

GeoFIK: A Fast and Reliable Geometric Solver for the IK of the Franka Arm Based on Screw Theory Enabling Multiple Redundancy Parameters

Pablo López-Custodio, Yuhe Gong, Luis Figueredo

Abstract—Modern robotics applications require an inverse kinematics (IK) solver that is fast, robust and consistent, and that provides all possible solutions. Currently, the Franka robot arm is the most widely used manipulator in robotics research. With 7 DOFs, the IK of this robot is not only complex due to its redundancy, but also due to the link offsets at the wrist and elbow. Due to this complexity, none of the Franka IK solvers available in the literature provide satisfactory results when used in real-world applications. Therefore, in this paper we introduce GeoFIK (Geometric Franka IK), an analytical IK solver that allows the use of different joint variables to resolve the redundancy. The approach uses screw theory to describe the entire geometry of the robot, computing the Jacobian matrix prior to the joint angles. All singularities are handled. As an example of how the geometric elements obtained by the IK can be exploited, a solver with the swivel angle as the free variable is provided. Several experiments are carried out to validate the speed, robustness, and reliability of GeoFIK against three state-of-the-art solvers.

I. INTRODUCTION

Inverse kinematics (IK) plays a pivotal role in robotics, bridging task-space goals with joint-space feasibility, execution, and performance. It is the core element of a wide range of tasks, from simple pick-and-place, grasping, and collision avoidance [1], [2] to highly dynamic task-and-motion planning, high-DoF planning, safety [3], [4] and even biomechanics [5]. Unlike forward kinematics, finding IK solutions is challenging due to the non-linearity of the equations and the branching of its solutions. While the IK of 6-DOF manipulators can have up to 16 solutions [6], the problem is more complex for redundant robots, such as 7-DOF arms, as they have infinitely many solutions organized along a continuous one-parameter manifold (self-motion) [7].

Standard 7-DOF manipulators designed with spherical wrist and shoulder, offer trivial kinematics which allows for simple redundancy resolution. This is however not the case for the most widely used manipulator in robotics research of today [8]–[14], the Franka Research Robot. This manipulator features an unconventional topology with joint offsets at the wrist and elbow that break the symmetry exploited by traditional solvers. Indeed, despite the excellent work within the robotics community (e.g. tailored tools including [9]–[12], [15]) a fundamental piece is missing: a fast, robust and reliable multi-parametrization analytical IK solver.

In this context, this work presents the first IK solution for the Franka Robot based on screw theory. To address the challenges and the wide range of applications required to improve state-of-the-art methods and systems, [3]–[5], our design choice aimed to satisfy the following conditions:

P. López-Custodio is with the Computer Science Department at Nottingham Trent University, Nottingham, UK, custodio825@gmail.com. Y. Gong and L. Figueredo are with the CHART Group at the School of Computer Science at the University of Nottingham, {yuhe.gong@nottingham.ac.uk, Luis.Figueredo@nottingham.ac.uk}. L. Figueredo is also an Associated Fellow at the MIRMI-TUM.

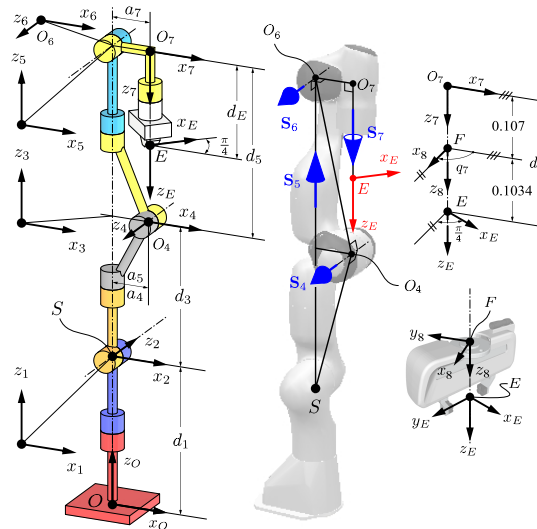


Fig. 1: Home configuration of the Franka arm with $q_i = 0$, $i = 1, \dots, 7$, and detail of the end-effector frames. Note that both frames 8 and E are rigidly attached to the gripper.

The IK solver should (i) be *faster and more robust* than any other solution, leading to solutions overlooked by other analytical methods; (ii) be *more reliable*, ensuring always finding a solution if one exists; (iii) return solutions even in *extreme singular poses* where other approaches fail; (iv) *return Jacobians* at no extra cost; (v) explore *multiple free parameters* (beyond standard q_7), thus revealing more valid configurations than existing alternatives, including (vi) using the *swivel-angle* [16]. Obtaining the Jacobian from the IK without committing to joint angles is useful for lazy-planning strategies, dynamic and static trajectory optimization, and geometric pullbacks.

The Franka robot arm has firmly established itself as a reference in robotics research, with a breadth of studies showing its versatility in both industrial and academic settings. Recent surveys have shown that more than 4700 papers from leading venues such as IEEE ICRA, RA-L, T-RO, IROS, CoRL, and RSS have been using the system with a majority of the work (33 %) focused on AI and ML.¹ It is also estimated that 90 % and 100 % of the top 10 US and EU institutions in Computer Science have the system in their labs [8]. The success of the Franka is mostly due to its tactile and compliance capabilities as well as increased safety capabilities, including a lower effective mass at its end-effector. See [17] for a comparison with other systems. As highlighted in [8], [17], achieving a safer design came with a cost, an offset at the wrist and at the elbow, which makes its IK more complex than existing systems, e.g., 7-DOF KUKA arm whose wrist,

¹Manually verified papers containing keywords *Franka robot, Emika, Panda {arm, gripper, manipulator, cobot, robot}, or 7 DoF Panda, 2017–23* [8].

elbow and shoulder are spherical.

This complex topology (Fig. 1) reduces the effective mass, critical for safety [3], [17], but introduces additional challenges for analytical and geometric analysis. This challenge is reflected in the scarce number of analytical solvers for the IK of the Franka.² Indeed, up to today, only four methods were capable to derive an analytical IK, [19]–[22]. Among these, the exceptional work by He and Liu [22] is the only complete and functional one. It is available as a C++ application [23]. However, all these publications limit the free parameter to the last joint, q_7 , which leads to an easier yet less robust analysis. To the best of our knowledge, no geometric solvers with other joint variables as free variable have been presented for the specific case of the Franka. Analytical solutions can be obtained from IKFast [24], which precomputes algebraic expressions to automatically generate a solver. The method is general, but this comes at the cost of lacking geometric insight, being slower than geometric methods, and instability in singularity handling. The lack of geometric insight prevents the user from extracting geometric information needed in certain applications. Another option to obtain analytical solutions is to use a solver for general 6-DOF robots (e.g. [25], [26]) after one of the joints has been locked. However, this involves the definition of a new kinematic structure every time the value of the locked joint is fixed which results in additional computational time. These changes in the topology of the 6-DOF robot can also lead to unexpected behavior as we show in Sec. VIII. Finally, being able to use a 6-DOF solver for the Franka involves the development of a considerable amount of code and requires non-trivial knowledge of the robot’s kinematics.

A. Contributions

Motivated by these gaps, this paper introduces **GeoFIK (Geometric Franka IK)**,³ a novel solver that allows exploring free parameters (rather than only locking q_7) while robustly handling singularities, maximizing available solutions, and offering explicit Jacobian computation at no extra cost.

In contrast to existing methods, including [19], [22], [24], which find individual expressions for the joint angles, our method explores the whole geometry of the robot, finding first all the screw axes. Hence, the Jacobian matrix can be returned as a byproduct. Then, the joint angles are systematically computed using the exponential and logarithmic maps of the special orthogonal group, $SO(3)$. Due to the use of screw theory, singularities are easily identified and handled. To our knowledge, no other IK solver for redundant manipulators works on finding the screw axes, although this is often done for mechanisms and parallel robots [27].

As an example of the extraction of geometric information before joint angle calculation, a fast numerical IK solver with the swivel angle as a free variable is presented. The *swivel angle* is roughly defined as an angle between the arm plane and a reference plane, providing a sense of posture [16]. We demonstrate how such solver allows to preserve flexibility, guaranteeing solutions even for poses that break joint-lock-based strategies, improving robustness.

²The IK can always be computed numerically by means of the *differential IK* using gradient-descent [18]. However, this strategy is hundred-to-thousand-fold slower and leads to a single solution.

³The C++ code for GeoFIK is available at <https://github.com/PabloLopezCustodio/GeoFIK>

We thoroughly compare the performance of our method against an analytical solver specifically designed for the Franka, the HeLiu solver [22]; a general analytical solver, IKFast [24]; and a 6-DOF state-of-the-art solver, IKGeo [26]. HeLiu is the go-to solver for the analytical IK of the Franka, e.g. it is currently used in the MuJoCo Playground [28]. Due to its effectiveness and generality, IKFast remains one of the most used solvers. It is, for example, the analytical solver for the popular framework MoveIt! [29]. Hence, both solvers represent valuable baselines. We illustrate that GeoFIK outperforms existing methods in terms of reliability, robustness, and speed (particularly, when computing the Jacobian and joints). Furthermore, we show scenarios at singular configurations where other solvers fail. With real-world experiments, we also show that joint-lock-based techniques inevitably falter in discovering valid poses even for predictably easier scenarios in the middle of the robot’s workspace, whereas our solver with swivel-angle as free variable consistently finds solutions and remains computationally efficient.

This paper is organized as follows: Sec. II introduces the problem statement and general method; Secs. III to VI present the solutions locking q_7 , q_6 , q_4 and swivel angle, respectively; Sec. VII discusses singularity handling; Sec. VIII provides comparisons with baselines; Sec. IX shows a real-world experiment; and conclusions are drawn in Sec. X.

II. PROBLEM FORMULATION AND METHOD DESCRIPTION

This section defines our IK problem, and outlines our screw-theory-based approach.⁴

Problem Definition: Given a desired pose of the end-effector frame⁵ E w.r.t. the base frame O defined by ${}^O\mathbf{r}_{E/O} \in \mathbb{R}^3$ and ${}^O\mathbf{R} \in SO(3)$ the goal is to find a set of joint-angles $q_i \in \mathbb{S}$, with $i=1, \dots, 7$ that leads to the desired pose of frame E .

Given the 1-DOF redundancy, one additional constraint must be provided. We provide solutions that use q_4, q_6, q_7 , or the swivel angle to resolve this redundancy. No q_5 solver is provided since it is geometrically the most complex and will be researched in future work.

Screw Theory-based IK Solution: Unlike traditional IK solvers that rely on purely algebraic formulations, our approach uses screw theory, offering a geometric perspective on the solution. The method is based on determining the screw axes of the manipulator ${}^O\mathbf{S}_i = ({}^O\hat{\mathbf{s}}_i; {}^O\mathbf{r}_i \times {}^O\hat{\mathbf{s}}_i)$, $i = 1, \dots, 7$, where ${}^O\hat{\mathbf{s}}_i$ is the direction of the axis, and ${}^O\mathbf{r}_i$ is the position of *any* point along the axis.

By using this formulation, the **Jacobian** can be computed even without prior knowledge of joint angles as $\mathbf{J} = \text{Adj}(\mathbf{I}_3, -{}^O\mathbf{r}_{E/O})[{}^O\mathbf{S}_1, \dots, {}^O\mathbf{S}_7]$, where $\text{Adj}(\cdot, \cdot)$ is the adjoint representation of an element of $SE(3)$ defined by a pair in $SO(3) \times \mathbb{R}^3$, see [31, Sec. 4.2].

We first exploit the spherical shoulder joint structure of the Franka, and hence we first find screw-axes ${}^O\mathbf{S}_4$, ${}^O\mathbf{S}_5$, and ${}^O\mathbf{S}_6$. The axis of the last joint, ${}^O\mathbf{S}_7$, is fully determined by the input, i.e., ${}^O\hat{\mathbf{s}}_7 = {}^O\hat{\mathbf{k}}_E$ and ${}^O\mathbf{r}_7 = {}^O\mathbf{r}_{E/O}$.

⁴Fig. 1 shows the frames and DH parameters of the Franka arm. We use the frame E as end-effector frame. When the gripper is connected, E matches the frame `panda_hand_tcp` in Franka Description [30]. We rename the centre of the shoulder as S .

⁵Notation: ${}^O\mathbf{r}_{P/Q} \in \mathbb{R}^3$ is the position vector of point P with respect to point Q measured in frame O . ${}^O\mathbf{R} \in SO(3)$ is the rotation matrix defining the orientation of frame E with respect to frame O . Unit vectors wear a hat, $\hat{\mathbf{u}} \in S^2$. We use the parallel symbol “ \parallel ” to avoid normalising the right-hand side when the left-hand side is a unit vector, e.i. $\hat{\mathbf{u}} \parallel \mathbf{r}$ means $\hat{\mathbf{u}} = \mathbf{r}/|\mathbf{r}|$.

Algorithm 1: Calculating the joint angles q_i from the directions of the joint axes ${}^O\hat{s}_i, i = 1, \dots, 7$

Input: ${}^O\hat{s}_{i,0}, {}^O\hat{s}_i, i = 1, \dots, 7, {}^O\hat{i}_E$

Output: q_1, \dots, q_7

- 1 $\hat{\mathbf{u}}_i \leftarrow {}^O\hat{s}_{i,0}, i = 1, \dots, 7, \hat{\mathbf{u}}_8 \leftarrow (\sqrt{2}/2, \sqrt{2}/2, 0)^\top$
- 2 $\hat{\mathbf{g}}_i \leftarrow {}^O\hat{s}_i, i = 1, \dots, 7, \hat{\mathbf{g}}_8 \leftarrow {}^O\hat{i}_E$
- 3 **for** $j \leftarrow 1$ to 7 **do**
- 4 $q_j \leftarrow \text{atan2}((\hat{\mathbf{u}}_{j+1} \times \hat{\mathbf{g}}_{j+1}) \cdot \hat{\mathbf{u}}_j, \hat{\mathbf{u}}_{j+1} \cdot \hat{\mathbf{g}}_{j+1})$
- 5 $[\hat{\mathbf{u}}_{j+1}, \dots, \hat{\mathbf{u}}_8] \leftarrow \exp(q_j \hat{\mathbf{u}}_j) [\hat{\mathbf{u}}_{j+1}, \dots, \hat{\mathbf{u}}_8]$

Once those are known, the spherical shoulder is solved as ${}^O\hat{s}_3 \parallel \exp(\arctan(a_4/d_3) {}^O\hat{s}_4) {}^O\mathbf{r}_{O_4/S}, {}^O\hat{s}_1 = (0, 0, 1)^\top,$

$${}^O\hat{s}_2^{(i)} \parallel (-1)^i ({}^O\hat{s}_1 \times {}^O\hat{s}_3), i = 1, 2,$$

$${}^O\mathbf{r}_j = {}^O\mathbf{r}_{S/O} = (0, 0, d_1)^\top, j = 1, 2, 3,$$

Once the screw axes are known, we compute the **joint angles** by comparing how these axes rotate away from a known home configuration ($q_i=0$). Let ${}^O\hat{s}_{i,0}$ be the directions of the axes at home configuration, then, from Fig. 1:

$$[{}^O\hat{s}_{1,0}, \dots, {}^O\hat{s}_{7,0}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & -1 \end{bmatrix}$$

With this information, we can compute the exponential map in $\text{SO}(3)$ to align each axis step by step. This process is shown in Algorithm 1. Note that line 4 is a simplification of $q_j \leftarrow |\log({}_{j+1}^O \mathbf{R}_j^\top \mathbf{R}_j)|$.

In the Secs. III, IV, V and VI we present solvers that have, respectively, q_7, q_6, q_4 and swivel angle as free variable. Each solver has as input the value of the free variable, and ${}^O\hat{i}_E, {}^O\hat{s}_7 = {}^O\hat{\mathbf{k}}_E$, and ${}^O\mathbf{r}_{O_7/O} = {}^O\mathbf{r}_{E/O} - d_E {}^O\hat{\mathbf{k}}_E$ (and ${}^O\mathbf{r}_{O_7/S}$). The goal is to find ${}^O\mathbf{S}_4, {}^O\mathbf{S}_5$ and ${}^O\mathbf{S}_6$.

III. SOLUTION LOCKING JOINT 7

It can be seen that ${}^O\hat{s}_6 = \exp((\frac{3}{4}\pi - q_7) {}^O\hat{s}_7) {}^O\hat{i}_E$ and ${}^O\mathbf{r}_6 = {}^O\mathbf{r}_{O_6/O} = {}^O\mathbf{r}_{O_7/O} - a_7 {}^O\hat{s}_6 \times {}^O\hat{s}_7$.

Consider ${}^O\mathbf{r}_{O_6/S} = {}^O\mathbf{r}_{O_6/O} - {}^O\mathbf{r}_{S/O}$ and let $l := |{}^O\mathbf{r}_{O_6/S}|$, $b_1 := \sqrt{d_3^2 + a_4^2}$, $b_2 := \sqrt{d_5^2 + a_5^2}$, $\rho_1 := \arctan(a_4/d_3)$ and $\rho_2 := \arctan(a_5/d_5)$ in Fig. 2, then:

$$\alpha_2^{(1)} = \rho_2 + \beta_2, \quad \alpha_2^{(2)} = \rho_2 - \beta_2, \quad (1)$$

$$\text{with } \beta_2 := \arccos((b_1^2 - b_2^2 - l^2)/(-2b_2l)).$$

These two solutions correspond to the elbow-up and elbow-down assemblies of triangle SO_4O_6 . However, as noted in [22], due to link interference, solution 2 only happens when SO_4O_6 is almost flat and $d_3 + d_5 < l < b_1 + b_2$, Fig. 8

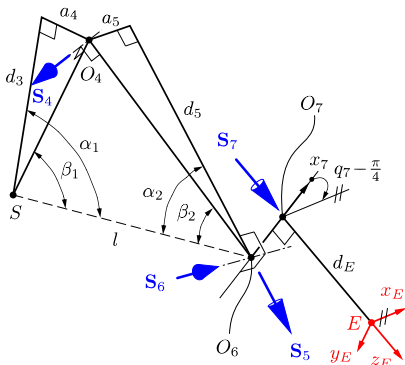


Fig. 2: Franka Robot Geometry for the IK solution fixing q_7 .

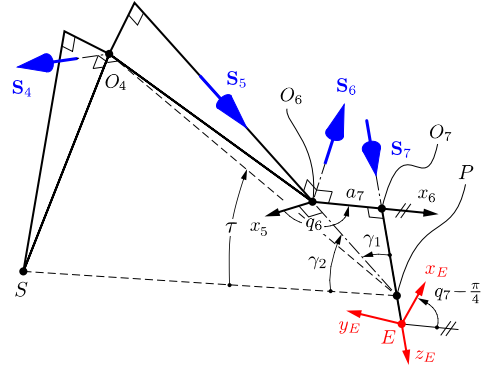


Fig. 3: Geometry of the Franka for the IK fixing $q_6 \notin \{0, \pi\}$.

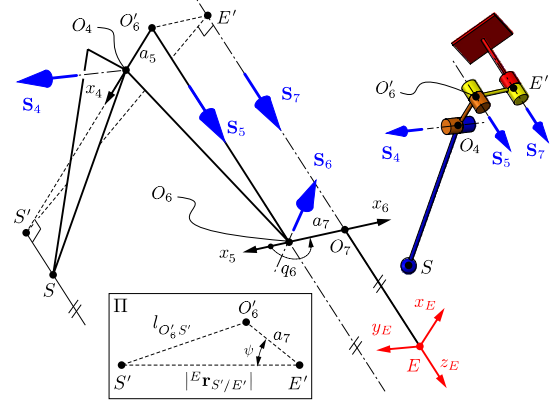


Fig. 4: Geometry of the Franka for the IK fixing $q_6 \in \{0, \pi\}$.

shows one of such scenarios.

Define a frame C with ${}^O\hat{\mathbf{k}}_C \parallel {}^O\mathbf{r}_{S/O_6}$ and ${}^O\hat{\mathbf{i}}_C \parallel {}^O\hat{\mathbf{k}}_C \times {}^O\hat{s}_6$. This allows us to easily write ${}^C\hat{s}_5 = -(\sin \alpha_2 \cos \gamma, \sin \alpha_2 \sin \gamma, \cos \alpha_2)^\top$, where, if ${}^C\hat{s}_6 = {}^C\mathbf{R} {}^O\hat{s}_6 = (0, s_{6,y}, s_{6,z})$, imposing the constraint ${}^C\hat{s}_5 \cdot {}^C\hat{s}_6 = 0$ results in $\gamma^{(1)} = \arcsin(A_\gamma), \gamma^{(2)} = \pi - \arcsin(A_\gamma)$,

where, $A_\gamma := -(s_{6,z}/s_{6,y}) \cot \alpha_2$. If $A_\gamma > 1$, the kinematic chain cannot be assembled, and if $A_\gamma = 1$ the two solutions for γ are repeated. Finally, ${}^O\hat{s}_5 = {}^C\mathbf{R} {}^C\hat{s}_5$ and ${}^O\mathbf{r}_5 = {}^O\mathbf{r}_6$. The remaining parameters are easily found as follows:

$${}^O\hat{s}_4 \parallel {}^O\hat{s}_5 \times {}^O\mathbf{r}_6, \quad {}^O\mathbf{r}_4 = {}^O\mathbf{r}_6 - d_5 {}^O\hat{s}_5 + a_5 {}^O\hat{s}_5 \times {}^O\hat{s}_4.$$

Since α_2, γ and ${}^O\hat{s}_2$ can have up to two solutions each, the maximum number of solutions is $2^3 = 8$.

IV. SOLUTION LOCKING JOINT 6

Define $\gamma_1 := \pi - q_6$ as the angle between \mathbf{S}_5 and \mathbf{S}_7 , and let P be their intersection point as shown in Fig. 3. The following two cases are analysed separately.

\mathbf{S}_5 and \mathbf{S}_7 are not parallel: It can be seen that ${}^O\mathbf{r}_{P/S} = {}^O\mathbf{r}_{O_7/S} + (a_7/\tan \gamma_1) {}^O\hat{\mathbf{k}}_E$. Let γ_2 be the angle from $\mathbf{r}_{S/P}$ to $-\mathbf{S}_5$ measured about $-\hat{s}_4$ as shown in Fig. 3. The goal is to first find γ_2 using the polygon SO_4O_6P and then use the result to find \hat{s}_5 .

Let l_C be the projection of \mathbf{r}_{P/O_6} on \mathbf{S}_5 , so $l_C = a_7/\sin \gamma_1$. The angle τ , defined as $\angle SPO_4$ can be easily found by solving first the triangle PO_6O_4 followed by triangle SPO_4 . It can be seen that

$$\gamma_2^{(1)} = \begin{cases} \tau + \arctan(a_5/(d_5 + l_C)) + \pi & \text{if } l_C < -d_5, \\ \tau + \arctan(a_5/(d_5 + l_C)) & \text{otherwise} \end{cases}$$

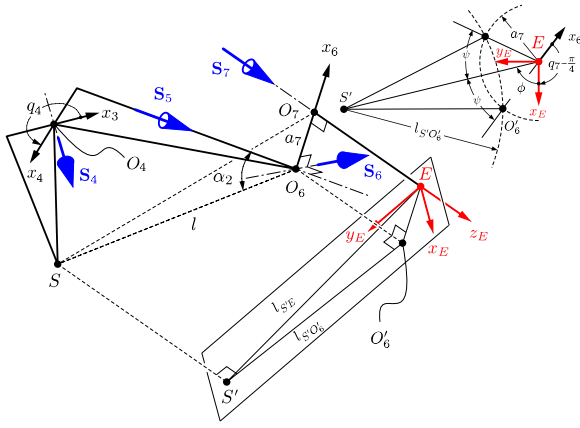


Fig. 5: Geometry of the Franka for the IK solution fixing q_4 .

The extreme case when SPO_4 is almost flat, i.e. $d_3 + d_5 + l_C < |{}^O\mathbf{r}_{S/P}| < b_1 + (a_5^2 + (l_C + d_5)^2)^{1/2}$, provides a second solution $\gamma_2^{(2)}$ in which τ appears with negative sign.

Now we can write an equation to constrain the angle between $\mathbf{r}_{S/P}$ and \mathbf{S}_5 to γ_2 . In frame E it is easy to write ${}^E\hat{\mathbf{s}}_5 = (-\sin \gamma_1 \cos u, -\sin \gamma_1 \sin u, \cos \gamma_1)^\top$, where $u := 5\pi/4 - q_7$. Then, the following equation in u is to be solved:

$$-{}^E\hat{\mathbf{s}}_5 \cdot {}^E\mathbf{r}_{S/P} = {}^E\hat{\mathbf{s}}_5 \cdot ({}^O\mathbf{R})^\top {}^O\mathbf{r}_{P/S} = |{}^O\mathbf{r}_{P/S}| \cos \gamma_2$$

which has the two solutions $u^{(1)} = u_1 - u_2$ and $u^{(2)} = \pi - u_1 - u_2$, where, if ${}^E\mathbf{r}_{S/P} = (C_1, C_2, C_3)^\top$:

$$u_1 := \arcsin \left(\frac{|{}^O\mathbf{r}_{P/S}| \cos \gamma_2 + C_3 \cos \gamma_1}{|\sin \gamma_1| \sqrt{C_1^2 + C_2^2}} \right),$$

$$u_2 := \operatorname{atan2}(C_1 \sin \gamma_1, C_2 \sin \gamma_1)$$

With ${}^O\mathbf{r}_{P/O} = {}^O\mathbf{r}_{P/S} + {}^O\mathbf{r}_{S/O}$, the remaining parameters are found easily:

$${}^O\mathbf{r}_5 = {}^O\mathbf{r}_6 = {}^O\mathbf{r}_{O_6/O} = {}^O\mathbf{r}_{P/O} - l_C {}^O\hat{\mathbf{s}}_5,$$

$${}^O\hat{\mathbf{s}}_6 \parallel {}^O\hat{\mathbf{k}}_E \times ({}^O\mathbf{r}_{O_7/O} - {}^O\mathbf{r}_{O_6/O}), \quad {}^O\hat{\mathbf{s}}_4 \parallel {}^O\hat{\mathbf{s}}_5 \times {}^O\mathbf{r}_{P/S},$$

$${}^O\mathbf{r}_4 = {}^O\mathbf{r}_{O_4/O} = {}^O\mathbf{r}_{P/O} - (d_5 + l_C) {}^O\hat{\mathbf{s}}_5 + a_5 {}^O\hat{\mathbf{s}}_5 \times {}^O\hat{\mathbf{s}}_4$$

Since γ_2 , u and ${}^O\hat{\mathbf{s}}_2$ have up to two solutions, the maximum number of solutions is $(2)^3 = 8$.

\mathbf{S}_5 and \mathbf{S}_7 are parallel: Since $q_6 \in \{0, \pi\}$, we define $\sigma = \cos q_6 \in \{1, -1\}$. Consider the plane Π which is perpendicular to \mathbf{S}_5 and \mathbf{S}_7 and contains \mathbf{S}_4 , see Fig. 4. Let E' , O'_6 and S' be the respective projections of E , O_6 and S onto Π . Note that ${}^E\mathbf{r}_{E'/E} = (0, 0, -d_E + \sigma d_5)^\top$. Since E' is known, the problem is reduced to solving the equivalent kinematic chain shown in Fig. 4 by working on plane Π and in frame E .

It can be seen that ${}^E\mathbf{r}_{S'/E'} = {}^E\mathbf{r}_{S/E'} - ({}^E\mathbf{r}_{S/E'} \cdot {}^E\hat{\mathbf{k}}_E) {}^E\hat{\mathbf{k}}_E$, where ${}^E\mathbf{r}_{S/E'} = {}^E\mathbf{r}_{S/E} - {}^E\mathbf{r}_{E'/E}$. Due to the two possible ways of reaching S from Π , the distance from O'_6 to S' along x_4 is $l_{O'_6 S'}^{(i)} = a_5 + (-1)^i (d_3^2 + a_4^2 - ({}^E\mathbf{r}_{S'/E'} \cdot {}^E\hat{\mathbf{k}}_E)^2)^{1/2}$, $i = 1, 2$. This leads to two solutions for ψ , the angle $\angle S'E'O'_6$:

$$\psi^{(i)} = \arccos \left[\frac{(l_{O'_6 S'}^{(i)})^2 - a_7^2 - |{}^E\mathbf{r}_{S'/E'}|^2}{-2a_7 |{}^E\mathbf{r}_{S'/E'}|} \right], \quad i = 1, 2.$$

Similarly, due to the two different ways of assembling the triangle $S'E'O'_6$, two solutions for the position of O'_6 appear

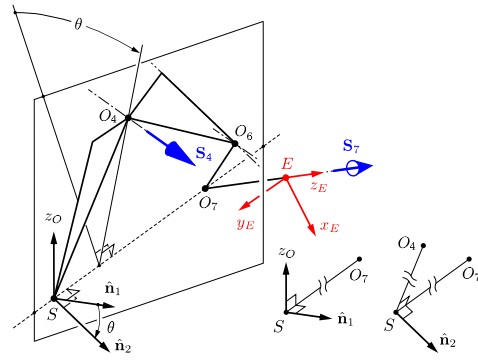


Fig. 6: Geometry of the Franka arm for the IK solution fixing the swivel angle θ .

for each solution of ψ :

$${}^E\mathbf{r}_{O'_6/E'}^{(i)} = a_7 \exp((-1)^i \psi) {}^E\hat{\mathbf{k}}_E {}^E\mathbf{r}_{S'/E'} / |{}^E\mathbf{r}_{S'/E'}|, \quad i = 1, 2.$$

It follows that ${}^E\mathbf{r}_6 = {}^E\mathbf{r}_{E'/E} + {}^E\mathbf{r}_{O'_6/E'} - \sigma d_5 {}^E\hat{\mathbf{k}}_E$ and ${}^E\hat{\mathbf{s}}_6 \parallel {}^E\mathbf{r}_{O'_6/E'} \times {}^E\hat{\mathbf{k}}_E$. For ${}^E\mathbf{S}_5$, it can be seen that ${}^E\hat{\mathbf{s}}_5 = -\sigma {}^E\hat{\mathbf{k}}_E$ and ${}^E\mathbf{r}_5 = {}^E\mathbf{r}_6$. To find ${}^E\mathbf{S}_4$, consider ${}^E\hat{\mathbf{i}}_4 \parallel \operatorname{sign}(l_{O'_6 S'}) ({}^E\mathbf{r}_{S'/E'} - {}^E\mathbf{r}_{O'_6/E'})$, then ${}^E\mathbf{r}_4 = {}^E\mathbf{r}_{E'/E} + {}^E\mathbf{r}_{O'_6/E'} + a_5 {}^E\hat{\mathbf{i}}_4$, and ${}^E\hat{\mathbf{s}}_4 = {}^E\hat{\mathbf{i}}_4 \times {}^E\hat{\mathbf{s}}_5$. All the screw axes are then transformed back to frame O .

Since ψ , ${}^E\mathbf{r}_{O'_6/E'}$ and ${}^O\hat{\mathbf{s}}_2$ can have up to two solutions each, the maximum number of solutions is $2^3 = 8$.

V. SOLUTION LOCKING JOINT 4

Since in this case q_4 is given, the triangle SO_4O_6 can be easily solved to obtain $l := |{}^O\mathbf{r}_{S/O_6}|$ and α_2 , see Fig. 5. We can then locate O_6 on a sphere of radius l and centre at S . From the pose of frame E , we can also constrain O_6 to a known circle of radius a_7 and centre at O_7 . Hence, the problem is solved by finding the intersection of this circle with the sphere. This is reduced to the intersection of two circles on the $X_E Y_E$ -plane.

As shown in Fig. 5, let S' and O'_6 be the respective projections of points S and O_6 onto the $X_E Y_E$ -plane, then define $l_{S'O'_6} := |{}^E\mathbf{r}_{S'/O'_6}| = (l^2 - ({}^E\mathbf{r}_{O_7/O} \cdot {}^E\hat{\mathbf{k}}_E)^2)^{1/2}$ and $l_{S'E} := |{}^E\mathbf{r}_{S'/E}| = (({}^E\mathbf{r}_{O_7/O} \cdot {}^E\hat{\mathbf{i}}_E)^2 + ({}^E\mathbf{r}_{O_7/O} \cdot {}^E\hat{\mathbf{j}}_E)^2)^{1/2}$.

To find O'_6 , consider ϕ , the angle from x_E to $\mathbf{r}_{S'/E}$ about z_E , and ψ , the angle $\angle S'E'O'_6$, as shown in Fig. 5. It can be seen that $\phi = \operatorname{atan2}({}^E\mathbf{r}_{S'/O_7} \cdot {}^E\hat{\mathbf{j}}_E, {}^E\mathbf{r}_{S'/O_7} \cdot {}^E\hat{\mathbf{i}}_E)$ and $\psi = \arccos((l_{S'E}^2 + a_7^2 - l_{S'O'_6}^2) / (2a_7 l_{S'E}))$, leading to the two solutions $q_7^{(i)} = (-1)^i \psi - \phi - 3\pi/4$, $i = 1, 2$.

With O'_6 in hand, it is easy to find ${}^O\mathbf{S}_6$ with ${}^O\hat{\mathbf{s}}_6 = {}^O\mathbf{R} \exp((-q_7 + 3\pi/4) {}^E\hat{\mathbf{k}}_E) {}^E\hat{\mathbf{i}}_E$ and ${}^O\mathbf{r}_{O_6/O} = {}^O\mathbf{R} {}^E\mathbf{r}_{O_6/O_7} + {}^O\mathbf{r}_{O_7/O}$, where ${}^E\mathbf{r}_{O_6/O_7} = -a_7 \exp((-q_7 + \pi/4) {}^E\hat{\mathbf{k}}_E) {}^E\hat{\mathbf{i}}_E$.

Since q_7 , ${}^O\mathbf{S}_6$ and α_2 are known now, we can proceed as we did in Sec. III, where q_7 was a free parameter. Following the same procedure, we find two solutions of ${}^O\hat{\mathbf{s}}_5$ per each value of q_7 . Since q_7 , ${}^O\hat{\mathbf{s}}_5$ and ${}^O\hat{\mathbf{s}}_2$ can have up to two solutions each, the total number of solutions is $2^3 = 8$.

VI. SOLUTION FOR SWIVEL ANGLE

Due to the lack of a spherical wrist, the swivel angle θ can be defined using either O_6 or O_7 as “wrist centre”. Let this be O_7 , like in [32]. Then the swivel angle is defined between the plane containing z_O and O_7 , and the plane containing

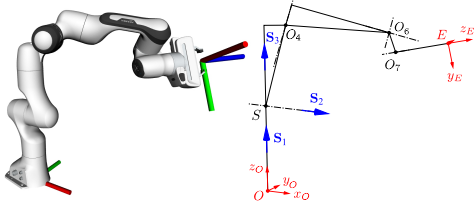


Fig. 7: An example of flat shoulder singularity.

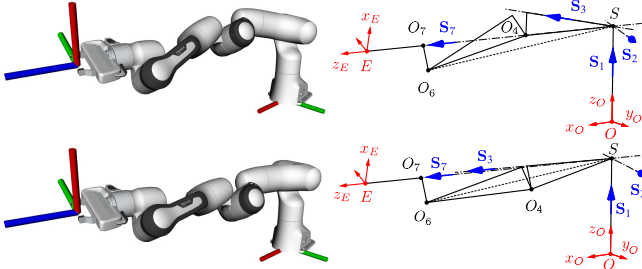


Fig. 8: Elbow-up and elbow-down solutions in a Shoulder-on- S_7 singularity. Note that these are *not* work-space boundary singularities as the arm is not fully stretched.

O_4 , S and O_7 as shown in Fig. 6. Hence, if:

$${}^O\hat{\mathbf{n}}_1 \parallel {}^O\mathbf{r}_{O_7/S} \times {}^O\hat{\mathbf{k}}_O \text{ and } {}^O\hat{\mathbf{n}}_2 \parallel \sigma {}^O\mathbf{r}_{O_7/S} \times {}^O\mathbf{r}_{O_4/S} \quad (2)$$

$$\text{with } \sigma = \text{sign}({}^O\mathbf{r}_{O_7/S} \times {}^O\mathbf{r}_{O_4/S} \cdot {}^O\hat{\mathbf{s}}_4) \quad (3)$$

then θ is the angle from ${}^O\hat{\mathbf{n}}_1$ to ${}^O\hat{\mathbf{n}}_2$ measured about ${}^O\mathbf{r}_{O_7/S}$.

In order to find all the solutions that have the desired value of θ , we discretise the range of q_7 , calculate θ for each point, pick the ones that are close enough to the desired value, and finally we interpolate to find the optimal values of q_7 . q_7 is used as free variable since, by doing so, the first singularity discussed in Sec. VII is avoided.

We do not need to solve the entire IK with q_7 as input, we only obtain the strictly necessary geometric elements that allow us to calculate θ . That means proceeding as done in Sec. III until ${}^O\mathbf{r}_7$, ${}^O\mathbf{r}_6$ and ${}^O\hat{\mathbf{s}}_4$ are found. To speed up the process, we ignore the elbow-down solution with $\alpha_2^{(2)}$ in Eq. (1), which appears in the unlikely cases where $d_3 + d_5 < l < b_1 + b_2$.

VII. SINGULARITY ANALYSIS

Singularities in 7-DOF robots can be classified as kinematic and algorithmic singularities [33]. *Kinematic singularities* are configurations in which $\text{rank}(\mathbf{J}(\mathbf{q})) < 6$. This rank-deficiency are the result of special geometric conditions between the screw axes. For our IK solvers, these conditions will appear as extreme cases of triangles becoming flat or axes becoming parallel. To provide numerical stability in those cases, GeoFIK includes a constant `SING_TOL` which defines how close to the extreme value a variable can be to still attempt a solution, e.g. if $x > 1$ but $x - 1 < \text{SING_TOL}$, $\arccos x = 0$ is considered. An example occurs when \mathbf{S}_4 and \mathbf{S}_6 are parallel and coplanar with S . As a consequence, \mathbf{S}_5 , \mathbf{S}_6 and S are coplanar too, resulting in $\text{rank}(\mathbf{J}(\mathbf{q})) = 5$ and the triangle SO_4O_6 becoming flat. It can be seen that work-space boundary configurations are kinematic singularities. In general, since the Franka has 7 joints, two special geometric conditions are needed to take $\text{rank}(\mathbf{J})$ below 6.

In *algorithmic singularities* the self-motion no longer affects the variable we want to fix to resolve the redundancy. If q_i is being fixed, $\text{rank}(\mathbf{J}(\mathbf{q})) = 6$ can hold, but if \mathbf{S}_i is

linearly independent from the other screw axes, then it is not possible to have q_i as free variable. In the Franka arm with the redundant variables considered in this paper, such cases can only occur in the following scenario:

Shoulder-on- S_7 singularity: If \mathbf{S}_7 intersects the spherical shoulder at S , the robot experiences self-motion with the polygon $SO_4O_6O_7$ spinning around \mathbf{S}_7 without affecting q_4 , q_5 and q_6 . Hence, only q_7 can be used as free variable with the resulting 6-DOF robot being in regular configuration. If GeoFIK detects that $S \in \mathbf{S}_7$, the solver with q_7 as free variable is called with an emergency value of q_7 provided by the user. Fig. 8 shows an example of this singularity.

Theoretically, other algorithmic singularities can appear whenever a pair of joint axes become collinear or a joint axis intersects S . However, due to the link-lengths and joint limits of the Franka, this will only occur in the two scenarios described above. As noted in [10], the home configuration (Fig. 1) is a kinematic singularity with $\text{rank}(\mathbf{J}(\mathbf{q})) = 5$, but it is also algorithmic-singular as \mathbf{S}_5 intersects S . This case is not implemented in GeoFIK since it violates joint limits.

As reported in [19], [22], if $q_2 = 0$ the self-motion is reduced to \mathbf{S}_2 spinning about $\mathbf{S}_1 = \mathbf{S}_3$ (Fig. 7). However, this configuration is not a kinematic singularity since, in general, $\text{rank}(\mathbf{J}) = 6$. The robot is neither in an algorithmic singularity as we ignore the first 3 joints to obtain the rest of the joint axes. However, the shoulder is in a singularity since $\text{rank}([\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3]) = 2$. Hence, we refer to this configuration as **flat shoulder singularity**. Only the definition of \mathbf{S}_2 is affected, with q_1 becoming a free choice. The user can provide an emergency value of q_1 to be used in this situation.

Another kind of singularity occurs in the swivel angle definition (2). Clearly, θ is not defined if $\mathbf{r}_{O_7/S}$ is parallel to z_O . This is the unavoidable singularity of any swivel angle definition [34]. GeoFIK prints an error message in those cases. Note that the sign σ prevents the sudden change of direction of the plane containing S , O_7 and O_4 . In addition, if these three points are collinear, $\hat{\mathbf{n}}_2 = \hat{\mathbf{s}}_4$ is chosen.

VIII. COMPARISON WITH BASELINE SOLVERS

This section presents an analytical comparison against the state-of-the-art solvers from He and Liu [22] (referred to here as HeLiu), IKFast [24], and the 6-DOF solver IKGeo [26]. HeLiu is only compared fixing q_7 since it is the only solver they provide. For IKGeo, we coded the necessary functions to solve the IK of the Franka using the general 6-DOF solver. The different solvers provided by IKGeo were tested and it was decided to use `IK_spherical_2_intersect` when locking q_7 , and `IK_spherical` for q_6 and q_4 .

A. Number of solutions attempted

As discussed in Sections III-V, each case of GeoFIK solver attempts 8 solutions which constitute all geometrically valid solutions. In contrast, HeLiu attempts only 4 solutions, explicitly omitting a secondary elbow-down (solution 2 in Eq. (1)), as acknowledged in [22]. Due to its general algebraic approach, IKFast attempts the 16 solutions that the system of equations can have. However, since only 8 solutions are geometrically possible, the other 8 must be complex or repeated. Among those, IKFast will return elbow-down solutions with link interference which have to be discarded by subsequent joint-limits checks. The number of solutions attempted by IKGeo depends on each subproblem it solves.

TABLE I: Solver Comparison for Singular Configurations (angles in degrees). Solutions in orange indicate larger errors (EE position) ≥ 1 cm, and in red, violations of joint limits.

Case 1 (Flat shoulder, solvers with q_7 as free variable)								
sol	q_1	q_2	q_3	q_4	q_5	q_6	q_7	
GeoFIK (ours)	1	-30.06	86.09	123.97	-106.86	48.58	nan	-21.32
	2	149.94	-86.09	-56.03	-106.86	48.58	nan	-21.32
	3	-90	0	115.96	-106.86	131.42	150.52	-21.32
	4	90	0	-64.04	-106.86	131.42	150.52	-21.32
HeLiu	1	90	0	-64.04	-106.86	131.42	150.52	-21.32
	2	90	0	-64.04	-106.86	131.42	150.52	-21.32
	3	-30.06	86.09	123.97	nan	nan	nan	nan
	4	149.94	-86.09	-56.03	nan	nan	nan	nan
IKFast	1	-30.06	86.09	123.97	-106.86	48.58	-137.3	-21.32
	2	149.94	-86.09	-56.03	-106.86	48.58	-137.3	-21.32
IKGeo	1	149.93	-86.09	-56.03	-106.9	48.58	-137.3	-21.32
	2	-30.06	86.09	123.97	-106.9	48.58	-137.3	-21.32
	3	-27.9	-0.02	61.7559	-106.9	131.4	150.5	-21.32
	4	152.1	0.02	-118.2	-106.9	131.42	150.5	-21.32
	5	-174.8	-48.83	65.06	53.35	-90	-173.4	-21.32
	6	5.17	48.83	-114.9	53.35	-90	-173.4	-21.32
Case 2 (algorithmic singularity, solvers with q_6 as free variable)								
sol	q_1	q_2	q_3	q_4	q_5	q_6	q_7	
GeoFIK (ours)	1	-18.99	78.13	28.75	-39.33	0	204.90	0
	2	161.01	-78.13	-151.25	-39.33	0	204.90	0
	3	2.37	nan	-148.49	-39.33	nan	169.24	0
	4	nan	nan	31.51	-39.33	nan	169.24	0
	5	-12.43	90.22	28.08	-14.19	0	193.44	0
	6	nan	-90.22	-151.92	-14.19	0	193.44	0
	7	-4.97	nan	-151	-14.19	nan	180.65	0
	8	nan	nan	29	-14.19	nan	180.65	0
IKFast	no solution							
IKGeo	no solution							

B. Handling of singularities

We examine the behavior of the solvers in the flat shoulder and wrist-on- S_7 configurations described in Sec. VII.

flat shoulder singularity: Consider the following IK problem that leads to a flat shoulder:

$${}^O_E \mathbf{R} = \begin{pmatrix} 0.6688331 & 0.31705344 & 0.672413 \\ -0.6398146 & -0.21507724 & 0.7378205 \\ 0.3785493 & -0.92369843 & 0.0590046 \end{pmatrix},$$

$${}^O_{\mathbf{r}_{E/O}} = (0.61674948, 0.32278029, 0.56790512)^\top,$$

$$q_7 = -21.32455095^\circ.$$

The problem is sent to the solvers with q_7 as a free parameter. The results are shown as Case 1 in Table I.

In GeoFIK, the user can provide a value of q_1 to resolve the redundancy. The default value is $\pi/2$. Solutions for the two ways of assembling the spherical shoulder are still returned with $\pm q_1$. One such solution is shown in Fig. 7. Other two solutions that are not singular are returned, however, in those q_6 violates joint limits. In HeLiu, the user provides a vector of joint angles from which it takes q_1 and uses it to resolve the redundancy induced by the singularity. Then, it returns two repeated solutions with q_1 , rather than two different ones with $\pm q_1$, which can lead to discontinuity in the real-world applications, and the other two regular solutions that violate joint limits. IKFast fails in this singularity and only returns the two regular configurations that violate joint limits. IKGeo returns these same two regular configurations but also returns 4 invalid solutions for which the end-effector position is more than 1cm away from that of the requested pose.

shoulder-on- S_7 : Consider the following IK problem that leads to an algorithmic singularity:

$${}^O_E \mathbf{R} = \begin{pmatrix} 0.0746454 & -0.1964604 & 0.9776662 \\ 0.281646 & -0.93633263 & -0.2096583 \\ 0.9566105 & 0.2910058 & -0.0145606 \end{pmatrix},$$

$${}^O_{\mathbf{r}_{E/O}} = (0.89948341, -0.1928922, 0.31960372)^\top,$$

$$q_6 = 193.48937052^\circ.$$

The problem is sent to the solvers with q_6 as a free parameter.

The results are shown as Case 2 in Table I. In the algorithmic singularities, q_6 stops being a free variable as there are only two values of q_6 that satisfy the pose. One of them is the value used in this IK problem.

GeoFIK is able to automatically detect that q_6 can no longer be freely chosen and transfers the problem to the solver with q_7 as a free variable. 8 solutions are found, indicating that triangle SO_4O_6 is almost flat, allowing elbow-up and elbow-down solutions as shown in Fig. 8. Among those, two solutions are within the locked value for q_6 .

Both IKFast and IKGeo fail to generate any solution. It is worth highlighting that this behaviour is inconsistent in IKFast as it is able to find solutions to some shoulder-on- S_7 singularities (yet, it is highly sensitive to numerical precision to work). This is not an issue with GeoFIK, since once it detects the singularity, it no longer requires the value of q_6 . Hence, it returns consistently valid solutions.

C. Quantitative Analysis

While Sec. VIII-B provided a comparison in singular configurations, here we quantitatively evaluate the performance in regular and near-singular configurations. We generated 2000 random valid joint configurations to produce EE poses, which in turn were sent as input to the solvers⁶.

We evaluated all the solvers against 4 parameters. The results are shown in Table II.

(i) Computational time for a single call for only joint angles, only Jacobians, and both joint angles and Jacobians. For only joint angles, GeoFIK outperformed the other solvers when locking q_6 and q_4 . HeLiu is the fastest in the case of q_7 , this is due to the lack of calculation of elbow-down solutions.

When comparing the Jacobian computation time, GeoFIK outperformed all other solvers. This is mostly due to our screw-axis-based design that returns the Jacobian as a byproduct of the IK. In fact, GeoFIK outputs Jacobians even faster than the joint-configurations.

(ii) Number of distinct valid solutions (total and per call): We consider a solution valid if it is within joint-limits and has an EE error smaller than 5mm in position and 3° in rotation. Repeated solutions are not counted. It can be seen that GeoFIK outperforms all the solvers in total number of solutions with IKFast almost providing the same number.

(iii) Total number of incorrect solutions: Incorrect solutions are defined here as being within joint-limits but having an EE error as described in (ii). HeLiu and IKGeo returned incorrect solutions in the q_7 case.

(iv) EE error: GeoFIK outperforms all the solvers in EE error with exception of the q_7 case.

IX. REAL-WORLD EXPERIMENT

We evaluated our GeoFIK approach on a real Franka Robot to investigate (i) feasibility and joint-space tracking of a desired task-space trajectory (herein obtained through vector fields) within the robot's central workspace, and (ii) whole-body tasks that require both the end-effector pose and the manipulability ellipsoid to match a human-demonstrated reference (i.e., sweeping and pushing).

Feasibility and Trajectory Tracking: This first scenario tested the ability to compute real-time IK solutions and

⁶Laptop used: 11th Gen Intel Core i7-11370H (3.30 GHz), 32GB running Windows 11. Compiler: gcc 13.2.0 and optimization flag $-O3$ for all solvers. For the case of locking q_7 , clang was used instead for all solvers as it gave better results, except for IKFast which was compiled with gcc.

TABLE II: IK Solver Quantitative Comparison in terms of computation time, number of solutions, and errors.

Min/Max Mean \pm Std	Locking q_7				Locking q_6			Locking q_4			Locking S. Angle
	GeoFIK	HeLiu	IKFast	IKGeo	GeoFIK	IKFast	IKGeo	GeoFIK	IKFast	IKGeo	GeoFIK
Joint angles computation time (μ s)	1.7/4.8 2.1 \pm 0.6	0.3/16.6 1.0 \pm 0.6	3.2/14.6 4.8 \pm 1.7	4.5/25.0 6.0 \pm 2.3	2.5/6.3 3.1 \pm 0.9	33.7/129.3 39.1 \pm 7.8	7.7/30.1 8.3 \pm 1.3	5.1/7.4 5.4 \pm 0.2	16.6/71.7 18.7 \pm 5.6	5.5/23.4 6.1 \pm 2.2	41.7/1429.2 198 \pm 123
Jacobian only computation time (μ s)	0.2/1.9 0.3 \pm 0.07	—	—	—	0.4/2.3 0.8 \pm 0.3	—	—	0.9/7.7 1.4 \pm 0.5	—	—	30.8/1405.5 192.1 \pm 125
Jacobian + joint angles computation time (μ s)	1.8/11.3 2.7 \pm 1.5	6/20 7 \pm 1.2	8.6/41 12.8 \pm 5	11.5/55.4 14.4 \pm 4.6	2.6/11.1 3.5 \pm 1.3	39.5/132.6 46.7 \pm 7.8	18/73.5 19.7 \pm 3	5.2/21 5.6 \pm 0.6	17/65.5 19 \pm 4.6	10.5/31.6 11.4 \pm 2.7	41.6/1322.8 204.2 \pm 135.3
Num. valid sols. per call	1/8 3.2 \pm 1.5	0/4 2.6 \pm 1.1	1/8 3.2 \pm 1.5	1/8 3.2 \pm 1.5	1/8 3.6 \pm 1.5	1/8 3.6 \pm 1.5	0/8 2.9 \pm 1.6	1/8 4.9 \pm 2.0	1/8 4.9 \pm 2.0	0/4 2.6 \pm 1.5	0/8 2.0 \pm 1.0
Total num. of valid sols.	6352	5272	6346	6350	7119	7119	5856	9742	9742	5137	4042
Total num. of wrong sols.	0	72	0	328	0	0	0	0	0	0	0
EE pos. error mean (m)	2 \times 10 ⁻⁶	3 \times 10 ⁻⁴	7 \times 10 ⁻¹⁰	5 \times 10 ⁻³	9 \times 10 ⁻¹⁴	7 \times 10 ⁻¹⁰	9 \times 10 ⁻⁸	6 \times 10 ⁻¹⁴	6 \times 10 ⁻¹⁰	5 \times 10 ⁻¹⁰	1 \times 10 ⁻⁵
EE rot. error mean (rads)	1 \times 10 ⁻⁵	4 \times 10 ⁻⁴	1 \times 10 ⁻⁹	3 \times 10 ⁻¹³	3 \times 10 ⁻¹³	1 \times 10 ⁻⁹	3 \times 10 ⁻¹³	3 \times 10 ⁻¹³	1 \times 10 ⁻⁹	3 \times 10 ⁻¹³	2 \times 10 ⁻¹³

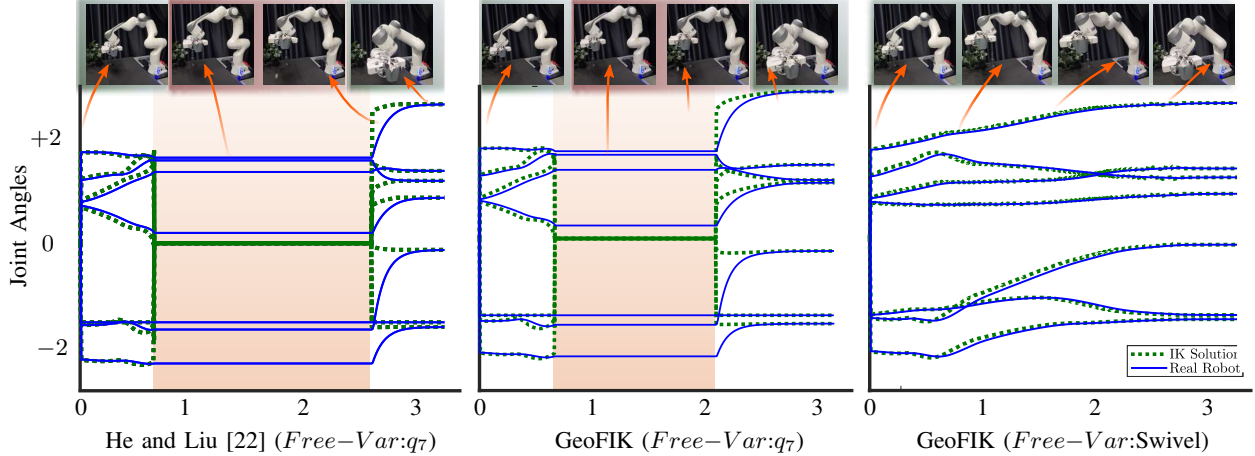


Fig. 9: Trajectory tracking using IK solvers: The obtained joints (dashed-green lines) compared to the real-robot (blue lines). HeLiu and GeoFIK (with q_7 as a free-var) both fail to find a feasible solution around 10 s. (shaded area) and the robot stops (real-robot on the top strip) only to continue with an aggressive motion at the end of the trajectory. GeoFIK with the swivel-angle as free-var always finds a solution—as expected from Table II—enabling smooth joint-trajectory.

maintain a continuous joint-space trajectory over a basic trajectory in the central region of the workspace, a domain expected to be straightforward. Despite this assumption, the IK solver by HeLiu failed after approximately 10 s, providing no valid solutions and forcing the robot to halt. Our GeoFIK solver, when using q_7 as the free parameter, also reached a point where it could not find a solution. Both failures led to abrupt reconfiguration once a valid solution was found again (Fig. 9 top-images with a red border).

Allowing the standard q_7 -locking solvers to sample a small range $\pm 10^\circ$ around the current q_7 mitigated total failure, but induced frequent chattering in joint angles (Fig. 11). Meanwhile, IKFast remained unable to find even a valid initial solution, all valid IKs demand an instantaneous flip of the base joints, generating torque reflexes and halting progress altogether, as shown in Fig. 12(a). In contrast, the swivel-angle-based GeoFIK solution never lost feasibility and preserved a smooth trajectory as shown in Fig 9.

Whole-Body (Manipulability) Tracking: The second set of experiments required each solver to produce both joint angles and Jacobian, which was used to compute the manipulability ellipsoid. The solver then aimed to match a reference ellipsoid provided by a user demonstration (e.g. sweeping or pushing motions). We sampled 250 configurations every 30ms, akin to a lightweight model-predictive control loop.

As shown in Fig. 10, GeoFIK achieves lower computational times thanks to its screw-transformation design, where the

Jacobian emerges naturally from the IK solution. Although HeLiu managed to complete the sweeping task, it proved slower in delivering joint-Jacobian pairs. IKFast again frequently produced no valid solutions (or an out-of-reach configuration), leading to task failure in the sweeping experiments, Figs. 12(b-c). GeoFIK, in contrast, consistently offered feasible joint trajectories while preserving the desired manipulability profile, underscoring the benefits of its geometry-aware method and multi-parameter redundancy resolution.

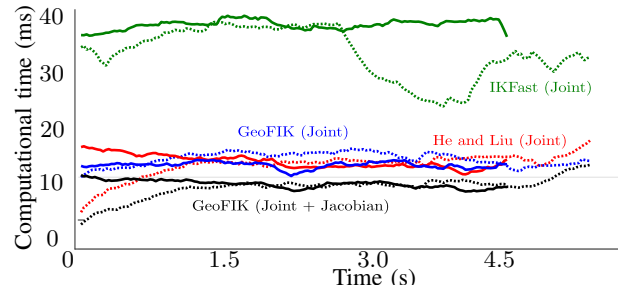


Fig. 10: Computation time in tracking manipulability for sweeping (dashed) and pushing (solid) tasks.

X. CONCLUSIONS

We introduced a fully geometric, screw-theory-based IK solver for the Franka Robot, accommodating multiple free parameters and automatically providing the Jacobian even

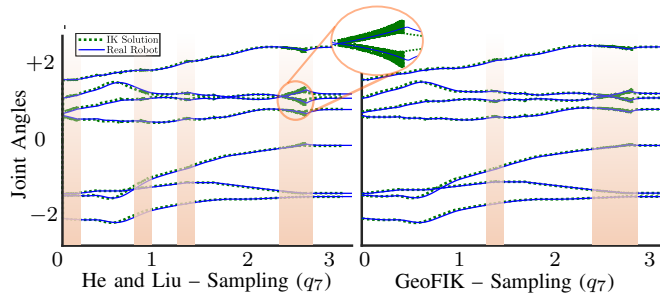
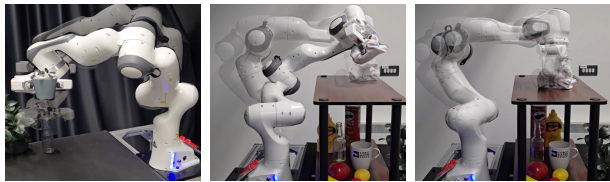


Fig. 11: Trajectory tracking using IK solvers and sampling $q_7 \pm 10^\circ$. Sampling allows q_7 -locking IKs to complete the task, but the resulting trajectory includes chattering due to discontinuities (the overlapped green areas).



(a) IKFast EE Track (b) IKFast Sweep (c) GeoFIK Sweep

Fig. 12: (a) IKFast traj. tracking, and (b)c IKFast and (c) GeoFIK sweeping tasks. IKFast often provides invalid solutions. In (a), the lack of solutions near the $\mathbf{q}(0)$ results in abrupt arm rotations exceeding torque reflexes. Similarly, in sweeping tasks, IKFast causes the arm to flip over lacking proper solutions. GeoFIK success in all tasks.

faster than the joint configuration. By systematically capturing all geometrically valid solutions, our method, GeoFIK, overcomes the limitations of existing solvers that either fix only q_7 or neglect key geometric insights. Extensive quantitative benchmarks showed that our approach consistently outperforms baseline solvers in terms of reliability, completeness, and efficiency. Real-world experiments confirmed its resilience, enabling the robot to smoothly execute tasks and even when controlling less conventional variables, such as the swivel angle. We believe that the flexibility and robustness of GeoFIK can facilitate more advanced planning, control, and learning algorithms, thereby fulfilling the growing demand for speed and reliability in modern robotics.

REFERENCES

- [1] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robotic manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.
- [2] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *IEEE International Conf. on Robotics and Automation (ICRA)*, 2021, pp. 9995–10001.
- [3] R. Laha, W. Wu, R. Sun, N. Mansfield, L. F. Figueredo, and S. Haddadin, "S*: On safe and time efficient robot motion planning," in *IEEE Int. Conf. on Robotics & Autom. (ICRA)*, 2023, pp. 12758–12764.
- [4] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.
- [5] L. F. Figueredo, R. D. C. Aguiar, L. Chen, T. C. Richards, S. Chakrabarty, and M. Dogar, "Planning to minimize the human muscular effort during forceful human-robot collaboration," *ACM Trans. on Human-Robot Interaction (THRI)*, vol. 11, no. 1, pp. 1–27, 2021.
- [6] M. Raghavan and B. Roth, "Inverse kinematics of the general 6R manipulator and related linkages," *Journal of Mechanical Design*, vol. 115, no. 3, pp. 502–508, 09 1993.
- [7] J. Burdick and H. Seraji, "Characterization and control of self-motions in redundant manipulators," in *Proceedings of the NASA Conference on Space Telecommunications, Volume 2*, 1989.

- [8] S. Haddadin, "The Franka Emika robot: A standard platform in robotics research [survey]," *IEEE Robotics and Automation Magazine*, 2024.
- [9] Q. Gallouédec, N. Cazin, E. Dellandréa, and L. Chen, "panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning," in *Proc. of the 4th Robot Learning Workshop: Self-Supervised and Lifelong Learning (NeurIPS)*, 2021.
- [10] P. Corke, "Matlab Robotics Toolbox: Franka Emika Panda kinematics & singularities," <https://petercorke.com/robotics/franka-emika-panda-kinematics-and-singularities/>.
- [11] A. A. Oliva, F. Spindler, P. R. Giordano, and F. Chaumette, "FrankaSim: A dynamic simulator for the Franka Emika robot with visual-servoing enabled capabilities," in *17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 868–874.
- [12] J. Elsner, "Taming the Panda with Python: A powerful duo for seamless robotics programming and integration," *SoftwareX*, vol. 24, 2023.
- [13] S. Feng, B. Burchfiel, and R. Tedrake, "TRI's robots learn new manipulation skills in an afternoon. Here's how," Medium (Toyota Research Institute), <https://medium.com/toyotaresearch/tris-robots-learn-new-skills-in-an-afternoon-here-s-how-2c30b1a8c573>.
- [14] C. Lenz, M. Schwarz, A. Rochow, B. Pätzold, R. Memmesheimer, M. Schreiber, and S. Behnke, "NimbRo wins ANA Avatar XPRIZE Immersive Telepresence Competition: Human-Centric Evaluation and Lessons Learned," *International Journal of Social Robotics*, 2023.
- [15] M. Tröbinger, A. Naceri, X. Chen, H. Sadeghian, and S. Haddadin, "Identification of a generalized base inertial parameter set of robotic manipulators considering mounting configurations," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [16] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic analysis of 7 DOF anthropomorphic arms," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 824–830.
- [17] R. Kirschner, K. Karacan, A. Melone *et al.*, "Categorizing robots by performance fitness into the tree of robots," *Nature Machine Intelligence*, 2025.
- [18] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010.
- [19] S. Tittel, "Analytical solution for the inverse kinematics problem of the Franka Emika panda seven-DOF light-weight robot arm," in *20th Intern. Conference on Advanced Robotics (ICAR)*, 2021, pp. 1042–1047.
- [20] Y. Wang, L. Li, and G. Li, "An analytical solution for inverse kinematics of 7-DOF redundant manipulators with offsets at elbow and wrist," in *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, vol. 6. IEEE, 2022, pp. 1096–1103.
- [21] L. Jiang, X. Huo, Y. Liu, and H. Liu, "An integrated inverse kinematic approach for the 7-DOF humanoid arm with offset wrist," in *IEEE int. conf. on robotics & biomimetics (ROBIO)*, 2013, pp. 2737–2742.
- [22] Y. He and S. Liu, "Analytical inverse kinematics for Franka Emika panda—a geometrical solver for 7-DOF manipulators with unconventional design," in *IEEE Int Conf Cir, Mech & Autom (ICCMA)*, 2021.
- [23] —, "Franka analytical IK," https://github.com/ffall007/franka_analytical_ik, 2024, accessed: 28-Feb-2025.
- [24] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, USA, 2010.
- [25] D. L. Pieper, *The kinematics of manipulators under computer control*. Stanford University, 1969.
- [26] A. J. Elias and J. T. Wen, "IK-Geo: Unified robot inverse kinematics using subproblem decomposition," *Mechanism and Machine Theory*, vol. 209, p. 105971, 2025.
- [27] P. López-Custodio, R. Fu, J. Dai, and Y. Jin, "Compliance model of Exechon manipulators with an offset wrist," *Mechanism and Machine Theory*, vol. 167, p. 104558, 2022.
- [28] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs *et al.*, "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.
- [29] "MoveIt! Tutorials - IKFast kinematics solver," http://docs.ros.org/en/kinetic/api/moveit_tutorials/html/doc/ikfast/ikfast_tutorial.html, accessed: Sep-2025.
- [30] "Franka Control Interface (FCI) - Panda description," https://frankaemika.github.io/docs/franka_ros.html#franka-description, accessed: 28-Feb-2025.
- [31] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2007.
- [32] X. Yang, Z. Zhao, Z. Xu, Y. Li, J. Zhao, and H. Liu, "General inverse kinematics method for 7-DOF offset manipulators based on arm angle parameterization," *Acta Astronautica*, vol. 202, pp. 263–277, 2023.
- [33] M. Asgari, I. A. Bonev, and C. Gosselin, "Singularities of ABB's YuMi 7-DOF robot arm," *Mechanism and Machine Theory*, 2025.
- [34] A. J. Elias and J. T. Wen, "Redundancy parameterization and inverse kinematics of 7-DOF revolute manipulators," *Mechanism and Machine Theory*, vol. 204, p. 105824, 2024.