# Geometric Approach for Inverse Kinematics of the FANUC CRX Collaborative Robot

Manel Abbes, Gérard Poisson

*robotics*

MDPI

*Article*

# Geometric Approach for Inverse Kinematics of the FANUC CRX Collaborative Robot

**Manel Abbes * and Gérard Poisson**

PRISME, University of Orléans, INSA-CVL, UR 4229, 18000 Bourges, France; gerard.poisson@univ-orleans.fr
* Correspondence: manel.abbes@univ-orleans.fr

**Abstract:** Because they are safe and easy to use, collaborative robots are revolutionizing many sectors, including industry, medicine, and agriculture. Controlling their dynamics, movements, and postures are key points in this evolution. Inverse kinematics is then crucial for robot motion planning. In 6R serial robots, achieving a desired pose is possible with different joint combinations. In this paper, our focus lies in studying forward and, mainly, inverse kinematics of the FANUC CRX-10iA cobot, a 6R cobotic arm with a non-spherical wrist. Its specific structural parameters implies that no analytical solutions exist except for some particular situations. FANUC does not provide the complete set of inverse kinematic solutions, even when 16 solutions are possible, only 8 of them are provided in Roboguide software. Furthermore, the existing literature on joints-to-workspace mapping for CRX cobots is currently very limited. It either lacks or provides partial or inconsistent inverse kinematics analysis. We present and detail a novel fully geometric method for numerically solving inverse kinematics meeting the requirement of high precision and a fast response. This approach provides both the exact number of inverse kinematics solutions and the sets of joint angles even for singular configuration. Its effectiveness was verified through simulations using the Roboguide Software and experimentation on the actual CRX-10iA cobot. Several examples (8, 12, or 16 inverse kinematic solutions) have enabled us to validate and prove the robustness and reliability of this geometric approach.

**Keywords:** cobots; inverse kinematics; FANUC CRX-10*i*A; 6R manipulators; non-spherical wrist; geometric; pose; posture; aspects

check for updates

**Citation:** Abbes, M.; Poisson, G. Geometric Approach for Inverse Kinematics of the FANUC CRX Collaborative Robot. *Robotics* **2024**, *13*, 91. https://doi.org/10.3390/robotics13060091

Academic Editor: Raffaele Di Gregorio

Received: 12 March 2024
Revised: 3 June 2024
Accepted: 4 June 2024
Published: 14 June 2024



## 1. Introduction

From industrial manufacturing to the medical field, robotics is nowadays being used in all the important sectors of life and society. The use of robots in the medical field is becoming increasingly prevalent, revolutionizing operating rooms around the world [1,2]. Over the past decade, there has been a notable emergence of collaborative robots (cobots) [3]. Consequently, cobots are being utilized more frequently to assist or even conduct delicate surgeries, offering the potential for faster, more precise, and less risky interventions [4].

Robotics manufacturers offer for instance promising cobotic modalities for many medical applications. The UR3 from Universal Robots, a 6R cobot, was used for a tele-robotic ultrasound solution for remote diagnostic echography [5]. In [6], authors used another 6R cobot, the TX2 60 from Stäubli company, for the comanipulation of an ultrasound probe. Furthermore, a FANUC cobot, the CRX-10*i*A, was used in [7] to present a study regarding the design and the experimental setup of a cobotic system for targeted drug delivery in the inner ear, where a dedicated magnetic probe [8] had to be controlled in position and orientation, as computed based on MRI pre-operative images [9].

To perform these different tasks and more, the knowledge of the robot's kinematics is most often required. Due to the interesting but complex nature of robots control, several approaches to the research of kinematic models (both Forward and Inverse) have been taken. The most challenging task for most cases is still Inverse Kinematics (IK) computing.

While classical methods such as those of Pieper [10] or Paul [11] can be applied to compute Inverse Kinematic solutions for Stäubli or, respectively, UR cobots mentioned above, they cannot be applied to the CRX series because of their kinematic architecture. Indeed, since 1968, Pieper [10] showed that a literal IK model can be defined for 6R robots with a spherical wrist. Up to eight solutions can be computed. In 1981, Paul [11] proposed an approach for solving IK of certain 6R robots, with a non-spherical wrist, that consists of successively multiplying the two sides of the Forward Kinematic (FK) equations by transformation matrices and thus isolate the joint values one after another. In [12], authors used this approach for solving the IK of the UR5 cobot: a 6 DoF mechanism with a non-spherical wrist, where J2, J3 and J4 axes are parallel. This robot, and all of the UR series, have four or eight inverse kinematic solutions, depending on the desired pose in the workspace.

For CRX cobots series, the 6R cobotic arm has a non-spherical wrist with perpendicular axes of J2 and J3. Thus, analytical IK solutions cannot be provided and are only available for some special cases. In this case, IK solutions can be computed, based on algebraic or geometric approaches, using numerical procedures [13–16].

For over 30 years, researchers have dealt with inverse kinematics problems. In [17,18], authors has addressed the inverse kinematics problem of 6 DoF manipulators and proposed a simplification process to analytically solve inverse kinematic equations if all adjoining joint axes are parallel or orthogonal. However, no analytical solution of the inverse kinematics applicable to arbitrary structures is available. They have presented, in [19], an iterative inverse kinematic approach based on the Newton–Raphson method that can be applied to manipulators requiring a numerical technique. This approach led them to discover that a 6 DoF manipulator is able to reach an end-effector pose (position and orientation) in 16 different configurations. It was also shown in [20,21], by an algebraic approach, proposed by Raghavan and Roth, that inverse kinematics solutions can be determined by a 16th degree univariate polynomial, which means up to 16 IK solutions, for 6R serial manipulators, can be computed. However, these general approaches suffer from numerical inaccuracies in several configurations. Therefore, researchers have been studying possible improvements and generalization of these approaches ever since [22–27].

Solving IK of CRX cobots series of FANUC remains challenging. Their architecture leads to 16 IK solutions. In fact, the degrees of characteristic polynomials depend on the structural design parameters of 6 DoF robotic arms (adjacent parallel, perpendicular, or intersecting axes) [28]. Different approaches have been shown to solve this issue. A differential control was used in [7] to solve IK positioning of a CRX cobot. This approach presents some inherent limitations. Indeed, finding an efficient initial pose for convergence while avoiding singularities during motion are still factors that affect or limit its applicability. In [29], authors present an iterative approach that bypasses the use of Jacobian and rotational matrices, it is based on several iterations through each joint, starting with the last joint (end-effector position) and connecting it to the first one. This method does not consider the end-effector orientation; moreover, it can be stuck in a local optimum leading to it being unable to find a global optimum. Kucuk et al. proposed in [30] a numerical algorithm where the idea is to chose a random initial value for the first joint in order to reduce the IK problem to 1 DoF and then compute analytically the rest of the joint parameters. These approaches provide a valid IK solution but without mapping all possible IK solutions which prevents changing aspects to find a better configuration for the same probe position and orientation.

In more recent works, an algebraic approach was used in [31,32] to determine IK solutions of the CRX-10*i*A/L, a particular cobot of the CRX series. The used methods involve solving a 16th degree univariate polynomial, which is constructed from a set of six constraint equations representing mechanism loop closures, along with the parameterization of a set of unknown variables. The authors in [31] state considering the vector of unknown variables characterizing the position of the frame attached to the fourth robot body makes the resolution possible. In [32], authors show precisely that the 16th degree closure polynomial they obtained can be factorized in two identical 8th degree polynomials,

which provides a significant simplification for zero determination. In these studies, authors have dealt with the IK problem of 6R with non-spherical wrist arms, more particularly CRX cobots; however, no attention has been given to the accuracy and efficiency of the provided IK solutions which can be an issue for exact manipulations. For instance, in [31], the 16 given solutions of the CRX Inverse Kinematics are not completely valid on the real cobot and present pose accuracy issues. Furthermore, in [32], an issue with the third joint values can be pointed out, distorting 8 pose configurations out of 16. Moreover, due to the simplicity of the chosen example, where J6 is vertical, an explicit inverse kinematics solution exists, thus reducing the algebraic solution interest, which is not the case for any desired pose.

The motivation of this paper is thus dealing with these limits by approaching 6R manipulator IK solutions with high accuracy and effectiveness. A new inverse kinematic algorithm for the CRX-10*i*A cobot is presented. The proposed method is based on a geometric approach and shows to be useful for solving IK problems with high precision and efficiency. The various steps are detailed in the following, aiming to simplify the implementation process for readers interested in utilizing this method for CRX IK control. Specific examples are provided to validate the effectiveness of the proposed method, both through simulation on Roboguide Software and experimentation with the actual cobot.

## 2. Materials and Methods

### *2.1. FANUC CRX Cobot*

The FANUC CRX-10*i*A is a 6 Degrees of Freedom (DoF) cobot with a 6R serial architecture (Figure 1). Its compact design and high reliability (the constructor gives a position repeatability of 0.05 mm) allows for the easy integration into any work area making it very popular in industry and robotics research. As a collaborative robot, it is intended for applications that include human interaction. It offers then all the safety guarantees (speeds, forces) required for this purpose. In fact, its sleek design, devoid of sharp edges, and illuminated indicators are all tailored for safe interaction and collaboration with humans.

The kinematics architecture of this cobot closely resemble those of classic 6R serial manipulators, in particular the original FANUC series. However, the main difference lies in the fact that the directions of J2, J3, and J4 are not parallel, which is now combined with the fact that the wrist is not spherical. These two simultaneous features considerably complicate the resolution of inverse kinematics.

As of today, the FANUC CRX cobot series consists of five cobots. They may vary in size, but they all share the same kinematic architecture. This implies that the geometric approach developed in this paper can be implemented in all CRX cobots.

### *2.2. Analysis of the 16 IK Solutions Proposed in [31]*

As explained earlier, the specific architecture of the CRX cobot makes the resolution of its inverse kinematics challenging. Recently, authors in [31] have dealt with the inverse kinematics of this particular cobot. Authors proposed an algebraic numerical IK solutions for the CRX-10*i*A/L cobot. They provided a numerical example illustrating, with a particular focus on the following posture [$Car_{\#4}$], 16 IK solutions.

A robot configuration in the joint space, also called a posture, is then denoted as [$Car_{\#i}$] (in reference to the first author L. Carbonari).

$[Car_{\#4}] = [+78, \quad +131, \quad +24, \quad +42, \quad -60, \quad -10]^T$ (in degrees).

Then, in this paper, authors presented 15 additional postures, intended to achieve the same pose (the tool position and orientation in the task space).

For the following, this notation is considered for these 16 postures [$Car_{\#i}$], $i \in [1, 16]$.

Interested in the need for inverse kinematics, we studied article [31] to apply its method. We thoroughly analyzed the example proposed by the authors, which illustrates a pose with 16 IK solutions, as shown in Table 1.

**Table 1.** The joint values $[J_i]$ for the 16 IK solutions proposed in [31] to reach the same pose. The right columns indicate an eventual equality (=) between postures, then accuracy in mm between the desired and the obtained TCP position, and finally if a solution is valid (Yes or Not).

| | ***J*1** | ***J*2** | ***J*3** | ***J*4** | ***J*5** | ***J*6** | = | **Dist.** | **Val** |
|---|---|---|---|---|---|---|---|---|---|
| $Car_{\#1}$ | −30.30 | 33.99 | 136.01 | 143.27 | −48.79 | 77.61 | - | 10 | N |
| $Car_{\#2}$ | 149.69 | 146.00 | 43.98 | −36.72 | −48.79 | 77.61 | - | 10 | N |
| $Car_{\#3}$ | −101.99 | 48.99 | 155.99 | −138.00 | −59.99 | −9.99 | - | 0.1 | Y |
| $Car_{\#4}$ | 78.00 | 131.00 | 24.00 | 42.00 | −60.00 | −10.00 | - | – | Y |
| $Car_{\#5}$ | 39.90 | 28.21 | 161.19 | −75.16 | 116.22 | −90.34 | - | 0.08 | Y |
| $Car_{\#6}$ | −140.09 | 151.78 | 18.80 | 104.83 | 116.22 | −90.34 | - | 0.08 | Y |
| $Car_{\#7}$ | −93.98 | 47.62 | 154.56 | −144.11 | −55.12 | −2.97 | - | 0.04 | Y |
| $Car_{\#8}$ | 86.01 | 132.37 | 25.43 | 35.88 | −55.12 | −2.97 | - | 0.12 | Y |
| $Car_{\#9}$ | −93.98 | 47.62 | 154.56 | −144.11 | −55.12 | −2.97 | $Car_{\#7}$ | 0.04 | N |
| $Car_{\#10}$ | 86.01 | 132.37 | 25.43 | 35.88 | −55.12 | −2.97 | $Car_{\#8}$ | 0.12 | N |
| $Car_{\#11}$ | −93.98 | 47.62 | 154.56 | −144.11 | −55.12 | −2.97 | $Car_{\#7}$ | 0.04 | N |
| $Car_{\#12}$ | 86.01 | 132.37 | 25.43 | 35.88 | −55.12 | −2.97 | $Car_{\#8}$ | 0.12 | Y |
| $Car_{\#13}$ | −29.41 | 33.90 | 135.67 | 142.25 | −49.24 | 78.79 | - | 10.3 | N |
| $Car_{\#14}$ | 150.58 | 146.09 | 44.32 | −37.74 | −49.24 | 78.79 | - | 10.3 | N |
| $Car_{\#15}$ | 114.69 | 42.07 | 151.46 | − 23.88 | 170.53 | 10.81 | - | 0.03 | Y |
| $Car_{\#16}$ | −65.30 | 137.92 | 28.53 | 156.11 | 170.53 | 10.81 | - | 0.04 | Y |

In our assessment, this example actually provides only 8 IK solutions.
In fact, after this analysis, we noticed the following:

- Four of the sixteen proposed solutions are not recognized as distinct because they are already included in the table (see column "=" of Table 1):
  [ $Car_{\#9}$] and [$Car_{\#11}$] are identical to [$Car_{\#7}$], (same joint parameters), and
  [$Car_{\#10}$] and [$Car_{\#12}$] are identical to [$Car_{\#8}$].
  So, [$Car_{\#9}$] to [$Car_{\#12}$] of the article are not distinct solutions.
- Four other solutions are not considered as valid because they fail to yield a relevant outcome in achieving the desired pose (see column "Dist" of Table 1). The distance "Dist" between two poses is defined as the Euclidean distance between the Tool Center Points (TCPs) of the two poses under consideration.
  This concerns [$Car_{\#1}$], [$Car_{\#2}$], [$Car_{\#13}$], and [$Car_{\#14}$]. For example, the distance between the pose given by posture [$Car_{\#13}$] and [$Car_{\#4}$] is 10.3 mm. Thus, these solutions present an accuracy issue and are not considered as valid.

Since our aim was to find a method of solving the IK, one of the first examples we will try to solve, after describing the method, is that of article [31]. These solutions (8 in fact and not 16) are given later on.

### *2.3. Accuracy Analysis of the 8 Valid Solutions Proposed in [31]*

On another hand, the accuracy of the provided IK solutions (the joints values) is of $10^{-2}$ deg. As a consequence, the global accuracy of the valid achieved positions (with a robot arm span of 1.4 m) can reach 0.12 mm (see column "Dist" of Table 1 for [$Car_{\#8}$]).

As 0.1 mm of accuracy can be unsatisfactory for certain applications requiring extremely precise positioning, this means that we need to find the joint solutions to at least $10^{-3}$ deg resolution rather than $10^{-2}$ deg.

### *2.4. Specifications for a Robust and Accurate IK*

This analysis highlights the necessity for a robust and precise IK solution.

It should allow us to determine the following: (i) the exact number of solutions, and (ii) providing joint values with satisfactory accuracy ($10^{-3}$ deg is an optimum), especially considering that FANUC robots operate with precision up to $10^{-3}$ (deg and mm).

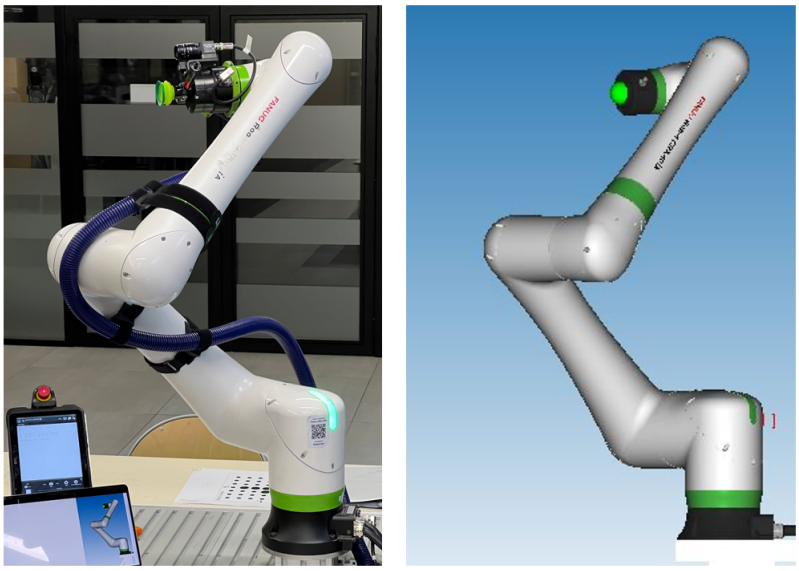

**Figure 1.** FANUC CRX-10*i*A cobot and its homologous in Roboguide software. This configuration corresponds to an end-effector pose, with 16 accessible solutions within the joint limitations.

### *2.5. FANUC CRX10-iA Forward Kinematics*

A posture for a 6 DoF serial robot is a configuration of its joint parameters defined by the vector:

$$[J] = [J1,\ J2,\ J3,\ J4,\ J5,\ J6]^T.$$

For a given posture $[J]$, the corresponding pose, obtained by the FK model is as follows:

$$P = FK(J) = [X,\ Y,\ Z,\ W,\ P,\ R]^T,$$

with $(X, Y, Z)$ the coordinates of the end-effector center (or Tool Center Point), and $(W, P, R)$ the three rotation angles with Cardan's approach to characterize the end-effector orientation. These three angles are, respectively, measured along the three axes $z$, $y$, and $x$ of the robot reference frame $R_0$.

#### 2.5.1. Denavit–Hartenberg Modified Formalism and **DHm** Parameters

The formalism we used for modeling our robots is inspired from the modified Denavit-Hartenberg approach [33], proposed in 1986 by Khalil and Kleinfinger, denoted as **DHm**, and inspired from Denavit and Hartenberg, [34] in 1955. The four elementary movements from frame $R_{i-1}$ to $R_i$ and the four associated DHm parameters are as follows (see Figure 2).

- Rotation of an angle $\alpha_{i-1}$ around $X_{i-1}$;
- Translation of $a_{i-1}$ along $X_{i-1}$;
- Rotation of an angle $\theta_i$ around $Z_i$;
- Translation of $r_i$ along $Z_i$.

The transformation matrix ${}^{i-1}T_i$ is then obtain from these four parameters, see Equation (1).

Table 2 provides all the DHm parameters for the 6 joints of the CRX10-*i*A cobot.

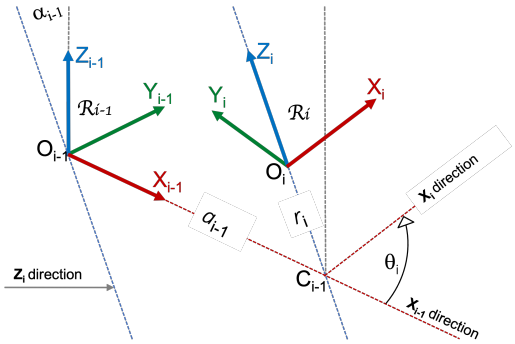


**Figure 2.** The four kinematic transformations between two successive frames $R_{i-1}$ and $R_i$ in the "Denavit–Hartenberg modified" formalism, with the four parameters: $\alpha_{i-1}$, $a_{i-1}$, $\theta_i$, and $r_i$.

$$ {}^{i-1}T_i = \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i.c\alpha_{i-1} & c\theta_i.c\alpha_{i-1} & -s\alpha_{i-1} & -r_i.s\alpha_{i-1} \\ s\theta_i.s\alpha_{i-1} & c\theta_i.s\alpha_{i-1} & c\alpha_{i-1} & r_i.c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1) $$

**Table 2.** CRX-10*i*A DHm parameters: lengths (mm), angles (deg). $Ji$, the joint values, are expressed following the same convention as FANUC.

| **Link** | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ |
|---|---|---|---|---|---|---|
| $a_{i-1}$ | 0 | 0 | 540 | 0 | 0 | 0 |
| $\alpha_{i-1}$ | 0 | −90 | +180 | −90 | +90 | −90 |
| $\theta_i$ | $J1$ | $J2-90$ | $J2+J3$ | $J4$ | $J5$ | $J6$ |
| $r_i$ | 0 | 0 | 0 | −540 | 150 | −160 |

We aim to develop a forward kinematic model that is fully compatible with the real cobot, ensuring that the joint values and pose parameters are identical between the model and the robot for any achievable configuration.

To respect this, firstly, the $Z_i$ axis, as depicted in Figure 3, must be oriented in the direction of increasing joint values. Secondly, the coupling between $J2$ and $J3$ imposed by FANUC must be considered. This coupling arises because when $J2$ is manually moved, $J3$ moves in the opposite direction, ensuring that the wrist maintains the same orientation relative to the ground. This peculiarity of FANUC robots is of significant interest to programmers. As a consequence, there is a term $(J2+J3)$ present in column $L3$ of the DHm Table 2. Although this inclusion may seem unconventional in a DHm table, it accurately reflects the behavior of FANUC robots.

$$ {}^{6}T_{tool} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2) $$

$$ {}^{0}T_{tool} = {}^{0}T_1 \cdot {}^{1}T_2 \cdot {}^{2}T_3 \cdot {}^{3}T_4 \cdot {}^{4}T_5 \cdot {}^{5}T_6 \cdot {}^{6}T_{tool} \quad (3) $$

The homogeneous matrix ${}^{0}T_{tool}$ (see Equation (3)) is then used to determine the position and orientation of the end-effector in the frame $R_0$, as a function of the 6 joint parameters $Ji$. Here, we assume that the Tool Center Point (TCP) is located at the center of the end-effector plate face (resulting in zero values in column 4 of ${}^{6}T_{tool}$. However, it is important to note that any TCP coordinate can be considered. We denote $T_{i,j}$ as the element of the $i$th row and $j$th column of ${}^{0}T_{tool}$.

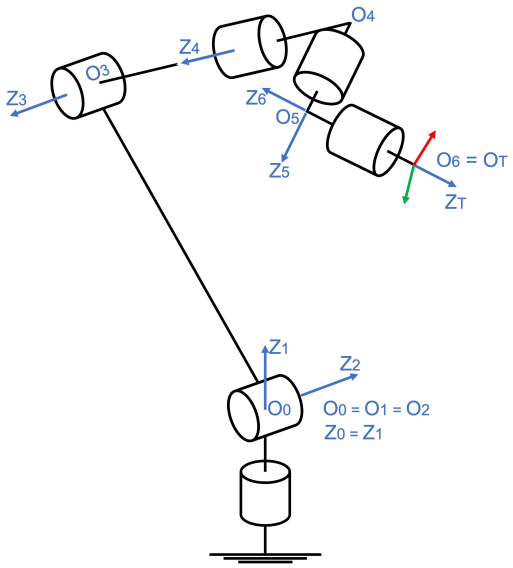


**Figure 3.** Kinematic diagram of the 6 DoF CRX-10*i*A cobot.

Determination of the Position Coordinates

The three coordinates $X$, $Y$, and $Z$ represent the position of the TCP in the base frame $R_0$. They are obtained by multiplying the homogeneous matrix ${}^{0}T_{tool}$ by the origin vector ${}^{Tool}O_6$, which is the vector of the homogeneous coordinates of the point $O_6$ (in this case, the TCP). In the tool frame,

$$ {}^{Tool}O_6 = [0,\ 0,\ 0,\ 1]^T \tag{4} $$

Then,

$$ [X,\ Y,\ Z,\ 1]^T = {}^{0}T_{tool} \cdot [0,\ 0,\ 0,\ 1]^T. \tag{5} $$

Determination of the Orientation Coordinates

The three parameters $W$, $P$, and $R$ represent the Cardan angles, which describe the tool orientation relative to the base frame $R_0$.

Using the FANUC's convention (which may differ from other robot manufacturers),

$W$, for "yaW" angle (also $\gamma$ in the literature), is measured around the $X$ axis,
$P$, for "Pitch" angle (also $\beta$), is measured around the $Y$ axis, and
$R$, for "Roll" (also $\alpha$), is measured around the $Z$ axis.

With this, the overall tool rotation matrix ${}^{0}R_{tool}$ (the top-left $3 \times 3$ block of ${}^{0}T_{tool}$) can be represented as the product of three elementary $3 \times 3$ rotation matrices corresponding to the three revolute movements around the $x$, $y$, and $z$ axes of $R_0$, respectively, with angles $R$, $P$, and $W$, as shown in Equation (6).

$$ {}^{0}R_{tool} = R_Z(R) \cdot R_Y(P) \cdot R_X(W) \tag{6} $$

Then, the three angles $W$, $P$, and $R$ are defined from the rotation matrix ${}^{0}R_{tool}$ extracted from ${}^{0}T_{tool}$ (obtained using DHm approach and Equation (3)), utilizing classic expressions from the literature, which we recall below (see Equation (7)) for the general case, excluding Cardan singularities (these occur when $cos(P) = 0$).

As the elements of ${}^{0}R_{tool}$ matrix are identical to those of ${}^{0}T_{tool}$, we denote them as $T_{i,j}$, as defined above.

$$ \begin{aligned} P &= atan2\left(-T_{3,1},\ \ sqrt\left(T_{1,1}^2 + T_{2,1}^2\right)\right) \\ R &= atan2\left(T_{2,1}/cos(P),\ \ T_{1,1}/cos(P)\right) \\ W &= atan2\left(T_{3,2}/cos(P),\ \ T_{3,3}/cos(P)\right) \end{aligned} \tag{7} $$

The particular cases of singularity associated with the pitch angle $P$ has been considered in our algorithm. However, the specific method used for this purpose is not detailed here, as it is of limited interest for understanding the Forward Kinematics algorithm.

Ultimately, thanks to this Forward Kinematics (FK) model (see Equations (3), (5) and (7)), we are able to compute the robot's tool pose parameters relative to the 6 joint parameters $Ji$ (see Equation (8)):

$$Pose\,(J) \;=\; Pose\,([J1,\;\; J2,\;\; J3,\;\; J4,\;\; J5,\;\; J6]^T) \;=\; [X,\;\; Y,\;\; Z,\;\; W,\;\; P,\;\; R]^T \tag{8}$$

The results obtained using this FK model have been confirmed on both the real CRX-10$i$A cobot and the Roboguide software. FANUC Roboguide is a robot simulator that perfectly emulates FANUC robots movements and dynamic behavior, greatly reducing the time required to create new motion configurations. Robotic cells can be designed, tested, and modified entirely offline which enables us to evaluate different configurations of the robot virtually before implementing them in a real-world setting.

#### 2.5.2. Forward Kinematic Model Validation

As example, we chose the particular configuration [Car$_{\#4}$] mentioned above.

[Car$_{\#4}$] is the configuration #4 of the example proposed by Carbonari et al. in [31].

[Car$_{\#4}$] = $[+78,\;\; +131,\;\; +24,\;\; +42,\;\; -60,\;\; -10]^T$, where angles are in degrees.

Considering the specificities of FANUC, the equivalent robot posture of [Car$_{\#4}$] configuration, should be expressed as [A$_{\#4}$].

From the Carbonari's notation and frames, [Car$_{\#i}$] = $[q1,\;\; q2,\;\; q3,\;\; q4,\;\; q5,\;\; q6]^T$, we can express the same cobot posture [A$_{\#i}$], with six FANUC-compatible joint parameters thanks to Equation (9), where $[J] \;=\; [J1,\;\; J2,\;\; J3,\;\; J4,\;\; J5,\;\; J6]^T$.

$$\begin{array}{lll} J1 \;=\; +\,q1 & & J4 \;=\; -\,q4 \\ J2 \;=\; 90 \,-\, q2 & & J5 \;=\; +\,q5 \\ J3 \;=\; -\,q3 - J2 & & J6 \;=\; -\,q6 \end{array} \tag{9}$$

Then, the posture [Car$_{\#4}$] corresponds to [A$_{\#4}$], see Equation (10):

$$[A_{\#4}] \;=\; [+78,\;\; -41,\;\; +\;17,\;\; -42,\;\; -60,\;\; +10]^T \tag{10}$$

We tested this joints configuration [A$_{\#4}$] on the two cobots CRX10-$i$A/L and CRX-10$i$A. The postures are represented in Figure 4, and they differ only because of changes in the parameter $a_2$.

- For CRX-10$i$A/L, the cobot considered in [31], we found

$$Pose_{A\,(10iA/L)} \;=\; [X,\;\; Y,\;\; Z,\;\; W,\;\; P,\;\; R]^T \;=$$

$$[57.132,\;\; 178.588,\;\; 522.657,\;\; -131.819,\;\; -45.268,\;\; 61.453]^T \tag{11}$$

In [31], authors provided the following coordinates $[X,\;\; Y]^T$ = $[57.1,\;\; 178.6]^T$.

These match what we found in terms of position, but the resolution is $10^{-1}$ whereas FANUC robots resolution is $10^{-3}$. Furthermore, they did not specify the end-effector orientation.

- For CRX-10$i$A, the cobot we are considering in this work, we found

$$Pose_{B\,(10iA)} \;=\; [X,\;\; Y,\;\; Z,\;\; W,\;\; P,\;\; R]^T \;=$$

$$[80.321,\;\; 287.676,\;\; 394.356,\;\; -131.819,\;\; -45.268,\;\; 61.453]^T \tag{12}$$

The end-effector orientations are obviously identical for both cobots.

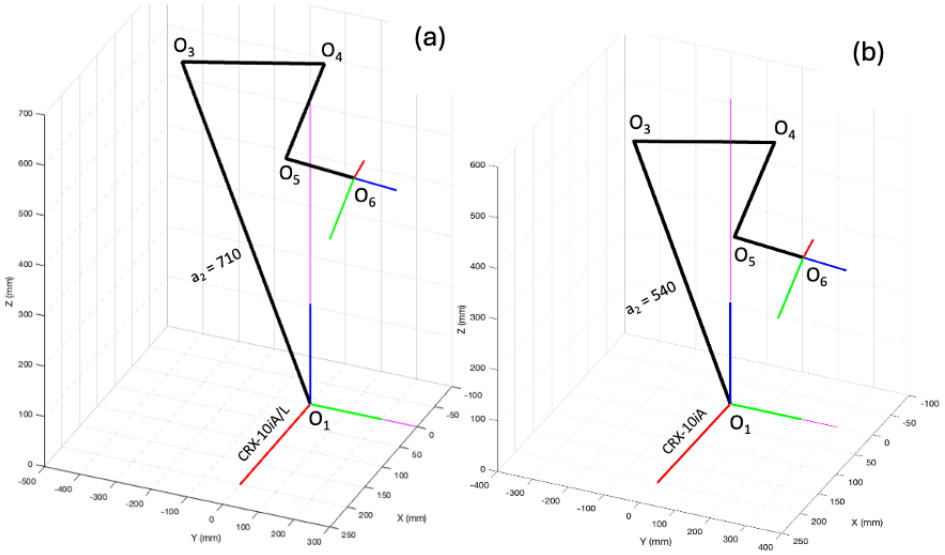


**Figure 4.** The two cobots (**a**) CRX-10*i*A/L and (**b**) CRX10-*i*A in the same configuration: $[J1,\ J2,\ J3,\ J4,\ J5,\ J6]^T = [A_{\#4}]^T = [+78,\ -41,\ +17,\ -42,\ -60,\ +10]^T$.

### *2.6. Geometric Approach for CRX-10iA Inverse Kinematics Resolution*

The inverse kinematic model provides 0, 4, 8, 12, or 16 solutions, depending on the different parameters of the *Desired-Pose*.

In certain exceptional configurations, intermediate numbers such as 2, 6, 10, or 14 solutions may occur, indicating the presence of "double" solutions.

Thus, cases with 2 solutions arise when the TCP precisely aligns with the edge of the workspace (i.e., 1249.039 mm from the center $O_0$).

We will see later another scenario, where the TCP is at the the workspace center, and which entails 14 solutions, including 12 distinct solutions and 2 double solutions.

When considering a *Desired-Pose* randomly selected within the cobot workspace, the highest probability corresponds to 8 solutions, estimated at approximately 80%.

The likelihood of encountering 12 solutions is notably lower.

The probability of encountering 4 or 16 solutions is even lower, approximately 5%.

Please note that these statistic values are approximate, as they depend on the parameterization of the poses: Cartesian or spherical coordinates for positions, Cardan or Euler formalisms, or Quaternions for orientations. Additionally, these values may slightly differ when considering real accessible poses, accounting for the cobot joint limitations.

Methodology for Geometric Approach to Inverse Kinematics

Given a *Desired-Pose* = $[X,\ Y,\ Z,\ W,\ P,\ R]^T$, the Inverse Kinematics geometrical approach gives the number $N$ of possible solutions to reach this *Desired-Pose*. If $N$ solutions exist, the algorithm provides $N$ vectors of 6 joint parameters for each solution:

$$[J1_i,\ J2_i,\ J3_i,\ J4_i,\ J5_i,\ J6_i]^T \ for\ i \in [1,\ N].$$

This method is developed through the following seven steps, see Figure 5. If the *Desired-Pose* is inside the workspace, the algorithm gives, at least, one Inverse Kinematic solution. These steps, and associated equations, will provide the reader with the ability to implement this IK model, applicable to any cobot of the CRX series, by simply modifying the four dimensional parameters of the cobot.

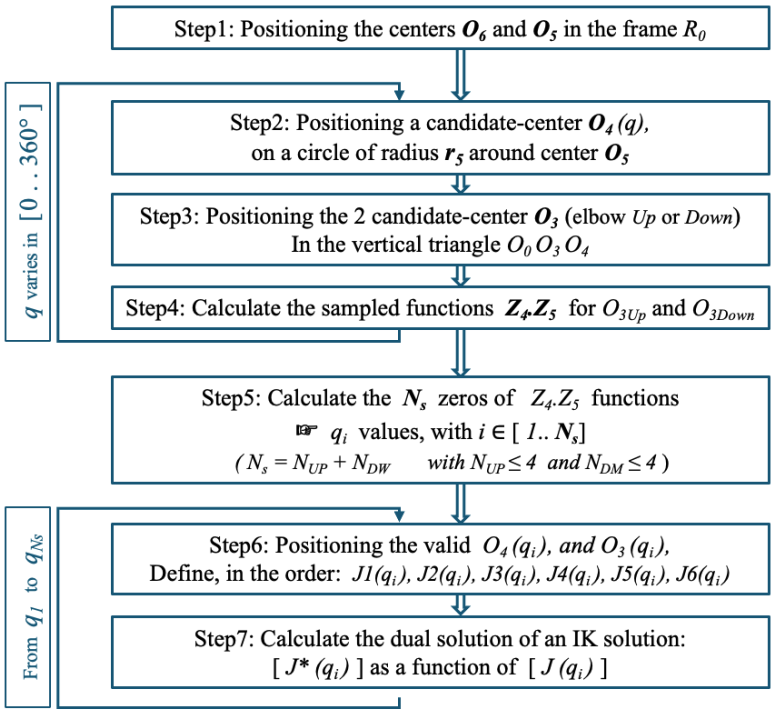


**Figure 5.** Flowchart of the CRX Inverse Kinematics determination.

**Step 1: Positioning the Points $O_6$ and $O_5$ in the frame $R_0$**

The three angles $W$, $P$, and $R$ are used to establish the rotation matrix ${}^0R_{tool}$, thanks to Equation (6). This $3 \times 3$ matrix is the left and upper block of the $4 \times 4$ matrix ${}^0T_{tool}$.

Furthermore, if we consider the translation vector $[X,\ Y,\ Z,\ 1]^T$, by combination, we can write the global homogeneous transformation matrix ${}^0T_{tool}$ and so the two expressions of $O_6$ and $O_5$, see Equation (13).

$$\begin{aligned} {}^0R_{tool} &= R_Z(R) \cdot R_Y(P) \cdot R_X(W) \\ {}^0O_6 &= [X,\ Y,\ Z,\ 1]^T \\ {}^0T_{tool} &= {}^0R_{tool} \oplus {}^0O_6 \\ {}^{tool}O_5 &= [0,\ 0,\ r_6,\ 1]^T \\ {}^0O_5 &= {}^0T_{tool} \cdot {}^{tool}O_5 \end{aligned} \tag{13}$$

**Step 2: Positioning the "Candidate-points" $O_4$, in the frame $R_0$**

Real point $O_4$ is at a constant distance $r_5$ from $O_5$.

By construction, the two vectors $\overrightarrow{O_5O_6}$ and $\overrightarrow{O_5O_4}$ are orthogonal.

That means that a candidate-point $O_4$ is on a circle whose center is $O_5$, and radius is $r_5$. This circle is in a plane, perpendicular to the vector $\overrightarrow{O_5O_6}$.

With $q \in [0,\ 2 \cdot \pi]$, a sample variable representing the characteristic angle of vector $\overrightarrow{O_5O_4}$ around axis $O_5O_6$, this point ${}^{tool}O_4(q)$, a function of $q$, can then be considered as a Candidate-point, as shown in System (14).

For example, $q$ can be varied with a 1 degree sample. There will then be 360 candidate-points $O_4$ along the considered circle.

$$\begin{aligned} {}^{tool}O_4(q) &= [r_5 \cdot cos(q),\ r_5 \cdot sin(q),\ r_6,\ 1]^T \\ {}^0O_4(q) &= {}^0T_{tool} \cdot {}^{tool}O_4(q) \end{aligned} \tag{14}$$

In Figure 6, with two different configurations, we can see the location of the Candidate-points $O_4(q)$ on the red circles around $O_5$.

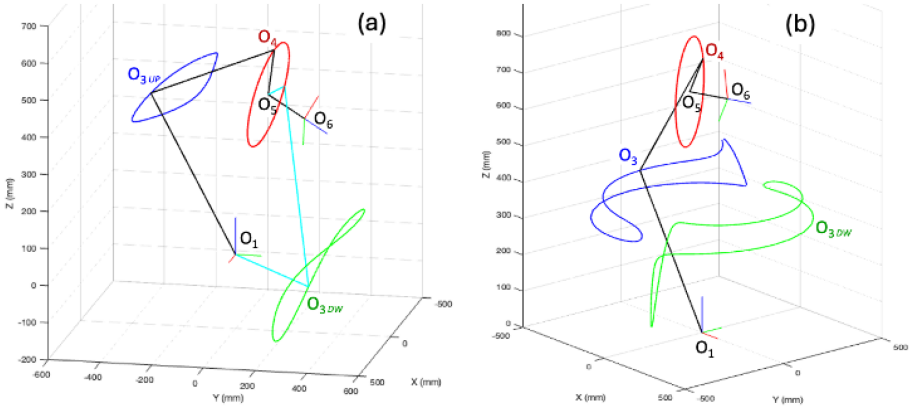


**Figure 6.** Location of the centers $O_4$ (red), $O_{3\,\mathrm{UP}}$ (blue), and $O_{3\,\mathrm{DW}}$ (green). **(a)**: example of a pose with 8 IK solutions. (**b**): example of a pose with 16 IK solutions.

**Step 3: Positioning the "Candidate-points" $O_3$, in the frame $R_0$**

Considering the position of a particular candidate-point $O_4(q)$, we can define the associated candidate-point $O_3(q)$ in the triangle $O_0O_3O_4$. In fact, there are two solutions for each $O_4(q)$: one above and one below (corresponding to left or right robot elbow, also called up or down), situated on either side of the line $O_0O_4$. These points $O_3$ serve as the apexes of a triangle, with the other two vertices identified as ($O_0$ and $O_4(q)$).

With $d_{04}$ the Euclidean distance from $O_0$ to $O_4(q)$, the lengths of the three sides of the triangle are known: $a_2$ (540 mm), $-r_5$ (540 mm), and $d_{04}$.

Note that there is a minus sign for $r_5$ because this value is negative in the DHm table.

Eventually, this triangle is in a vertical plane.

According to Inequality (15), for a candidate-point $O_4(q)$, there are two solutions, labeled as candidate $O_{3\mathrm{UP}}(q)$ and $O_{3\mathrm{DW}}(q)$. The first one, referred to as UP, corresponds to the "left elbow" configuration for the robot arm. The other solution, denoted as DW, corresponds to the "right elbow" configuration.

$$d_{04} \leq a_2 + (-r_5) \tag{15}$$

The coordinates for positions ${}^0O_{3\mathrm{UP}}(q)$ and ${}^0O_{3\mathrm{DW}}(q)$ are established and integrated in our algorithm using the relationships derived from the oblique triangle.

In Figure 6, the position ${}^0O_{3\mathrm{UP}}(q)$ is shown in blue and ${}^0O_{3\mathrm{DW}}(q)$ in green. Certainly, although this is not very easy to see on the plane projection of Figure 6, these two points lie on 3D curves over a sphere with a radius of $a_2$.

Figure 6a corresponds to the $Pose(A_{\#4})$, see Equation (12) for these *Pose* coordinates.

Figure 6b corresponds to a *Desired-Pose* for which 16 solutions are valid. For better readability, only one posture has been represented.

While the loci of $O_4(q)$ are represented as circles (red) in both cases, the 3D closed curves of the two loci of $O_3(q)$ (blue and green for *Up* and *Down* loci) corresponding to a 16-*solution Desired-Pose* (Figure 6b) are considerably less uniform compared to those corresponding to 8-*solution Desired-Pose* (Figure 6a).

**Step 4: Perpendicularity of $\overrightarrow{O_3O_4}$ and $\overrightarrow{O_4O_5}$ and the two $Z_4 \cdot Z_5$ dot-product values**

By construction, on the CRX-10*i*A cobot, the two vectors $\overrightarrow{O_3O_4}$ and $\overrightarrow{O_4O_5}$ are perpendicular. This means that the Candidate-positions $O_4(q)$ and $O_3(q)$ are viable if the dot-product between $\overrightarrow{O_3O_4}$ and $\overrightarrow{O_4O_5}$ is equal to zero.

In order to verify this condition, we computed the two dot-products $Z_4 \cdot Z_5$ (first for $O_{3\mathrm{UP}}$ and secondly for $O_{3\mathrm{DW}}$) of the two Candidate-unit-vectors of $\overrightarrow{O_3O_4}$ and $\overrightarrow{O_4O_5}$ (see Figure 3) and expressed them as functions of the sample variable $q$.

$$\begin{aligned} \mathrm{UP}(q) &= Z_4 \cdot Z_5 \quad for the couple O_4(q) and O_{3\mathrm{UP}}(q) \\ \mathrm{DW}(q) &= Z_4 \cdot Z_5 \quad for the couple O_4(q) and O_{3\mathrm{DW}}(q) \end{aligned}$$

Figures 7–10 show, for different *Desired-Poses*, the representation of these dot-products: $\mathrm{UP}(q)$ in blue and $\mathrm{DW}(q)$ in red. Parameter $q$ varies from 0° to 360°.

**Step 5: Determination of the zeros of the two $Z_4 \cdot Z_5$ dot-product values**

At the end of Step 4, the two functions $Z_4 \cdot Z_5$ are known as sample signals. The number of samples is the number taken for $q$ in Step2.

As these are dot-products, the various samples of $Z_4 \cdot Z_5$ move between $-1$ and $+1$. Finding a valid candidate-point $O_4$ is like finding the zeros of the continuous and bounded functions $Z_4 \cdot Z_5$ known per samples.

The number of intersections of the curves with the zero line is between 0 and 8. We denote as $N_{\mathrm{UP}}$ and $N_{\mathrm{DW}}$ the number of zeros, respectively, for the functions $\mathrm{UP}(q)$ and $\mathrm{DW}(q)$.

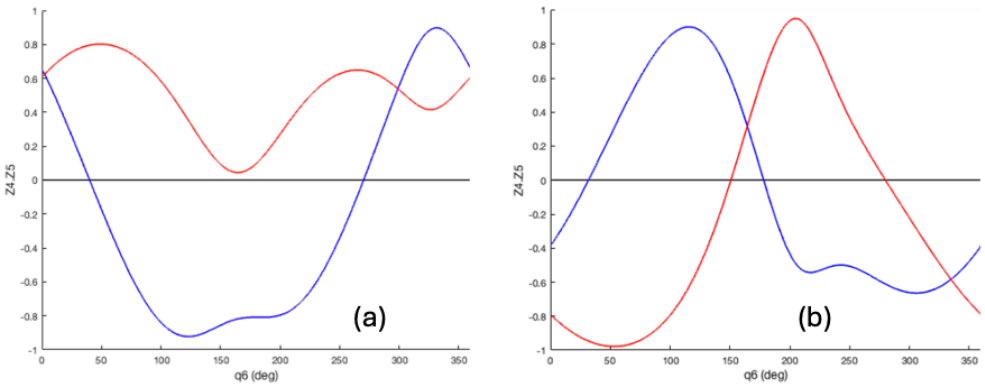


**Figure 7.** The two dot-products $Z_4 \cdot Z_5$ function of the parameter $q$: $\mathrm{UP}(q)$ in red and $\mathrm{DW}(q)$ in blue. (**a**): The blue curve intersects the zero line twice, whereas the red curve does not. (**b**): The two curves intersect the zero line twice. The cobot configuration is as depicted in Figure 6a.

Figure 7 illustrates examples where the two curves intersect the zero line 2 or 4 times.

In Figure 8, the curves intersect the zero line 6 or 8 times.

In Figure 9, two examples are presented where the curves intersect the zero line 3 or 7 times. In such cases, double solutions are observed: (a) the red curve and (b) the blue curve are tangent to the zero line for these double solutions.

The last example, depicted in Figure 10a, shows a specific configuration where the $\mathrm{UP}(q)$ and $\mathrm{DW}(q)$ solutions are not defined for all $q$ values between 0° and 360°. This occurs when the inequality (15) is not satisfied.

Figure 10b illustrates one robot's posture for this configuration. The end-effector is then positioned close to the outer boundary of the workspace. This particular configuration inevitably reduces the number of possible solutions (here $N\_\mathrm{UP} = 0$).

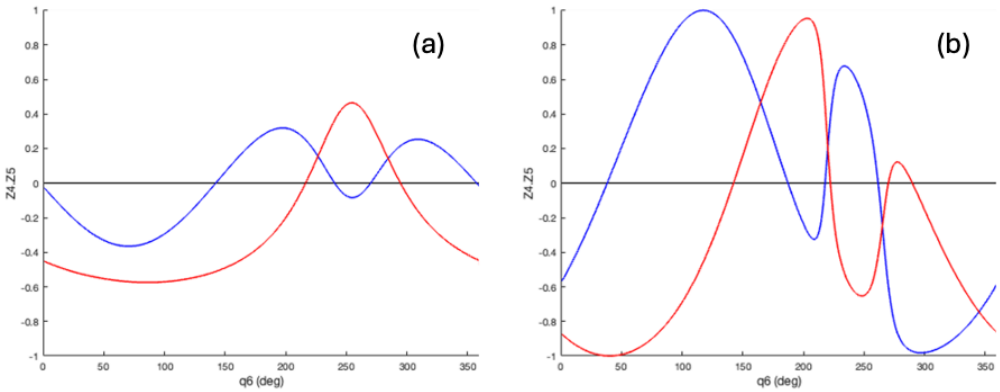


**Figure 8.** The two dot-products $Z_4 \cdot Z_5$ intersect 6 times (**a**) or 8 times (**b**) the zero line.

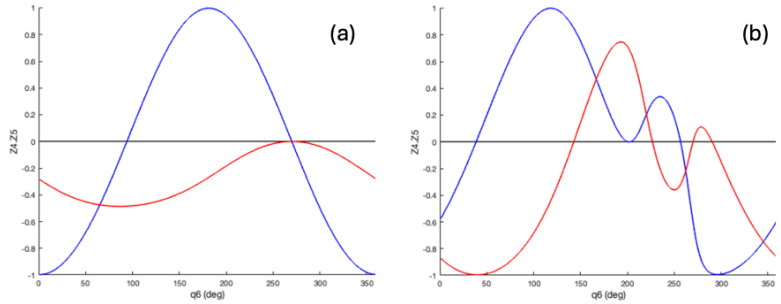


**Figure 9.** The two dot-products $Z_4 \cdot Z_5$ intersect 3 times (**a**) or 7 times (**b**) the zero line.

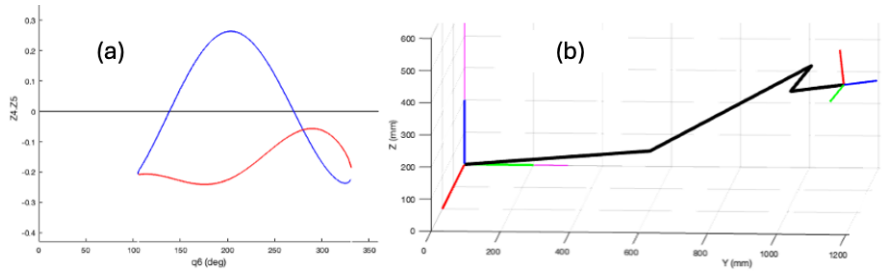


**Figure 10.** Configuration close to the outer boundary of the reachable workspace, for which $\mathrm{UP}(q)$ and $\mathrm{DW}(q)$ are not defined for all parameter $q$ (**a**). Corresponding the robot's posture (**b**).

At the end of this 4th process step, two functions $\mathrm{UP}(q)$ and $\mathrm{DW}(q)$ are established. The search for the zeros of these two functions gives:

Up to 4 values $q_{i_{UP}}$ ($N_{\mathrm{UP}} \leq 4$, most often 2; although, occasionally, it may be 0 or 4) corresponding to $\mathrm{UP}(q_{i_{UP}}) = 0$;

Up to 4 values $q_{i_{DW}}$ ($N_{\mathrm{DW}} \leq 4$, most often 2; although, occasionally, it may be 0 or 4) corresponding to $\mathrm{DW}(q_{i_{DW}}) = 0$.

As dot-products of unit vectors, the $\mathrm{UP}(q)$ and $\mathrm{DW}(q)$ functions are monotonic and their values lie $\in [-1, 1]$. They exhibit behavior akin to a "distorted sinusoidal curve" with two alternations, thus potentially providing a maximum of 4 zeros. Identifying zeros is a classic conventional and straightforward procedure, which we will not delve into extensively in this paper as it does not present a challenge for the IK problem.

The considered *Pose* corresponds to that of Equation (12), for which the positions of $O_3$ and $O_4$ are given in Figure 6a. In Figure 7b, the corresponding curves of $\mathrm{UP}(q)$ (in red) and $\mathrm{DW}(q)$ (in blue) are represented.

These two curves intersect the zero a total of 4 times. So, there are therefore 4 $q_i$ values. Listed from smallest to largest (in degrees),

$$\begin{aligned} q1_{DW} &= 31.782 \\ q1_{UP} &= 150.776 \\ q2_{DW} &= 178.196 \\ q2_{UP} &= 280 \end{aligned}$$

A specialized zero-finding procedure has been integrated to the algorithm. This procedure allows zeros to be obtained at a low computational cost (10 ms on a microcomputer, without any specific code optimization), with an accuracy better than $10^{-5}$ deg (more than sufficient for Fanuc robots, given their programming precision of $10^{-3}$ deg for joint values).

**Step 6: From $N_s$ valid $O_4$ and $O_3$ points to $N_s$ $[Ji]$ joint values**

From a *Desired-Pose*, the 4th process step gives the number $N_s$ of intersections with zero line ($N_{\mathrm{UP}} + N_{\mathrm{DW}}$), providing then $q_i$ solutions for which the candidate points $O_4$ and $O_3$ ($O_{3\mathrm{UP}}$ or $O_{3\mathrm{DW}}$) are valid.

For any considered $q_i$ among the $N_s$ established values, we can compute the correspondent point $O_4$ using Equation (14), then deduce the parameter of the cobot joint1 thanks Equations (16) and (17).

In fact, two solutions for *J1* exist, for a known point $O_4$, see Figure 11.

Equation (16) give the precession angle of the plane containing the triangle $O_0O_3O_4$. For a known precession, the two solutions for *J1* are given by Equations (17) or (18).

Equation (17) gives the *J1* primal value, and Equation (18) gives the *J*1* dual one. As we have $N_s$ solutions for $q_i$, that means there are $N_s$ primal solutions to the IK and on another hand $N_s$ dual ones, i.e., a total $N = 2 \cdot N_s$ of IK solutions.

Step 6 of the algorithm deals with the primal joint parameters, step 7 below with the dual ones.

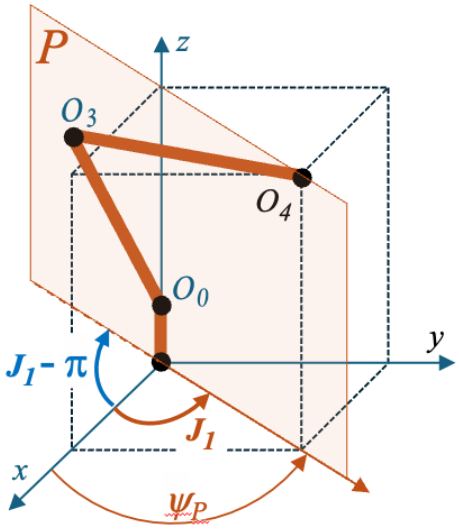


**Figure 11.** The two *J1* solutions for a precession $\Psi_P$ of the plane $P$ of the triangle $O_0O_3O_4$.

$$\begin{aligned} {}^{tool}O_4(q_i) &= [r_5 \cdot cos(q_i),\ r_5 \cdot sin(q_i),\ r_6,\ 1]^T \\ {}^{0}O_4(q_i) &= {}^{0}T_{tool} \cdot {}^{tool}O_4(q_i) \\ {}^{0}O_4(q_i) &= [X_{O_4},\ Y_{O_4},\ Z_{O_4},\ 1]^T \\ \psi_P &= atan2\,(Y_{O_4},\ X_{O_4}) \cdot 180/\pi \end{aligned} \tag{16}$$

$$J1 = \psi_P \tag{17}$$

$$J^*1 = \psi_P - \pi \tag{18}$$

For a known *J1* parameter, *J2* and *J3* can be expressed using the Equations of System (19), under the condition that the Inequality (15) is verified. Instructions given in Equation (19) must be followed in the specified order. We define and use a posture parameter $\delta$ such that:

($\delta = -1$) if the robot posture is UP and ($\delta = +1$) if the robot posture is DW.

Equation (19) is derived from relationships within the triangle ($O_0O_3O_4$), while considering the specific convention for FANUC's *Ji* parameters. The intermediate parameters for determining *J2* and *J3* can be seen in the $DHm\ (Z_1X_1)$ plane, in Figure 12.

*N.B.*, angles are measured positively from $Z_1$ to $X_1$, and $r_3$ is negative.

$$\begin{aligned} u1 &= atan2\,(Z_{O_4},\ \ sqrt\,(X_{O_4}^2 + Y_{O_4}^2)) \\ u2 &= -((X_{O_4}^2 + Y_{O_4}^2 + Z_{O_4}^2) - (a_2^2 + r_4^2))/(2 \cdot a_2 \cdot r_4) \\ u3 &= \delta \cdot acos(u2) \\ u4 &= atan2\,(-r_4 \cdot sin(u3),\ \ a_2 - r_4 \cdot cos(u3)) \\ J2 &= (\pi/2 - u1 + u4) \cdot 180/\pi \\ J3 &= (u1 + u3 - u4) \cdot 180/\pi \end{aligned} \tag{19}$$

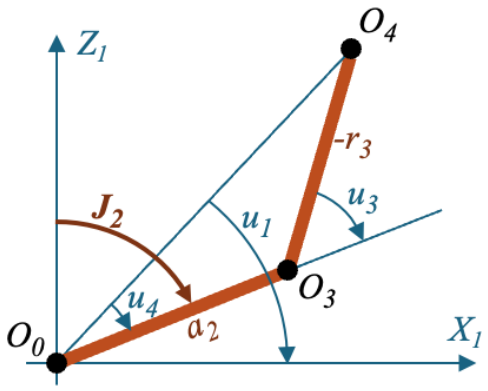


**Figure 12.** Angular parameters used in Equation (19) to determine *J*2. On this Figure, $\delta$ = +1.

For *J*4 determination, see the System of Equations (20).
We first establish the three transformation matrices:
${}^0T_1$, ${}^1T_2$, and ${}^2T_3$, as functions of *J*1, *J*2, and *J*3, as defined in Equation (1).
Then, we define the two matrices ${}^3T_4^*$ and ${}^4T_5^*$ with specific values $J_4^* = J_5^* = 0$.

$$\begin{array}{l} {}^0T_4^* = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4^* \\ {}^0T_5^* = {}^0T_4^* \cdot {}^4T_5^* \\ Z_4 = {}^0T_4^* \cdot [0,\ 0,\ 1,\ 0]^T \\ {}^0O_5^* = {}^0T_5^* \cdot [0,\ 0,\ 0,\ 1]^T \\ V0 = O_5^* - O_4 \\ V1 = O_5 - O_4 \\ scal1 = V0 \cdot V1 \\ scal4 = (V0 \wedge V1) \cdot Z_4 \\ s4 = sign(scal4) \\ J4 = s4 \cdot acos(scal1) \cdot 180/\pi \end{array} \quad (20)$$

The System of Equations (21) allows to determine *J*5 value using the same method explained above for *J*4 determination.

$$\begin{array}{l} {}^0T_4 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \\ {}^0T_6^* = {}^0T_4 \cdot {}^4T_5^* \cdot {}^5T_6^* \\ Z_5 = {}^0T_5^* \cdot [0,\ 0,\ 1,\ 0]^T \\ {}^0O_6^* = {}^0T_6^* \cdot [0,\ 0,\ 0,\ 1]^T \\ W0 = O_6^* - O_5 \\ W1 = O_6 - O_5 \\ scal3 = W0 \cdot W1 \\ scal5 = (W0 \wedge W1) \cdot Z_5 \\ s5 = sign(scal5) \\ J5 = s5 \cdot acos(scal3) \cdot 180/\pi \end{array} \quad (21)$$

For *J*6 determination, we employ a similar algorithm to the previous one, but it operates with rotation matrices rather than homogeneous ones, see the System of Equations (22).
We first establish ${}^4T_5$, with the value *J*5 obtained just above.
We calculate ${}^0T_6^*$, with $J_6^* = 0$, and deduce ${}^0R_6^*$, the rotational block element of ${}^0T_6^*$.

$$\begin{array}{ll} {}^0T_6^* = {}^0T_4 \cdot {}^4T_5 \cdot {}^5T_6^* & \\ \text{With } {}^{tool}T_6 = inv({}^6T_{tool}) & {}^0T_6 = {}^0T_{tool} \cdot {}^{tool}T_6 \\ \text{For } i \in [1;3] \text{ and } j \in [1;3] & {}^0R_6^*(i,j) = {}^0T_6^*(i,j) \\ \text{For } i \in [1;3] \text{ and } j \in [1;3] & {}^0R_6(i,j) = {}^0T_6(i,j) \\ R = inv({}^0R_6^*) \cdot {}^0R_6 & \\ J6 = atan2(-R(1,2),\ R(1,1)) & \end{array} \quad (22)$$

**Step 7: $N_s$ IK solutions by the dual property**

For CRX cobots, when a solution $[J]$ (6 joint parameters) is valid for a *Desired-Pose*, it is easy to verify that the other corresponding solution, called the dual solution, is also valid.

The global dual IK solution of $[J]$ is then denoted as $[J^*]$.

Its first joint parameter is the value $J^*1$, which is the dual of $J1$.

According to the FK model, when the dual change is applied to all matrices used in Equation (3), it can also be demonstrated that two dual joint solutions give the same pose.

The change in variables, for the 6 joints, between $[J]$ and $[J^*]$ is given by Equation (23).

$$\begin{aligned} [J\,] &= [J1, \quad J2, \quad J3, \quad J4, \quad J5, \quad J6]^T \\ [J^*] &= [J1 - 180, \quad -J2, \quad 180 - J3, \quad J4 - 180, \quad J5, \quad J6]^T \end{aligned} \tag{23}$$

This means that if a solution is known, consequently, another one is valid. It is important to note that in any configuration, two dual solutions are distinct because their joint values are never the same. However, their kinematic diagrams are exactly identical.

As shown at Step 6, we can have up to 8 solutions for a given *Desired-Pose*. Thus, this implies that after dual transformations, the maximum number $N_{max}$ of all viable IK solutions is 16.

## 3. Results: CRX IK Validation

This section illustrates results provided using this geometric approach dedicated to CRX series cobots. We present various examples of *Desired-Poses*, highlighting different scenarios in terms of the number of solutions produced by this IK model.

In our first example, typical of the most common scenarios, we observe eight solutions. The second example presents a case with 12 solutions, while the third illustrates a pose yielding 16 solutions. This range demonstrates the model's versatility in addressing various poses and the number of feasible solutions it can generate.

### *3.1. Examples with 8 IK Solutions for CRX-10iA/L and CRX-10iA*

The examples chosen here correspond to the poses described in Section 2.5.2.

- The first example concerns the CRX-10*i*A/L cobot, with the following desired values:

$$Pose_A = [57.132, \quad 178.588, \quad 522.657, \quad -131.819, \quad -45.268, \quad 61.453]^T$$

The different sets of joint angles for the eight solutions are given in Table 3. These solutions have been verified and shown to be valid on Roboguide Software.

It becomes evident that among the 16 solutions proposed in [31], only 8 are feasible. This highlights the importance of carefully evaluating the validity of potential solutions mainly in terms of accuracy.

**Table 3.** The joint values $[J_i]$ for the 8 IK solutions to reach $Pose_A$ on CRX-10*i*A/L cobot.

| | ***J*1** | ***J*2** | ***J*3** | ***J*4** | ***J*5** | ***J*6** |
|---|---|---|---|---|---|---|
| $[J_{A\#1}]$ | 39.902 | 61.782 | 137.023 | 75.169 | 116.229 | 90.344 |
| $[J_{A\#2}]$ | 114.690 | 47.928 | 160.609 | 23.885 | 170.539 | −10.812 |
| $[J_{A\#3}]$ | 86.018 | −42.379 | 16.943 | −35.883 | −55.129 | 2.977 |
| $[J_{A\#4}]$ | 78 | −41 | 17 | −42 | − 60 | 10 |
| $[J_{A\#5}]$ | −140.098 | −61.782 | 42.977 | −104.831 | 116.229 | 90.344 |
| $[J_{A\#6}]$ | −65.310 | −47.928 | 19.391 | −156.116 | 170.539 | −10.812 |
| $[J_{A\#7}]$ | −93.982 | 42.379 | 163.057 | 144.118 | −55.129 | 2.977 |
| $[J_{A\#8}]$ | −102 | 41 | 163 | 138 | −60 | 10 |

- The second example concerns the CRX-10*i*A, with the following pose values:

$$Pose_B = [80.321, \quad 287.676, \quad 394.356, \quad -131.819, \quad -45.268, \quad 61.453]^T$$

The different sets of joint angles are given in Table 4. These eight solutions have been confirmed, first virtually on Roboguide software then implemented on the real cobot. Two of these solutions are represented in Figure 6a. The 1st line of Table 4 corresponds to the cobot posture represented in cyan, the 4th line corresponds to the one represented in black.

By comparing Tables 3 and 4, it is apparent that postures $[J_{A\#4}]$ and $[J_{B\#4}]$ are identical (as are dual postures $[J_{A\#8}]$ and $[J_{B\#8}]$). This outcome was expected since the desired poses were derived from line 4 of the tables, which served as the starting point for the analysis (giving *Desired-Poses*: $Pose_A$ and $Pose_B$).

**Table 4.** The joint values $[J_i]$ for the 8 IK solutions to reach $Pose_B$ on CRX-10*i*A.

| | ***J*1** | ***J*2** | ***J*3** | ***J*4** | ***J*5** | ***J*6** |
|---|---|---|---|---|---|---|
| $[J_{B\#1}]$ | 44.611 | 89.087 | 109.193 | 94.703 | 121.416 | 121.782 |
| $[J_{B\#2}]$ | 35.162 | 88.468 | 140.150 | −108.846 | −111.920 | −91.804 |
| $[J_{B\#3}]$ | 29.462 | −39.473 | −8.392 | 117.682 | 85.679 | −119.224 |
| $[J_{B\#4}]$ | 78 | −41 | 17 | −42 | − 60 | 10 |
| $[J_{B\#5}]$ | −135.389 | −89.087 | 70.807 | −85.297 | 121.416 | 121.782 |
| $[J_{B\#6}]$ | −144.839 | −88.468 | 39.850 | 71.154 | −111.920 | −91.804 |
| $[J_{B\#7}]$ | −150.538 | 39.473 | 188.392 | −62.318 | 85.679 | −119.224 |
| $[J_{B\#8}]$ | − 102 | 41 | 163 | 138 | − 60 | 10 |

### *3.2. Examples with 12 IK Solutions for CRX-10iA*

- We examine the following example, denoted as $Pose_C$, which yields 12 distinct IK solutions, see Table 5:

$$Pose_C = [600, \quad 0, \quad 100, \quad -180, \quad 0, \quad 70]^T$$

This example is particularly intriguing because, owing to its unique desired orientation, it can be solved manually without the need for inverse kinematics, relying simply on trigonometric relationships and DHm parameters.

The same end-effector pose could be achieved through 12 sets of joint angles, as shown in Table 5. It is important to note that values such as 78.001 or 179.999 are not are not rough estimates of the whole numbers 78 or 180, but rather precise solutions of the Inverse Kinematics, resolved to an accuracy of $10^{-3}$ degrees while taking into account joint limitations.

**Table 5.** The joint values $[J_i]$ for the 12 IK solutions to reach $Pose_C$ on CRX-10*i*A.

| | ***J*1** | ***J*2** | ***J*3** | ***J*4** | ***J*5** | ***J*6** |
|---|---|---|---|---|---|---|
| $[J_{C\#1}]$ | −14.478 | 119.780 | 78.001 | −180 | 168.001 | −95.522 |
| $[J_{C\#2}]$ | −6.336 | 121.233 | 89.999 | −116.197 | 179.999 | −39.861 |
| $[J_{C\#3}]$ | 6.335 | 121.233 | 90 | −63.811 | 179.999 | −0.146 |
| $[J_{C\#4}]$ | 14.478 | 119.780 | 78.001 | 0 | −168.001 | 55.522 |
| $[J_{C\#5}]$ | −14.478 | 11.999 | −29.780 | −180 | 60.220 | −95.522 |
| $[J_{C\#6}]$ | 14.478 | 11.999 | −29.780 | 0 | −60.220 | 55.522 |
| $[J_{C\#7}]$ | 165.522 | −119.780 | 101.999 | 0 | 168.001 | −95.522 |
| $[J_{C\#8}]$ | 173.664 | −121.233 | 90.001 | 63.811 | 179.999 | −39.861 |
| $[J_{C\#9}]$ | −173.665 | −121.233 | 90 | 116.190 | 179.999 | −0.146 |

**Table 5.** *Cont.*

| | *J1* | *J2* | *J3* | *J4* | *J5* | *J6* |
|---|---|---|---|---|---|---|
| $[J_{C\#10}]$ | −165.522 | −119.780 | 101.999 | −180 | −168.01 | 55.522 |
| $[J_{C\#11}]$ | 165.522 | −11.999 | −150.220 | 0 | 60.220 | −95.522 |
| $[J_{C\#12}]$ | −165.522 | −11.999 | −150.220 | 180 | −60.220 | 55.522 |

For this particular pose, the desired direction $Z_{tool}$ of the end-effector is vertical, see Figure 13a, a common case for many *pick-and-place* applications in robotics. This is reflected by the orientation values ($W = 180°$ and $P = 0°$) and by the common points of intersection between the two $\text{UP}(q)$ and $\text{DW}(q)$ curves, see Figure 13b (blue and red curves intersect the zero line at the same points).

This means that, for specific wrist configurations, there could be two solutions $\text{UP}(q)$ and $\text{DW}(q)$ where the configuration of the wrist in both solutions are precisely the same.

This phenomenon mirrors what occurs with a spherical wrist configuration. Consequently, we can observe identical values in lines #1 and #5 of Table 5 ($J1 = -14.478°$) and the exact opposite values in lines #4 and #6 ($J1 = +14.478°$).

Ensuring that Inverse Kinematic solutions are both reliable and accurate is essential. Equally important is the consideration of a robot's joint constraints. Our algorithm addresses this by incorporating joint limits into the solution process, the desired $Pose_C$ is such an illustration.

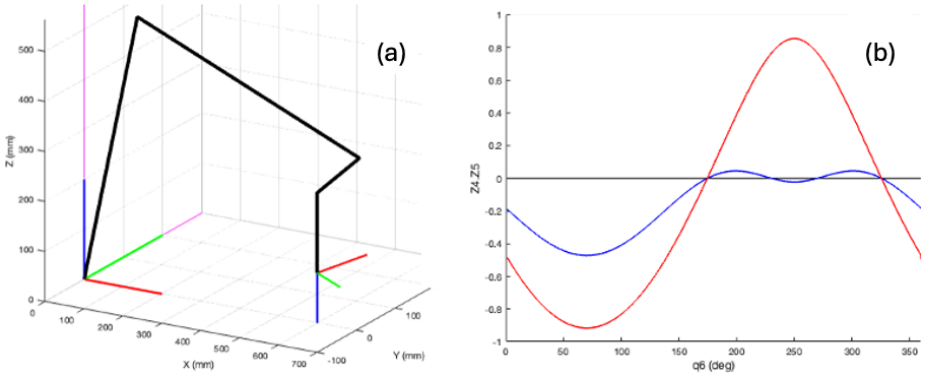


**Figure 13.** (**a**) Cobot posture corresponding to line 6 of Table 5. (**b**) The two $\text{UP}(q)$ and $\text{DW}(q)$ dot-products ($Z_4 \cdot Z_5$) corresponding to $Pose_C$.

### *3.3. Example with 16 IK Solutions for CRX-10iA*

- For this example, the pose parameters were obtained using FK from posture $[J_{D\#7}]$, where joint parameters are in whole degrees (as well as its dual posture, $[J_{D\#15}]$). This is why the values of the following $Pose_D$ are defined with a resolution of $10^{-3}$ mm and $10^{-3}$ deg.

$$Pose_D = [209.470, \quad -42.894, \quad 685.496, \quad -95.378, \quad -64.226, \quad -56.402]^T$$

This example demonstrates the existence of a *Desired-Pose*, with 16 solutions, as shown in Figure 14. The 7th image of this figure corresponds to the cobot posture represented, in black, in Figure 6b. All of the 16 configurations were verified using Roboguide software (version 9).

Table 6 displays the six joint parameters corresponding to each of these solutions. All these solutions have been tested on the real cobot.

In this example, the distance between the *Tool Center Point* and the vertical axis $Z_0$ of frame $R_0$ is 214 mm. This point is located near the center of the cobot's workspace, away from the outer boundary. This characteristic is common to all poses with 16 IK solutions.

**Table 6.** The joint values $[J_i]$ for the 16 IK solutions to reach $Pose_D$ on CRX-10*i*A.

| | ***J*1** | ***J*2** | ***J*3** | ***J*4** | ***J*5** | ***J*6** |
|---|---|---|---|---|---|---|
| $[J_{D\#1}]$ | −60.125 | 62.707 | 112.015 | 90.165 | 92.586 | 132.291 |
| $[J_{D\#2}]$ | −63.318 | 62.684 | 143.064 | −93.111 | −89.750 | −78.691 |
| $[J_{D\#3}]$ | 11.855 | 54.151 | 144.007 | −28.773 | −142.889 | −48.616 |
| $[J_{D\#4}]$ | 49.247 | 46.825 | 135.954 | 29.379 | −134.512 | −5.010 |
| $[J_{D\#5}]$ | −62.156 | −40.094 | 14.333 | 92.039 | 91.458 | −129.964 |
| $[J_{D\#6}]$ | 47.115 | −53.924 | 35.922 | 28.105 | −41.924 | −48.121 |
| $[J_{D\#7}]$ | 0 | −45 | 44 | −37 | −53 | 0 |
| $[J_{D\#8}]$ | −47.369 | −40.310 | 45.272 | −78.410 | −81.969 | 17.952 |
| $[J_{D\#9}]$ | 119.875 | −62.707 | 67.985 | −89.835 | 92.586 | 132.291 |
| $[J_{D\#10}]$ | 116.682 | −62.684 | 36.936 | 86.889 | −89.750 | −78.691 |
| $[J_{D\#11}]$ | −168.145 | −54.151 | 35.993 | 151.227 | −142.889 | −48.616 |
| $[J_{D\#12}]$ | −130.753 | −46.825 | 44.046 | −150.621 | −134.512 | −5.01 |
| $[J_{D\#13}]$ | 117.844 | 40.094 | 165.667 | −87.961 | 91.458 | −129.964 |
| $[J_{D\#14}]$ | −132.885 | 53.924 | 144.078 | −151.895 | −41.924 | −48.121 |
| $[J_{D\#15}]$ | −180 | 45 | 136 | 143 | −53 | 0 |
| $[J_{D\#16}]$ | 132.631 | 40.310 | 134.728 | 101.590 | −81.969 | 17.952 |

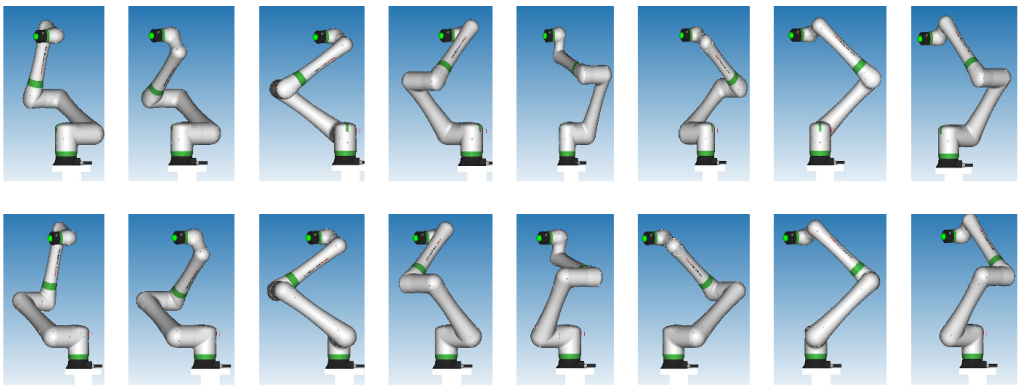

**Figure 14.** Visualizations of 16 distinct postures to access the same $Pose_D$. $Pose_D = [209.470, -42.894, 685.496, -95.378, -64.226, -56.402]^T$.

## 4. Discussion and Conclusions

In this paper, we discussed the CRX cobots series introduced by FANUC. After emphasizing the significance of cobotics in various applications, we proceeded to develop both Forward and Inverse Kinematic models for CRX cobots.

The Forward Kinematic model that we have introduced adheres closely to the FANUC approach. It is a model designed to ascertain the pose delivered by the cobot for a given configuration, represented by the six programmed $Ji$ values.

This implies that the model developed accurately incorporates the origins and directions of the joint values, as well as the specific coupling between the $J2$ and $J3$ joints, in the Forward Kinematics calculations.

We have demonstrated the potential utility of the differential approach for joint control under 6D path-planning constraints in certain specific scenarios. It is nevertheless crucial to emphasize the necessity of a complete and reliable Inverse Kinematic solution.

While previous solutions have offered partial sets of equations, notably the one proposed by FANUC in Roboguide software, and others have exhibited inconsistencies, this

study provides a numerical geometric approach for the inverse kinematic solution for cobotic arms with non-spherical wrists, specifically the CRX-10*i*A, with a particular focus on completeness and accuracy.

This novel approach

- Enables the determination of the exact number of possible joint solutions for a desired pose.
- Provides the different sets of joint angles values $[J_i]$ for these distinct solutions.

Through Forward and Inverse kinematic models computing, simulations on the Roboguide software, and experiments on the real cobot CRX-10*i*A, we have demonstrated that the number of IK solutions can range from 2 to 16. In most cases, 8 IK solutions are valid. However, IK solutions may also occur in increments of 4, 8, 12, or 16, with occasional instances of 2, 6, 10, or 14 solutions.

The technique we have developed has been experimentally validated and proven to be robust. It accurately determines the number of IK solutions corresponding to the actual valid solutions within joint limits, ensuring neither more nor fewer solutions are provided. Moreover, this method provides joint angle values with sub-millidegree accuracy in a few milliseconds, so operates rapidly and efficiently from a computational point of view.

This approach can be applied to all 6R serial robots with a non-spherical wrist. For example, as part of a comparative analysis, we tested it on the Universal Robot, UR5, for which, however, an explicit IK model can be established. The results provided were consistent with the literal IK.

In the continuation of this work on CRX cobots series, we will aim to characterize workspace domains with a specific number of IK solutions and define their boundaries, particularly when this number exceeds eight. Additionally, we will explore the issue of crossing these boundaries, for instance, when transitioning from a 16-solution domain to one with a 12-solution domain, and vice versa.

We will also study the concept of "aspect" and changes in aspect, examining the link between aspect and p-solution domains. As mentioned above, we will explore the issue of transitioning between aspects and within a given aspect.

Figure 15 highlights a particular example of this future work: transitioning from one solution to another within the same set of solutions (from $[J_{D\#2}]$ to $[J_{D\#4}]$), while maintaining the same initial and final poses, without encountering any singularities during the movement.

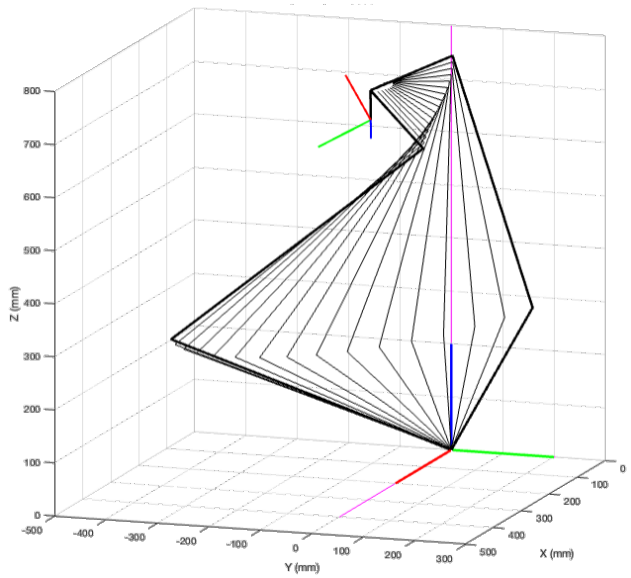


**Figure 15.** Envelope of cobot postures, representing the displacement between two distinct solutions (from $[J_{D\#2}]$ to $[J_{D\#4}]$ of Table 6), in the case where there are 16 IK solutions for the same pose.

**Author Contributions:** Both authors M.A. and G.P. contributed identically to all items of this publication. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All data are contained within this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| DHm | Denavit–Hartenberg modified |
| DoF | Degree-of-Freedom |
| FK | Forward Kinematics |
| IK | Inverse Kinematics |
| MRI | Magnetic Resonance Imaging |
| MSD | Musculoskeletal Disorders |
| TCP | Tool Center Point |
| UR | Universal Robots |
| XYZ | End-effector position |
| WPR | Cardan's angles |

## References

1. Ginoya, T.; Maddahi, Y.; Zareinia, K. A historical review of medical robotic platforms. *J. Robot.* **2021**, *2021*, 6640031. [CrossRef]
2. Morgan, A.A.; Abdi, J.; Syed, M.A.; Kohen, G.E.; Barlow, P.; Vizcaychipi, M.P. Robots in healthcare: A scoping review. *Curr. Robot. Rep.* **2022**, *3*, 271–280. [CrossRef] [PubMed]
3. Weidemann, C.; Mandischer, N.; van Kerkom, F.; Corves, B.; Hüsing, M.; Kraus, T.; Garus, C. Literature Review on Recent Trends and Perspectives of Collaborative Robotics in Work 4.0. *Robotics* **2023**, *12*, 84. [CrossRef]
4. Baumkircher, A.; Seme, K.; Munih, M.; Mihelj, M. Collaborative robot precision task in medical microbiology laboratory. *Sensors* **2022**, *22*, 2862. [CrossRef] [PubMed]
5. Diab, J. Hybrid Robotic Control by Teleoperation and Comanipulation: Application to Ultrasound Probe Positioning in a Constrained Environment. Ph.D. Thesis, Université d'Orléans, Orléans, France, 2021.
6. Moutsinga, I.; Paccot, F.; Chanal, H.; Bouton, N. Co-manipulation robotique transparente pour l'assistance à l'examen échographique. In Proceedings of the 17ème Colloque National S-mart AIP-PRIMECA, Université Polytechnique Hauts-de-France [UPHF], Laval, France, 31 March–2 April 2021.
7. Abbes, M.; Belharet, K.; Mekki, H.; Poisson, G. Use of the CRX-10*i*A cobot for microparticles delivery inside the cochlea. In Proceedings of the 12th Conference on New Technologies for Computer/Robot Assisted Surgery (CRAS), Paris, France, 11–13 September 2023.
8. Abbes, M.; Belharet, K.; Mekki, H.; Poisson, G. Permanent magnets based actuator for microrobots navigation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 7062–7067.
9. Abbes, M.; Belharet, K.; Souissi, M.; Mekki, H.; Poisson, G. Design of a Robotized Magnetic Platform for Targeted Drug Delivery in the Cochlea. *IRBM* **2023**, *44*, 100728. [CrossRef]
10. Pieper, D.L. *The Kinematics of Manipulators under Computer Control*; Stanford University: Stanford, CA, USA, 1969.
11. Paul, R.P. *Robot Manipulators: Mathematics, Programming, and Control: The Computer Control of Robot Manipulators*; MIT Press: Cambridge, MA, USA, 1981.
12. Andersen, R.S. *Kinematics of a UR5*; Aalborg University: Aalborg, Denmark, 2018.
13. Whitney, D.E. Resolved motion rate control of manipulators and human prostheses. *IEEE Trans. Man-Mach. Syst.* **1969**, *10*, 47–53. [CrossRef]
14. Fournier, A. Génération de Mouvements en Robotique, Applications des Inverses Généralisées et des Pseudo-Inverses. Ph.D. Thesis, Université des Sciences et Techniques du Languedoc, Montpellier, France, 1980.
15. Wolovich, W.A.; Elliott, H. A computational technique for inverse kinematics. In Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, NV, USA, 12–14 December 1984; pp. 1359–1363.
16. Sciavicco, L.; Siciliano, B. Coordinate transformation: A solution algorithm for one class of robots. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 550–559. [CrossRef]
17. Manseur, R.; Doty, K.L. A fast algorithm for inverse kinematic analysis of robot manipulators. *Int. J. Robot. Res.* **1988**, *7*, 52–63. [CrossRef]
18. Manseur, R.; Doty, K.L. Structural kinematics of 6-revolute-axis robot manipulators. *Mech. Mach. Theory* **1996**, *31*, 647–657. [CrossRef]

19. Manseur, R.; Doty, K.L. A robot manipulator with 16 real inverse kinematic solution sets. *Int. J. Robot. Res.* **1989**, *8*, 75–79. [CrossRef]
20. Raghavan, M.; Roth, B. A general solution for the inverse kinematics of all series chains. In Proceedings of the 8th CISM-IFTOMM Symposium on Robots and Manipulators, Cracow, Poland, 2–5 July 1990.
21. Raghavan, M.; Roth, B. Inverse kinematics of the general 6R manipulator and related linkages. *Trans. ASME J. Mech. Des.* **1993**, *115*, 502–508. [CrossRef]
22. Manocha, D.; Canny, J.F. Real time inverse kinematics for general 6R manipulators. In Proceedings of the ICRA, Nice, France, 12–14 May 1992, pp. 383–389.
23. Manocha, D.; Canny, J.F. Efficient inverse kinematics for general 6R manipulators. *IEEE Trans. Robot. Autom.* **1994**, *10*, 648–657. [CrossRef]
24. Gómez, S.R.; Cerrada, J.A.; Feliu, V. A faster algorithm for calculating the inverse kinematics of a general 6R manipulator for robot real time control. *IFAC Proc. Vol.* **1999**, *32*, 833–838. [CrossRef]
25. Liqing, N.; Qingjiu, H. Inverse kinematics for 6-DOF manipulator by the method of sequential retrieval. In Proceedings of the 1st International Conference on Mechanical Engineering and Material Science (MEMS 2012), Paris, France, 29 January–2 February 2012; Atlantis Press: Amsterdam, The Netherlands, 2012; pp. 600–603.
26. Almusawi, A.R.; Dülger, L.C.; Kapucu, S. A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242). *Comput. Intell. Neurosci.* **2016**, *2016*, 5720163. [CrossRef] [PubMed]
27. Li, J.; Yu, H.; Shen, N.; Zhong, Z.; Lu, Y.; Fan, J. A novel inverse kinematics method for 6-DOF robots with non-spherical wrist. *Mech. Mach. Theory* **2021**, *157*, 104180. [CrossRef]
28. Huang, B.; Milenkovic, V. Kinematics of major robot linkages. In Proceedings of the 13th International Symposium on Industrial Robots, Chicago, IL, USA, 17–21 April 1983; pp. 17–19.
29. Aristidou, A.; Lasenby, J. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graph. Model.* **2011**, *73*, 243–260. [CrossRef]
30. Kucuk, S.; Bingul, Z. Inverse kinematics solutions for industrial robot manipulators with offset wrists. *Appl. Math. Model.* **2014**, *38*, 1983–1999. [CrossRef]
31. Carbonari, L.; Palpacelli, M.C.; Callegari, M. Inverse Kinematics of a Class of 6R Collaborative Robots with Non-Spherical Wrist. *Robotics* **2023**, *12*, 36. [CrossRef]
32. Thomas, F.; Porta, J. The inverse kinematics of lobster arms. *Mech. Mach. Theory* **2024**, *196*, 105630. [CrossRef]
33. Khalil, W.; Kleinfinger, J. A new geometric notation for open and closed-loop robots. In Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 7–10 April 1986; Volume 3, pp. 1174–1179.
34. Denavit, J.; Hartenberg, R.S. A kinematic notation for lower-pair mechanisms based on matrices. *J. Appl. Mech.* **1955**, *22*, 215–221. [CrossRef]