}_2|| \tag{9} $$

where

$$ \boldsymbol{p}_2 = [\, 0 \ \ 0 \ \ d_1 \,]^{\mathrm{T}} \tag{10} $$

Theoretically two equivalent cases, denoted as A1 and A2 in this work, exist for solving q_4, as shown in Fig. 4. The triangle

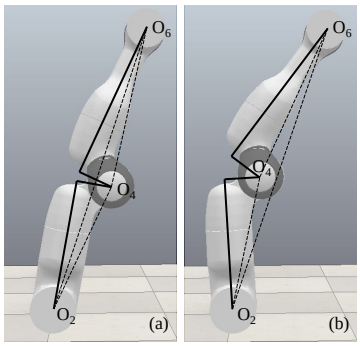

Fig. 4: Two equivalent cases when solving Joint 4 angle, with Case A1 in (a) and A2 in (b)

$\triangle O_2O_4O_6$ in Fig. 4.a and that in Fig. 4.b are symmetric about line O_2O_6, and they lead to the same position of Frame 6, thus also the same end effector pose. This "elbow-up/down" bifurcation occurs on many other robots, too. However on Franka Emika Panda due to the shape of the elbow as well as mechanical joint limit, the available room for joint motion in Case A1 is extremely restricted, with $q_4 \in [-26.76°, -4°]$, making it difficult to use in practice, therefore only A2 will be solved in this work. When needed, A1 could be solved very similarly.

For better visualisation, another configuration belonging to Case A2 is shown in Fig. 5. It is obvious that

$$q_4 = \angle O_2O_4O_3 + \angle HO_4O_6 + \angle O_2O_4O_6 - 2\pi \tag{11}$$

with

$$\begin{aligned} \angle O_2O_4O_3 &= \operatorname{atan}(d_3/a_4) \\ \angle HO_4O_6 &= \operatorname{atan}(d_5/|a_5|) \\ \angle O_2O_4O_6 &= \operatorname{acos} \frac{\overline{O_2O_4}^2 + \overline{O_4O_6}^2 - \overline{O_2O_6}^2}{2 \cdot \overline{O_2O_4} \cdot \overline{O_4O_6}} \end{aligned} \tag{12}$$

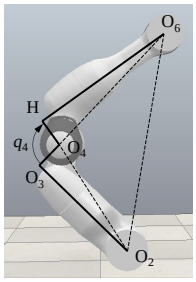

Fig. 5: Calculating Joint 4 angle

C. Calculating q_6

For the solution of q_6 there are also two equivalent cases, as shown in Fig. 6. Note that in fact due to joint limits the pose in Fig. 6.b is impossible in reality, but this is ignored in the picture just for showing the principle. The triangles $\triangle O_2O_4O_6$ in these two cases are still symmetric about line O_2O_6, however the difference between this bifurcation and the previous one with q_4 is that now q_4 remains the same, while q_3 and q_5 change by 180 degrees.

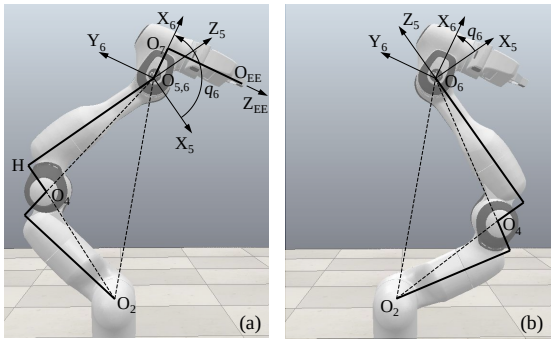

Fig. 6: Two cases for calculating Joint 6 angle (B1 and B2)

To solve q_6 in Fig. 6.a, $\angle O_2O_6H$ must first be calculated

$$\angle O_2O_6H = \angle O_2O_6O_4 + \angle HO_6O_4 \tag{13}$$

where

$$\begin{aligned} \angle O_2O_6O_4 &= \operatorname{acos} \frac{\overline{O_2O_6}^2 + \overline{O_4O_6}^2 - \overline{O_2O_4}^2}{2 \cdot \overline{O_2O_6} \cdot \overline{O_4O_6}} \\ \angle HO_6O_4 &= \pi/2 - \angle HO_4O_6 = \operatorname{atan}(|a_5|/d_5) \end{aligned} \tag{14}$$

and the orientation matrix of Frame 6, $\boldsymbol{R}_6$, is needed, whose first column is $\boldsymbol{x}_6$ as in Eq. (7), the second column, i.e. the Y axis, is obviously $\boldsymbol{y}_6 = -\boldsymbol{z}_{\mathrm{EE}}$, and the third column is determined by the right-hand rule

$$\boldsymbol{z}_6 = \boldsymbol{x}_6 \times \boldsymbol{y}_6 \tag{15}$$

By observing the robot structure, the following two conditions can be derived:

- Point O_5, O_6, O_7, O_{EE}, H are located in a same plane;
- The angle between $\boldsymbol{z}_5$ and $\overrightarrow{O_2O_6}$ is equal to $\angle O_2O_6H$.

Based on these conditions, a representation of $\boldsymbol{z}_5$ in Frame 6 can be obtained

$$\begin{aligned} {}^6\boldsymbol{z}_5 &= [\ \cos(-q_6 + \pi/2) \quad \sin(-q_6 + \pi/2) \quad 0\ ]^{\mathrm{T}} \\ &= [\ \sin(q_6) \quad \cos(q_6) \quad 0\ ]^{\mathrm{T}} \end{aligned} \tag{16}$$

where the $\pi/2$ offset is a result of the manufacturer definition of the zero pose, and the following condition holds

$${}^6\boldsymbol{z}_5 \cdot {}^6\overrightarrow{O_2O_6} = \overline{O_2O_6} \cdot \cos\angle O_2O_6H \tag{17}$$

Denoting the elements in ${}^6\overrightarrow{O_2O_6}$ as

$${}^6\overrightarrow{O_2O_6} = [\ {}^6x_{26} \quad {}^6y_{26} \quad {}^6z_{26}\ ]^{\mathrm{T}} = \boldsymbol{R}_6^{\mathrm{T}} \cdot \overrightarrow{O_2O_6} \tag{18}$$

and substituting Eq. (16) and (18) into Eq. (17) leads to

$$\sqrt{{}^6x_{26}^2 + {}^6y_{26}^2}\sin(q_6 + \phi_6) = \overline{O_2O_6} \cdot \cos\angle O_2O_6H \tag{19}$$

with

$$\phi_6 = \operatorname{atan2}({}^6y_{26}, {}^6x_{26}) \tag{20}$$

Due to the robot design, Eq. (20) is never singular. Let

$$\psi_6 = \operatorname{asin}\left(\frac{\overline{O_2O_6} \cdot \cos\angle O_2O_6H}{\sqrt{{}^6x_{26}^2 + {}^6y_{26}^2}}\right) \tag{21}$$

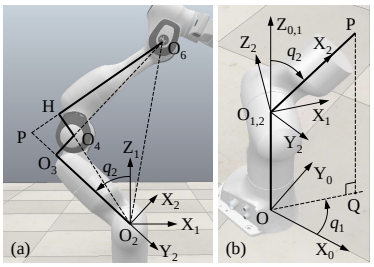

Fig. 7: Calculating the angles of Joint 1 and 2

then from Eq. (19) it can be shown that two possibilities of q_6 exist

$$q_6 = \begin{cases} \pi - \psi_6 - \phi_6 + 2k\pi \\ \psi_6 - \phi_6 + 2k\pi \end{cases} \tag{22}$$

Despite starting from Fig. 6.a, one of the solutions in Eq. (22) is actually corresponding to Fig. 6.b. The integer k should be selected in such a way that q_6 is within its joint limit range. These two q_6 solutions above will be referred to as Case B1 and B2, respectively.

D. Calculating q_1 and q_2

The intersection point of line $\mathrm{O_6H}$ and $\mathrm{O_2O_3}$ is denoted as P, as shown in Fig. 7.a. The direction of vector $\overrightarrow{\mathrm{O_2P}}$ determines q_1 and q_2. Since $\angle \mathrm{O_2O_6P} = \angle \mathrm{O_2O_6H}$ as in Eq. (13) and

$$\angle \mathrm{PO_2O_6} = \angle \mathrm{O_3O_2O_4} + \angle \mathrm{O_4O_2O_6} \tag{23}$$

it can be shown that

$$\begin{aligned} \angle \mathrm{O_2PO_6} = {} & \angle \mathrm{O_2O_6O_4} + \angle \mathrm{O_2O_4O_6} + \angle \mathrm{O_2O_4O_3} \\ & - \angle \mathrm{O_2O_6P} - \pi/2 \end{aligned} \tag{24}$$

Due to the mechanical limit of Joint 4, $\angle \mathrm{O_2PO_6}$ is never zero or π, therefore according to the law of sines

$$\overline{\mathrm{PO_6}} = \overline{\mathrm{O_2O_6}} \cdot \frac{\sin \angle \mathrm{PO_2O_6}}{\sin \angle \mathrm{O_2PO_6}} \tag{25}$$

Vector $\overrightarrow{\mathrm{O_2P}}$ is then calculated by

$$\overrightarrow{\mathrm{O_2P}} = \overrightarrow{\mathrm{O_2O_6}} + \overrightarrow{\mathrm{O_6P}} = \overrightarrow{\mathrm{O_2O_6}} - \overline{\mathrm{PO_6}} \cdot \boldsymbol{R}_6 \cdot {}^6\boldsymbol{z}_5 \tag{26}$$

There are two equivalent cases for q_1 and q_2 leading to the same $\overrightarrow{\mathrm{O_2P}}$, as illustrated in Fig. 8, where (a) and (d) lead to the same $\boldsymbol{T}_{\mathrm{EE}}$. In one case, as shown in Fig. 7.b, q_1 is the rotating angle of $\overrightarrow{\mathrm{OQ}}$ from $\boldsymbol{x}_0$, with Q being the projection of P in the base X-Y plane, and q_2 is the rotating angle of $\overrightarrow{\mathrm{O_2P}}$ from $\boldsymbol{z}_0$. By denoting the elements in $\overrightarrow{\mathrm{O_2P}}$ as

$$\overrightarrow{\mathrm{O_2P}} = [\ x_{2\mathrm{P}} \quad y_{2\mathrm{P}} \quad z_{2\mathrm{P}}\ ]^{\mathrm{T}} \tag{27}$$

it is easy to get one solution, labelled as Case C1:

$$\begin{aligned} q_1 &= \mathrm{atan2}\left(y_{2\mathrm{P}}, x_{2\mathrm{P}}\right) \\ q_2 &= \mathrm{acos}\left(z_{2\mathrm{P}} / \overline{\mathrm{O_2P}}\right) \end{aligned} \tag{28}$$

And according to Fig. 8, the other solution, labelled as Case C2, is

$$\begin{aligned} q_1 &= \mathrm{atan2}\left(-y_{2\mathrm{P}}, -x_{2\mathrm{P}}\right) \\ q_2 &= -\mathrm{acos}\left(z_{2\mathrm{P}} / \overline{\mathrm{O_2P}}\right) \end{aligned} \tag{29}$$

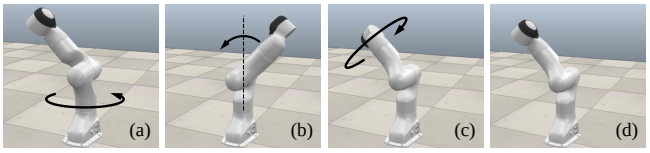

Fig. 8: Transformation between two cases for q_1 and q_2

Singularity problem may arise here, since $x_{2\mathrm{P}}$ and $y_{2\mathrm{P}}$ could be zero at the same time and the atan2 function could fail. This happens when $\overrightarrow{\mathrm{O_2P}}$ points straight up, i.e., $q_2 = 0$. As shown in Fig. 9, the links between Joint 1 and 3 can rotate freely about $\boldsymbol{z}_0$ without moving the rest of the robot body, allowing infinite solutions to q_1 and q_3. In this situation, a pre-defined value must be assigned to q_1 for a unique result.

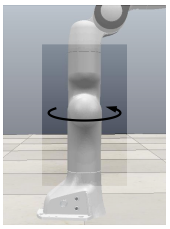

Fig. 9: IK singularity caused by $q_2 = 0$

E. Calculating q_3

It can be observed from Fig. 10 that q_3 is the rotating angle of $\overrightarrow{\mathrm{O_2M}}$ from $\boldsymbol{x}_2$, with $\overrightarrow{\mathrm{O_2M}}$ being a projected direction of $\boldsymbol{x}_3$ in the $\mathrm{Z_2}$-$\mathrm{X_2}$ plane. The Y axis of Frame 3 is perpendicular to triangle $\triangle \mathrm{O_2PO_6}$, therefore $\boldsymbol{y}_3$ is the normalisation of $\overrightarrow{\mathrm{O_2P}} \times \overrightarrow{\mathrm{O_2O_6}}$. Since $\overrightarrow{\mathrm{O_2P}}$ and $\overrightarrow{\mathrm{O_2O_6}}$ are never aligned because of Joint 4 limitation, $\boldsymbol{y}_3$ is never singular. $\boldsymbol{z}_3$ is simply the normalisation of $\overrightarrow{\mathrm{O_2P}}$. Then according to the right hand rule

$$\boldsymbol{x}_3 = \boldsymbol{y}_3 \times \boldsymbol{z}_3 \tag{30}$$

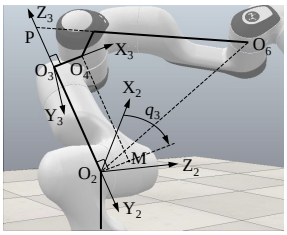

Fig. 10: Calculating Joint 3 angle

With q_1 and q_2 solved in the previous section, the orientation of Frame 2 can be obtained by forward kinematics

$$\boldsymbol{R}_2 = \boldsymbol{R}_1\left(q_1\right) \cdot {}^1\boldsymbol{R}_2\left(q_2\right) \tag{31}$$

After transforming $\boldsymbol{x}_3$ into Frame 2 and extracting its internal elements by

$${}^2\boldsymbol{x}_3 = [\ {}^2x_{x3} \quad 0 \quad {}^2z_{x3}\ ]^{\mathrm{T}} = \boldsymbol{R}_2^{\mathrm{T}} \cdot \boldsymbol{x_3} \tag{32}$$

the value of q_3 can be calculated with

$$q_3 = \mathrm{atan2}\left({}^2z_{x3}, {}^2x_{x3}\right) \tag{33}$$

F. Calculating q_5

With S being the projection of O_4 in the X_5-Y_5 plane, q_5 can be interpreted as the rotating angle of $\boldsymbol{x}_5$ from $\overrightarrow{\mathrm{O}_5\mathrm{S}}$, as depicted in Fig. 11. Vector $\overrightarrow{\mathrm{HO}_4}$ is equal to

$$\overrightarrow{\mathrm{HO}_4} = \boldsymbol{p}_4 - \boldsymbol{p}_{\mathrm{H}} = \boldsymbol{p}_4 - (\boldsymbol{p}_6 - d_5\boldsymbol{z}_5) \tag{34}$$

where from Fig. 10 it is clear that

$$\boldsymbol{p}_4 = \boldsymbol{p}_2 + d_3\boldsymbol{z}_3 + a_4\boldsymbol{x}_3 \tag{35}$$

With q_6 already solved, $\overrightarrow{\mathrm{O}_5\mathrm{S}}$ can be computed by

$$\begin{aligned}\overrightarrow{\mathrm{O}_5\mathrm{S}} &= [\ {}^5x_{5\mathrm{S}} \quad {}^5y_{5\mathrm{S}} \quad 0\ ]^{\mathrm{T}} \\ &= \boldsymbol{R}_5^{\mathrm{T}} \cdot \overrightarrow{\mathrm{HO}_4} = {}^5\boldsymbol{R}_6\left(q_6\right) \cdot \boldsymbol{R}_6^{\mathrm{T}} \cdot \overrightarrow{\mathrm{HO}_4}\end{aligned} \tag{36}$$

And then the solution to q_5 can be obtained

$$q_5 = -\mathrm{atan2}\left({}^5y_{5\mathrm{S}}, {}^5x_{5\mathrm{S}}\right) \tag{37}$$

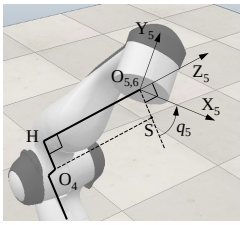

Fig. 11: Calculating Joint 5 angle

G. General rules

The preceding analysis based on Franka Emika Panda can be extended to some general procedures for the analytical derivation with other 7-DOF anthropomorphic manipulators:

1) Fix q_7 as the redundancy parameter;
2) Investigate $\triangle \mathrm{O}_2\mathrm{O}_4\mathrm{O}_6$ to solve two possible q_4's;
3) For each case of q_4, solve two possibilities of q_6;
4) For each q_4-q_6 combination, solve two possibilities of (q_1, q_2) pair;
5) For each q_4-q_6-(q_1, q_2) combination, solve the unique q_3 and q_5.

It is easy to follow that 2^3=8 different mathematical solutions could generally be obtained for a given gripper pose and redundancy parameter. Note that Step 2 works even with an offset shoulder and elbow (e.g. on ABB GoFa and OB7), so the self-motion characterisation is no longer needed at all. Actually since Euler-joint design is a special case of offset joints, the proposed technique can also be applied to conventional 7-DOF anthropomorphic robot arms.

IV. ALGORITHM IMPLEMENTATION

A. Complete solver

Ignoring the ill case A1, there exist four solutions in total: Case (A2-) B1-C1, B1-C2, B2-C1 and B2-C2. Since one or more of these mathematical results may violate the joint limits, necessary checking must be conducted at the end of the complete solver to get rid of practically invalid outputs.

Additionally the following requirement

$$\overline{\mathrm{O}_4\mathrm{O}_6} - \overline{\mathrm{O}_2\mathrm{O}_4} \leqslant \overline{\mathrm{O}_2\mathrm{O}_6} \leqslant \overline{\mathrm{O}_4\mathrm{O}_6} + \overline{\mathrm{O}_2\mathrm{O}_4} \tag{38}$$

must be examined before solving Eq. (11) to guarantee the existence of $\triangle \mathrm{O}_2\mathrm{O}_4\mathrm{O}_6$. Physically speaking, it means the desired $\boldsymbol{T}_{\mathrm{EE}}$ and q_7 must be reachable. Furthermore, when singularity is detected, i.e. $x_{2\mathrm{P}}^2 + y_{2\mathrm{P}}^2 \approx 0$, q_1 will be assigned its actual value. In general, the implemented complete solver algorithm has the following form:

$$[\boldsymbol{q}_{\mathrm{B1C1}},\ \boldsymbol{q}_{\mathrm{B1C2}},\ \boldsymbol{q}_{\mathrm{B2C1}},\ \boldsymbol{q}_{\mathrm{B2C2}}] = \mathrm{solveIK}\left(\boldsymbol{T}_{\mathrm{EE}}, q_7, \boldsymbol{q}_{\mathrm{a}}\right)$$

where $\boldsymbol{q}_{\mathrm{a}}$ is the actual joint pose.

B. Case-consistent solver

In many applications, it is not desired that the robot switch between different cases during trajectory planning and execution, which could lead to unexpected robot motion, thus a case-consistent solver is designed. This algorithm first checks which case the actual pose $\boldsymbol{q}_{\mathrm{a}}$ belongs to, and then only computes an output of the same case.

For classifying $\boldsymbol{q}_{\mathrm{a}}$, its corresponding forward kinematics should be calculated. Then based on Section III it is clear that $\boldsymbol{q}_{\mathrm{a}}$ belongs to Case B1 if

$$\overrightarrow{\mathrm{O}_2\mathrm{O}_6}\left(\boldsymbol{q}_{\mathrm{a}}\right) \cdot \boldsymbol{x}_5\left(\boldsymbol{q}_{\mathrm{a}}\right) \leqslant 0 \tag{39}$$

otherwise it is B2. And when the second joint angle in $\boldsymbol{q}_{\mathrm{a}}$

$$q_2|_{\boldsymbol{q}_{\mathrm{a}}} > 0 \tag{40}$$

$\boldsymbol{q}_{\mathrm{a}}$ belongs to Case C1, otherwise C2. The pseudocode of the proposed algorithm is attached, marked as Algorithm 1.

V. PERFORMANCE EVALUATION

A. Test in MATLAB

The case-consistent solver is first implemented in MATLAB R2019a. To get the best computation speed, a MEX function instead of an M-script is composed. The CPU is an Intel Core i5-4670 (3.4GHz) and the RAM size is 8 GB.

For the testing, 10 uniformly distributed values in the available range of each joint are selected, which in total forms 10^7 different sample poses. Note that to avoid the ill case in Fig. 4.a, q_4 is restricted below -27°. Each of these poses will be provided to a forward kinematic function developed by the author based on the official DH parameters and then its corresponding end effector pose will be used as the input to the IK solver. The other two arguments, $\boldsymbol{q}_{\mathrm{a}}$ and the desired q_7, are equal to the original sample pose.

The norms of the error between the output of the proposed solver and the original pose, i.e. $||\boldsymbol{q} - \boldsymbol{q}_{\mathrm{a}}||$, are recorded. Its average value is 7.1598e-09 (rad), and the maximum is 6.2567e-07, which is negligible in practice. Then the output $\boldsymbol{q}$ is again fed into forward kinematics to try reproducing the input $\boldsymbol{T}_{\mathrm{EE}}$. The mean norm of the translational reproducing error is 3.2509e-11 (m), and maximum 3.1231e-10. For orientation, the error is so small that only 0 was displayed.

Algorithm 1 Case-consistent IK solver

function SOLVEIK_CC($\boldsymbol{T}_{\mathrm{EE}}$, q_7, $\boldsymbol{q}_{\mathrm{a}}$)
  Compute $\boldsymbol{x}_1(\boldsymbol{q}_{\mathrm{a}})$, $\boldsymbol{y}_2(\boldsymbol{q}_{\mathrm{a}})$, $\boldsymbol{x}_5(\boldsymbol{q}_{\mathrm{a}})$, $\overrightarrow{\mathrm{O_2O_6}}(\boldsymbol{q}_{\mathrm{a}})$
  if Inequality (38) does not hold **then**
    return ERROR
  end if
  Compute q_4
  if $\overrightarrow{\mathrm{O_2O_6}}(\boldsymbol{q}_{\mathrm{a}}) \cdot \boldsymbol{x}_5(\boldsymbol{q}_{\mathrm{a}}) \leqslant 0$ **then**
    $q_6 = q_6|_{\mathrm{B1}}$
  else
    $q_6 = q_6|_{\mathrm{B2}}$
  end if
  if $x_{2\mathrm{P}}^2 + y_{2\mathrm{P}}^2 \approx 0$ **then**
    $q_1 = q_1|_{\boldsymbol{q}_{\mathrm{a}}}$, $q_2 = 0$
  else if $q_2|_{\boldsymbol{q}_{\mathrm{a}}} > 0$ **then**
    $q_1 = q_1|_{\mathrm{C1}}$, $q_2 = q_2|_{\mathrm{C1}}$
  else
    $q_1 = q_1|_{\mathrm{C2}}$, $q_2 = q_2|_{\mathrm{C2}}$
  end if
  Compute q_3, q_5
  if any q_i out of limit **then**
    return ERROR
  end if
  return $\boldsymbol{q} = [q_1,\ q_2,\ q_3,\ q_4,\ q_5,\ q_6,\ q_7]^{\mathrm{T}}$
end function

Since the original poses are sampled over the whole joint space, the small joint-space error demonstrates that the derivation has covered all possible different solutions. The average computation time is only 17.187 μs, generally much faster than numerical approaches, which often require milliseconds.

B. Test with C++

The algorithm is then implemented using C++ on an Ubuntu laptop connected to the real Franka robot in the laboratory. The computer is powered by an Intel Core 2 Duo CPU P8700 (2.53GHz) and 4GB RAM.

The same testing methodology is used, but the forward kinematics part is replaced by a Franka built-in function running in its control box. Larger joint pose errors are observed this time, with the average norm being 3.6237e-06 and maximum 4.5328e-04. The Cartesian pose reproducing error has however only very slightly increased, with the mean value of the translational part being 5.6775e-10, maximum 8.6161e-10; for orientation the average is 1.1035e-12 (rad), and maximum 1.1117e-12. The distinction is probably a result of numerical error caused by using different operating systems and compilers. Nevertheless the results are still more than satisfactory in practice.

The average computation time decreases to only 5.901 μs. Taking the outdated hardware into consideration, it is evident that a modern computer could run it even faster, making this algorithm very applicable for online motion planning and control. The C++ code can be found in the GitHub repository: https://github.com/ffall007/franka_analytical_ik .

VI. CONCLUSIONS

This paper has derived the analytical IK solution for Franka Emika Panda, a popular 7-DOF redundant manipulator with offset wrist and elbow. By investigating the geometrical relationship within the robot structure, the existence of multiple equivalent joint-space solutions is explicitly discussed, and a purely geometry-based solving process is presented, which is generalisable for the application to other redundant robot arms. Some implementation details have been highlighted and the proposed algorithm is tested in two different environments. It can be concluded from the test results that this method is able to produce very accurate outputs and its computational efficiency makes it remarkably suitable for both simulation and real-time applications.

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. Springer London Ltd, 2008.

[2] R. Manseur and K. L. Doty, "A robot manipulator with 16 real inverse kinematic solution sets," *The International Journal of Robotics Research*, vol. 8, no. 5, pp. 75–79, Oct. 1989.

[3] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. D. Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, Oct. 2019.

[4] M. Gong, X. Li, and L. Zhang, "Analytical inverse kinematics and self-motion application for 7-DOF redundant manipulator," *IEEE Access*, vol. 7, pp. 18 662–18 674, 2019.

[5] Kassow Robots. (2020) Kassow Robots - strong· fast· simple. [Accessed on 8 October 2021]. [Online]. Available: https://www.kassowrobots.com/

[6] Productive Robotics, Inc. (2020) Cobots — Productive Robotics — United States. [Accessed on 8 October 2020]. [Online]. Available: https://www.productiverobotics.com/

[7] C. Yu, M. Jin, and H. Liu, "An analytical solution for inverse kinematic of 7-DOF redundant manipulators with offset-wrist," in *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, Aug. 2012.

[8] L. Jiang, X. Huo, Y. Liu, and H. Liu, "An integrated inverse kinematic approach for the 7-DOF humanoid arm with offset wrist," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, Dec. 2013.

[9] A. Sinha and N. Chakraborty, "Geometric search-based inverse kinematics of 7-DoF redundant manipulator with multiple joint offsets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019.

[10] D. L. Pieper, "The kinematics of manipulators under computer control," Ph.D. dissertation, Stanford University, 1968. [Online]. Available: https://searchworks.stanford.edu/view/2197810

[11] R. Paul, *Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators*. Cambridge, Mass: MIT Press, 1981.

[12] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, Aug. 2010. [Online]. Available: http://www.programmingvision.com/rosen_diankov_thesis.pdf

[13] A. Colome and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 944–955, Apr. 2015.

[14] D. W. Arathorn, "Fast robot arm inverse kinematics and path planning under complex static and dynamic obstacle constraints."

[15] Franka Emika GmbH. (2017) Robot and interface specifications - Franka Control Interface (FCI) documentation. [Accessed on 6 October 2020]. [Online]. Available: https://frankaemika.github.io/docs/control_parameters.html