Analysis of computational efficiency for the solution of inverse kinematics problem of anthropomorphic robots using Gröbner bases theory

Sérgio Ricardo Xavier da Silva¹, Leizer Schnitman¹ and Vitalino Cesca Filho²

Abstract
This article presents an analysis of computational efficiency to solve the inverse kinematics problem of anthropomorphic robots. Two approaches are investigated: the first approach uses Paul’s method applied to the matrix obtained by the Denavit–Hartenberg algorithm and the second approach uses Gröbner bases theory. With each approach, the problem of inverse kinematics for an anthropomorphic robot will be solved. When comparing each method, this article will demonstrate that the method using Gröbner bases theory is more computationally efficient.

Keywords
Robotics, anthropomorphic robots, inverse kinematics, computational efficiency, Paul’s method, Gröbner bases theory

Introduction
Two classic problems in the kinematics of robotic manipulators are forward kinematics (FK) and inverse kinematics (IK). In FK, the position and orientation of the end-effector are calculated from the joint variables. In the IK, the movements of individual joints are calculated from the position and orientation of the end-effector of a robotic manipulator.

According to Daya et al.,¹ one of the biggest problems in kinematics and robot control lies in determining the solution of IK. Still, according to the same authors, as the complexity of the robotic manipulator increases, obtaining the IK is difficult and computationally expensive. Traditional methods, such as geometric, iterative, and algebraic, are inadequate as the geometric structure of the manipulator becomes more complicated.

Paul et al.² used matrix algebra to obtain a solution to the IK problem of a specific manipulator robot. Lee³ and Yih and Youm⁴ used a geometric approach to solve this problem. Subsequently, Huang and Milenkovic⁵ and Mansur and Doty⁶ developed iterative procedures to solve the IK problem.

Denavit and Hartenberg⁷ introduced a convention to standardize the reference coordinate systems for spatial links; after a decade, the same authors presented an algorithm for solving the kinematics problems of articulated
systems. The importance of this algorithm for the kinematics analysis of robotic systems was studied by Paul 16 years ago.

An alternative method for the solution of IK will be used in this article, namely the Gröbner bases theory. The computational performance data obtained in this article will prove that this choice will reduce the computational cost for the solution of the IK problem of serial robotic manipulators compared to Paul’s method.

The Gröbner bases theory, developed by Bruno Buchberger in his PhD thesis, consists of an algebraic algorithm that can be applied to a given set of nonzero polynomials, producing a finite set of generators, so that it is possible to identify from these generators when a specific polynomial belongs to the original given set. When finding solutions for this set of generators, those are also solutions for the whole set. With the aid of MAPLE, an algebraic computing software, all these solutions can be calculated, including cases not so easily solved using matrix algebra or geometric applications due to the complexity of the geometry of some manipulator robots.

Kendricks, in her PhD thesis, compared the algorithm of Denavit and Hartenberg with Gröbner bases theory for the solution of the IK problem. Her analysis focused on the robot Fanuc A-510, a SCARA type robot, with three degrees of freedom.

To familiarize the reader with each method, the anthropomorphic robot Unimation PUMA 560, a manipulator with six rotary joints, will be used as a case study applying Paul’s method from the Denavit–Hartenberg matrix and the method proposed in this work to solve the IK problem. The main goal of this article is to analyze the computational efficiency in solving the problem of the IK of anthropomorphic robots using Grobner bases theory compared to Paul’s method.

briefly, a Grobner basis for a polynomial equation system is a new collection of polynomials in which the null set coincides with the solution set of the original system; however, this new system is simpler to solve. This definition justifies the choice of Gröbner bases theory to solve the IK of serial manipulators with the reduction of computational cost.

Grobner bases theory is widely applied in other areas besides mathematics. For example, Cox et al. have dedicated their work to the applications of Gröbner bases theory in geometry, graph theory, and robotics.

According to Sturmfels, a Gröbner basis can be computed in order that the polynomials have a property known as the elimination property. This elimination property makes the computation of solutions of a polynomial system, which is a Gröbner basis to be mathematically efficient because, through retroactive substitutions, a solution can be found for the remaining set of variables, with a reduction in the number of floating-point operations.

For polynomials with several variables, the complexity for the determination of a Gröbner basis depends not only on the number of polynomials in the set but also on how their terms are ordered. The lexicographic order is particularly suitable for solving systems of polynomial equations. In a lexicographic order, a variable dominates any monomial involving only minor variables, regardless of the total

increase the computational efficiency of the calculations to obtain all the joint configurations.

El-Sherbiny et al. presented a comparative study between different soft-computing-based methods (artificial neural network, adaptive neuro-fuzzy inference system, and genetic algorithms) applied to the problem of IK. Their analysis focused on a serial manipulator with five degrees of freedom, and the results of the analysis demonstrate that the methods are not accurate and computationally efficient at the same time. The choice of method depends on the application, being necessary to decide which one has more priority, the calculation time or the accuracy.

However, the present work intends to go further. It brings an analysis of the computational efficiency in the solution of the IK problem of a serial manipulator with six degrees of freedom using Gröbner bases theory. The obtained results will be compared with those obtained by Paul’s method from the matrix obtained by the Denavit–Hartenberg algorithm, demonstrating that the Gröbner bases theory method is more computationally efficient and that the solutions obtained from this method are accurate.

Gröbner bases theory

The core of Gröbner bases theory is an algorithm used to solve systems of polynomial equations. It is known as Buchberger’s algorithm and applies computational algebra techniques to polynomial ideals, producing a so-called Gröbner basis, a new polynomial set.

Briefly, a Gröbner basis for a polynomial equation system is a new collection of polynomials in which the null set coincides with the solution set of the original system; however, this new system is simpler to solve. This definition justifies the choice of Gröbner bases theory to solve the IK of serial manipulators with the reduction of computational cost.

Gröbner bases theory is widely applied in other areas besides mathematics. For example, Cox et al. have dedicated their work to the applications of Gröbner bases theory in geometry, graph theory, and robotics.

According to Sturmfels, a Gröbner basis can be computed in order that the polynomials have a property known as the elimination property. This elimination property makes the computation of solutions of a polynomial system, which is a Gröbner basis to be mathematically efficient because, through retroactive substitutions, a solution can be found for the remaining set of variables, with a reduction in the number of floating-point operations.

For polynomial equations with several variables, the complexity for the determination of a Gröbner basis depends not only on the number of polynomials in the set but also on how their terms are ordered. The lexicographic order is particularly suitable for solving systems of polynomial equations. In a lexicographic order, a variable dominates any monomial involving only minor variables, regardless of the total
degree. For example, for the lexicographic order \( x \succ y \succ z \), thus \( x^2 >_{\text{lex}} y^3 z^8 \).

It can be seen in Figure 1 that all variables can be solved consecutively because the obtained Gröbner basis is a triangular equation system.

In some cases, one chooses to consider the total degree of a monomial and to order the monomials of larger degrees first. In this specific case, the graduated lexicographical order (grlex) is used. Another order is given by the reverse graduated lexicographical order (grevlex), which is commonly used because it generally provides the basis more quickly due to a reduction in computational cost.

Buchberger’s algorithm constructs a Gröbner basis starting from a finite set of polynomials. According to Cox et al., to develop Buchberger’s algorithm, it is necessary to understand the nature of S-polynomials.

Let \( f \) and \( g \) be two nonzero polynomials and consider \( L = \text{lcm}(\text{lt}(f), \text{lt}(g)) \), where \( \text{lcm} \) is the least common multiple among the leading power products (lp) of polynomials. The combination \( S(f, g) = \frac{l_f}{\text{lt}(f)} f - \frac{l_g}{\text{lt}(g)} g \) is called the S-polynomial of \( f \) and \( g \), where (lt) stands for the leading term of each polynomial.

As an example, \( f = x^3 y^2 - x^2 y^3 + x \), \( g = 3 x^4 y + y^2 \in \mathbb{R}[x, y] \), with lexicographical order \( x \succ y \), it is intended to determine: \( S(f, g) \). To solve this problem, it is necessary to determine \( L = \text{lcm}(\text{lp}(f), \text{lp}(g)) = \text{lcm}(x^3 y^2, x^4 y^2) \).

Thus

\[
S(f, g) = \frac{x^4 y^2}{x^3 y^2} f - \frac{x^4 y^2}{3 x^4 y} g
\]

(1)

\[
S(f, g) = x(x^3 y^2 - x^2 y^3 + x) - \frac{y}{3} (3 x^4 y + y^2)
\]

(2)

\[
S(f, g) = -x^3 y^3 + x^2 - \frac{y^3}{3}
\]

(3)

Note that the S-polynomial is produced to cancel the leading terms of the polynomials involved. The Buchberger’s algorithm receives as input a finite set \( F = \{ f_1, f_2, \cdots, f_m \} \) of polynomials and provides a set \( G \), which is a Gröbner basis of \( F \). According to Stifter, this algorithm can be described as follows.

**Algorithm 1. Algorithm for computing Gröbner bases.**

Input: \( F \), a finite subset of \( \mathbb{K}[x_1, \ldots, x_n] \).

Output: \( G = \text{GB}(F) \).

1. \( G = F \)
2. for each pair \( (f, g) \) of elements in \( G \) do
   - \( h = S(f, g) \)
   - \( h \) is normal form of \( h \)
   - if \( h \neq 0 \) then \( G = G \cup \{ h \} \)

Faugère presented an efficient algorithm for computing Gröbner bases: the F4 algorithm. This algorithm computes successive truncated Gröbner bases, and it replaces the classical polynomial reduction found in the Buchberger’s algorithm by the simultaneous reduction of several polynomials. This powerful reduction mechanism utilizes symbolic pre-computation and sparse linear algebra methods.

With the help of algebraic computing software, a Gröbner basis can be calculated more quickly. Many algebraic computing systems include packages to deal with Gröbner bases, for example, MAPLE, Singular, muMath, AXIOM, Mathematica, Macaulay, CoCoA, and Reduce.

**Inverse kinematics using Paul's method**

The PUMA 560 is a robotic manipulator with six degrees of freedom and has only rotary joints, that is, it is a 6R mechanism. According to Ghosal, PUMA 560 is a robotic manipulator extensively seen in textbooks, industry, and academic and research institutions.

The PUMA 560 is shown in Figure 2, with its six frames attached to the links.

It can be seen that the reference system \{0\}, not shown in Figure 2, coincides with the reference system \{1\} when \( \theta_1 \) is null. It is also noted that for the Puma 560 manipulator, as for other industrial robots, the axes of joints 4, 5, and 6 intersect at a common point, and that point of intersection coincides with the origin of frames \{4\}, \{5\}, and \{6\}, the axes of these three joints being mutually orthogonal.

According to Craig, to determine a general transformation matrix \( T \) for the solution of the FK of the Puma 560
manipulator, it is necessary to construct a transformation that defines the system \( \{i\} \) in relation to the system \( \{i-1\} \),

\[
T = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 = 
\begin{bmatrix}
    c_1 & -s_1 & 0 & 0 \\
    s_1 & c_1 & 0 & 0 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix} \quad 
\begin{bmatrix}
    c_2 & -s_2 & 0 & 0 \\
    s_2 & c_2 & 0 & 0 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix} \quad 
\begin{bmatrix}
    c_3 & -s_3 & 0 & a_2 \\
    s_3 & c_3 & 0 & 0 \\
    0 & 0 & 1 & d_3 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

that is, it is intended to determine \( i^{-1} A_i \). By defining a reference system for each link, the kinematic problem is divided into \( n \) subproblems, and to solve them, each subproblem is further broken up into four others.

Initially, three intermediate frames \( \{P\} \), \( \{Q\} \), and \( \{R\} \) are defined for each link of the manipulator. It can be seen in Figure 3 that the reference system \( \{R\} \) differs from the reference system \( \{i-1\} \) only by a rotation of \( \alpha_{i-1} \), the reference system \( \{Q\} \) differs from \( \{R\} \) by a translation \( a_{i-1} \), the reference system \( \{P\} \) differs from \( \{Q\} \) by a rotation \( \theta_i \), and the reference system \( \{i\} \) differs from \( \{P\} \) by a translation \( d_i \).

Thus

\[
i^{-1} A_i = R_{x, \alpha_{i-1}} \cdot T_{z, \theta_i} \cdot R_{z, \theta_i} \cdot T_{z, d_i} \\
= \begin{bmatrix}
    c_{\theta_i} & -s_{\theta_i} & 0 & a_{i-1} \\
    s_{\theta_i} c_{\alpha_{i-1}} & c_{\theta_i} c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -s_{\alpha_{i-1}} d_i \\
    s_{\theta_i} s_{\alpha_{i-1}} & c_{\theta_i} s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & c_{\alpha_{i-1}} d_i \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

It is used \( s_\theta \) for \( \sin(\theta) \) and \( c_\theta \) for \( \cos(\theta) \).

All Denavit–Hartenberg parameters of this manipulator are listed in Table 1, where \( i \) represents the joint number.

After constructing all six transformation matrices using the Denavit–Hartenberg parameters and multiplying them (5), the homogeneous transformation matrix (6) is obtained

**Table 1.** Denavit–Hartenberg parameters of Puma 560.

| \( i \) | \( \alpha_{i-1} \) (rad) | \( a_{i-1} \) (mm) | \( d_i \) (mm) | \( \theta_i \) (rad) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | \( \theta_1 \) |
| 2 | \( \frac{-\pi}{2} \) | 0 | 0 | \( \theta_2 \) |
| 3 | 0 | \( a_2 \) | \( d_3 \) | \( \theta_3 \) |
| 4 | \( \frac{-\pi}{2} \) | \( a_3 \) | \( d_4 \) | \( \theta_4 \) |
| 5 | \( \frac{-\pi}{2} \) | 0 | 0 | \( \theta_5 \) |
| 6 | \( \frac{-\pi}{2} \) | 0 | 0 | \( \theta_6 \) |

![Figure 2](image2.png)

**Figure 2.** Establishing coordinate frames for the Puma 560. Source: Adapted from Chakravarti and Sivakumar.\(^{22}\)

![Figure 3](image3.png)

**Figure 3.** Location of intermediate frames \( \{P\} \), \( \{Q\} \), and \( \{R\} \). Source: Adapted from Craig.\(^{23}\)
where the entries of $T$ are given by
\begin{align}
n_x &= c_1(c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_6) + s_1(s_4c_{23}c_6 + c_4s_6) \\
n_y &= s_1(c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_6) + c_1(s_4c_5c_6 + c_4s_6) \\
n_z &= -s_{23}(c_4c_5c_6 - s_4s_5) - c_{23}s_5c_6 \\
a_x &= c_1(c_{23}(-c_4s_5c_6 - s_4c_6) + s_{23}s_5s_6) + s_1(c_4c_6 - s_4c_5s_6) \\
a_y &= s_1(c_{23}(-c_4s_5c_6 - s_4c_6) + s_{23}s_5s_6) - c_1(c_4c_6 - s_4c_5s_6) \\
a_z &= -s_{23}(-c_4s_5c_6 - s_4c_6) + c_{23}s_5c_6 \\
a_x &= -c_1(c_{23}c_5s_6 + s_{23}c_5) - s_1s_4s_5 \\
a_y &= -s_1(c_{23}c_5s_6 + s_{23}c_5) + c_1s_4s_5 \\
a_z &= s_{23}c_4s_5 - c_{23}c_5 \\
x &= c_1(a_2c_2 + a_1c_{23} - d_4s_23) - d_3s_1 \\
y &= s_1(a_2c_2 + a_3c_{23} - d_4s_23) + d_3c_1
\end{align}

It is used $s_{23}$ for $\sin(\theta_2 + \theta_3)$ and $c_{23}$ for $\cos(\theta_2 + \theta_3)$.

According to Craig, for the solution of IK of a robotic manipulator with six degrees of freedom, there are 12 equations and 6 unknowns. However, among the nine equations that arise from the portion of the rotational matrix of $T$, only three are independent. These, added to the equations of the portion of the position vector of $T$, result in six equations with six unknowns. Such equations are non-linear, transcendental, which can be quite difficult to solve.

The systematic approach to solve kinematic equations, proposed by Paul, rewrites FK equations in several different ways. Thus, both sides of the FK matrix equation are left and right multiplied by inverse transformation matrices. In doing so, one gets equivalent equations having the same solutions. Each matrix equation provides 12 non-linear equations from the three main matrix lines. The fourth line of the matrix is always trivial. The main problem is to find, in this universe, suitable equations that can be solved analytically.

The solution of IK of this robotic manipulator, as determined by Craig, is given by equations to

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}\left(d_3, \pm \sqrt{x^2 + y^2 - d_3^2}\right)$$

$$\theta_3 = \text{Atan2}(a_3, d_4) - \text{Atan2}\left(\pm \sqrt{\frac{a_3^2 + d_4^2 - \left(\frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}\right)^2}{2a_2}}, \left(\frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}\right)^2\right)$$

$$\theta_2 = \text{Atan2}\left((-a_3 - a_2c_3)z - (c_1x + s_1y)(d_4 - a_2s_3), (a_2s_3 - d_4)z - (a_3 + a_2c_3)(c_1x + s_1y)\right) - \theta_3$$

$$\theta_4 = \text{Atan2}\left((-a_4s_1 + a_4c_1), (-a_4c_1c_{23} - a_4s_1c_{23} + a_4s_{23})\right)$$

$$\theta_4' = \theta_4 + \pi$$

$$\theta_5 = \pm \text{Atan2}\left((-a_4(c_1c_{23}c_4 + s_1s_4) + a_6(s_1c_{23}c_4 - c_1s_4) - a_2(s_23c_4), (a_4(-c_1s_{23}) + a_6(-s_1s_{23}) + a_6(-c_{23}))\right)$$

$$\theta_6 = \text{Atan2}\left((-n_x(c_1c_{23}s_4 - s_1c_4) - n_y(s_1c_{23}s_4 + c_1c_4) + n_z(s_23s_4)), (n_x((c_1c_{23}c_4 + s_1s_4)c_5 - c_1s_{23}s_5) + n_y((s_1c_{23}c_4 - c_1s_4)c_5) - s_1s_{23}s_5), n_z(c_4s_{23}c_5 + s_1c_{23}c_4s_5)\right)$$

$$\theta_6' = \theta_6 + \pi$$
All the equations of the FK and IK of the PUMA 560 manipulator were deduced in detail by Craig\textsuperscript{23} and Ghosal.\textsuperscript{21}

The tree structure in Figure 4 represents the order of computations to determine all eight joint configurations of the PUMA 560.

To measure the computational cost to determine all joint configurations to position and orient the end-effector arbitrarily in the space, the following joint values:

\[ q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{pmatrix} \] \[ \approx \begin{pmatrix} \frac{\pi}{4} \\ \frac{\pi}{4} \\ 0 \\ \frac{\pi}{4} \\ \frac{\pi}{4} \\ \frac{\pi}{4} \end{pmatrix} \] \[ \text{rad} \]

and the nonzero parameters:

\[ a_2 = 432, \quad a_3 = 20, \quad d_3 = 125, \quad \text{and} \quad d_4 = 430 \]

were replaced in equation (5), thus obtaining equation (27):

\[ T = \begin{bmatrix} 1 & 0 & 0 & \frac{125}{2} + 215\sqrt{3} \\ 0 & -1 & 0 & 215 - \frac{125\sqrt{3}}{2} \\ 0 & 0 & -1 & -452 \\ 0 & 0 & 0 & 1 \end{bmatrix} \] \[ \text{(27)} \]

From this point, equations (7) to (18) have been rewritten by substituting \( n_1, n_2, o_x, o_y, o_z, a_x, a_y, a_z, x, y, \) and \( z \) by the respective values of the obtained numerical matrix.

The Solve command in MAPLE is used to find all the values of \( \theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] \), which allow the robot to assume the configuration given by equation (27).

Performing 34,514 floating-point operations (2409 additions, 4261 multiplications, 175 divisions, and 3393 functions), eight different joint configurations are obtained, as given in Table 2.

This article will prove that using Gröbner bases theory, the computational cost to determine the solution of the IK of serial manipulators is considerably reduced.

**Inverse kinematics using Gröbner bases theory**

The IK problem consists of finding all combinations of joint configurations that will position the end-effector of a robot at a given point in space.\textsuperscript{23} To solve this problem, one must determine polynomial equations that model the movement of the robotic arm in each joint configuration. With all these elements, one must seek a set of solutions from the possible movements of each joint.

The joint values are set to \( \theta = \left[ -\frac{5\pi}{6}, \frac{\pi}{2}, 0, \frac{\pi}{2}, 0, \frac{\pi}{2} \right] \) rad, and the parameters are set to \( a_2 = 432, \ a_3 = 20, \ d_3 = 125, \) and \( d_4 = 430 \) in equation (5), exactly as in the third section, to determine all joint configurations to position and orient the end-effector of PUMA 560 robotic manipulator in space.

From this point, equations (7) to (18) are rewritten by substituting \( n_1, n_2, o_x, o_y, o_z, a_x, a_y, a_z, x, y, \) and \( z \) by the respective values of the matrix (27). These 12 polynomials, added to the six polynomials obtained from the fundamental trigonometric relation for each joint angle, are used to generate a Gröbner basis.
The selected lexicographic order for this robot is the one shown in equation (28). This order represents the same sequence in which the variables were solved while applying Paul’s method presented in the third section:

\[ c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1 \]  

(28)

The same values assigned in the third section are used, intending to compare the computational efficiency to solve the IK problem using Paul’s method from the Denavit–Hartenberg matrix and Gröbner bases theory of the same 6R manipulator.

Performing 466 floating-point operations (46 additions, 100 multiplications, and 40 functions), the following Gröbner basis consisting of 12 polynomials is obtained using Faugère’s F4 algorithm:

\[ g_1 = 8600\sqrt{3}s_1 + 32084s_1^2 + 4300\sqrt{3} + 2500s_1 - 6771 \]  

(29)

\[ g_2 = -8021\sqrt{3}s_1 - 1250\sqrt{3} + 5521c_1 - 8600s_1 - 4300 \]  

(30)

\[ g_3 = 1853s_1^2 + 172s_3 \]  

(31)

\[ g_4 = 2c_3 - 43s_3 - 2 \]  

(32)

\[ g_5 = 18578641250s_1s_3\sqrt{3} + 862257500\sqrt{3}s_1 \]
\[ + 9289320625\sqrt{3}s_3 + 63910525900s_1s_3 \]
\[ + 4311287750\sqrt{3} + 2966165800s_1 \]
\[ + 2148795284s_2 + 995987500s_3 - 665712384 \]  

(33)

\[ g_6 = -226593250\sqrt{3}s_1 - 113296625\sqrt{3} + 537198821c_2 \]
\[ - 779480780s_1 - 5780183345s_3 - 389740390 \]  

(34)

\[ g_7 = s_4 \]  

(35)

\[ g_8 = c_4^2 - 1 \]  

(36)

\[ g_9 = 43313400c_4s_5s_1\sqrt{3} - 431128750s_1c_4\sqrt{3} \]
\[ + 216567000c_4s_3\sqrt{3} + 1489980960c_4s_5s_1 \]
\[ - 215564375\sqrt{3}c_4 - 1483082900s_1c_4 \]
\[ + 1182126484c_4s_3 + 332856192c_4 + 1074397642s_5 \]  

(37)

\[ g_{10} = -4656190500s_3s_3\sqrt{3} - 226593250\sqrt{3}s_3 \]
\[ - 2328095250\sqrt{3}s_3 - 16017295320s_1s_3 \]
\[ - 113296625\sqrt{3} + 537198821c_5 \]
\[ - 779480780s_1 - 226636864s_3 - 389740390 \]  

(38)

\[ g_{11} = -c_4s_1 + s_6 \]  

(39)

\[ g_{12} = -8021s_1c_4\sqrt{3} - 1250\sqrt{3}c_4 - 8600s_1c_4 - 4300c_4 \]
\[ + 5521c_6 \]  

(40)

It is worth noting that this reduced number of polynomials, compared to the 72 polynomials found by Wang et al., will positively reflect on the computational efficiency for the IK solution of a 6R serial manipulator.

To determine all the values of \( \theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] \), the Solve command is used in MAPLE. Eight different joint configurations are obtained, performing 954 floating-point operations (46 additions, 100 multiplications, and 101 functions), as can be seen in Table 3. These are approximately the same configurations obtained in Table 2.

It is worth noting that all the equations presented in the third section, using Paul’s method to solve the IK of the PUMA 560 manipulator, were calculated using the computer algebra system MAPLE performing 34,514 floating-point operations to determine the eight different configurations, as given in Table 2.

Adding the 466 floating-point operations to obtain the Gröbner basis to the number of floating-point operations to obtain the eight different joint configurations through the generated Gröbner basis (954 flop), 1420 floating-point operations are obtained. There was a reduction from 34,514 to 1420 in the number of floating-point operations.
performed to solve the IK problem for this set of configurations.

Performance analysis

To confirm the computational efficiency for the solution of the IK of anthropomorphic robotic manipulators using Gröbner bases theory, 19 new joint configurations will be analyzed. The following joint values are used:

\[-2.617993878, 3.091716375, -3.048636391, 3.141592654, 0.04307998349, 0.5235987758\]

Table 3. Joint configurations of the Puma 560, obtained through Gröbner bases theory.

| $\theta_1$ (rad) | $\theta_2$ (rad) | $\theta_3$ (rad) | $\theta_4$ (rad) | $\theta_5$ (rad) | $\theta_6$ (rad) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| -2.617993878    | 3.091716375     | -3.048636391    | 3.141592654     | 0.04307998349   | 0.5235987758    |
| -2.617993878    | 3.091716375     | -3.048636391    | 0               | -0.04307998349  | -2.617993878    |
| -0.04220293154  | 1.570796327     | -3.048636391    | 0               | 1.477840064     | -0.04220293154  |
| -0.04220293154  | 1.570796327     | -3.048636391    | 3.141592654     | -1.477840064    | 3.099389722     |
| -2.617993878    | 1.570796327     | 0               | 0               | -1.570796327    | -2.617993878    |
| -2.617993878    | 1.570796327     | 0               | 0               | -0.04987627939  | -0.04220293154  |
| -0.04220293154  | 0.04987627939   | 0               | 0               | -0.04987627939  | 3.099389722     |

Unlike the results presented by Guzmán-Giménez et al.,\textsuperscript{12} all joint configurations provide the same amount of solutions determined by Paul’s method. The two methods were simulated in MAPLE, testing their ability to solve the IK problem, and all the 100 solutions obtained for both methods were compared, testing their efficiency in solving.

Comparisons of the number of mathematical operations performed using the two methods are shown in Figure 5. The $x$-axis presents a partition of 20 sets used to determine 100 different configurations for the PUMA 560 using Paul’s method and Gröbner bases theory, while the $y$-axis shows the total number of operations performed on a logarithmic scale.

The performance comparisons of the two methods are shown in Figure 6. The $x$-axis presents a partition of 20 sets used to determine 100 different configurations for the PUMA 560 using both methods, while the $y$-axis shows the total number of floating-point operations on a logarithmic scale.

It was possible to observe in Figure 5 that there was a significant reduction in the number of mathematical operations to obtain all solutions using Gröbner bases theory. As expected, the number of floating-point operations is reduced, as shown in Figure 6.

The histogram of the RMS errors in the calculations of all six joint angles of the PUMA 560 manipulator in all analyzed configurations is shown in Figure 7. The $x$-axis presents a partition of 100 sets of the obtained root mean square (RMS) errors in ascending order, while the $y$-axis shows the frequency of occurrence of each of these error sets. The RMS error is approximately $8.14 \times 10^{-10}$ rad.

It can be verified in the histogram presented that most of the obtained RMS errors occur within the first sets, the ones that represent less error. It can also be seen that the higher sets still represent very insignificant errors. Figure 8 shows the simulation results of joint variable errors in all 100 configurations analyzed.

The comparative analysis of IK solutions is shown in Figure 9. The red dots represent the angles obtained through Paul’s method for each joint, while the continuous blue lines represent the angles obtained through Gröbner bases theory. The data show that the outputs follow those obtained by Paul’s method with high accuracy and a negligible error.

The histograms of the RMS errors in the calculations of all positions and rotation angles of the end-effector of the PUMA 560 are shown in Figure 10. The $x$-axis presents a partition of 100 sets of the obtained RMS errors, while the $y$-axis shows the frequency of occurrence of each of these error sets. The RMS error obtained is $4.34 \times 10^{-7}$ mm for the position and $1.27 \times 10^{-9}$ rad for the orientation of the robot’s end-effector.

All the histograms presented (Figures 7 and 10) show that most of the obtained RMS errors occur within the first sets, the ones that represent less error. It can also be seen that the higher sets still represent a completely insignificant error.

Figure 11 shows the simulation results of positioning and orientation errors in all 100 configurations analyzed.

The comparative analysis of FK solutions is shown in Figure 12. The green dots represent the $x$, $y$, and $z$ coordinates of the points obtained from the angles determined through Paul’s method, while the solid orange lines represent the coordinates of the points obtained from the angles determined through Gröbner bases theory. The blue dots represent the rotation angles obtained from the angles determined through Paul’s method, while the solid red lines represent the rotation angles obtained from the angles determined through Gröbner bases theory. The data show
that the results continue to maintain high accuracy and negligible error.

El-Sherbiny et al.\textsuperscript{16} presented a comparative study between three different soft-computing-based methods (artificial neural network, adaptive neuro-fuzzy inference system, and genetic algorithms) applied to the problem of IK. The analysis results demonstrate that the methods are not accurate and computationally efficient at the same time. Using the artificial neural network method with a mini-
mized error function and the adaptive neuro-fuzzy inference system method with the same function, the calculation time is quite similar. The difference is in the accuracy of the end-effector position and the angle of rotation. When the genetic algorithm method is used, the position error is very low, which is less than $10^{-3}$ mm, while the calculation time is high compared to the other methods.
It is worth mentioning that in this work, the RMS error obtained is $4.34 \times 10^{-7}$ mm for the position of the end-effector and $1.27 \times 10^{-9}$ rad for the orientation of the end-effector. As shown in Figure 12, the solutions obtained through Gröbner bases theory produce negligible positioning and orientation errors in all configurations.

**Open problem: Analysis of singularities**

Stifter\(^1\) presented an algorithm to solve the IK of a 2R robot using Gröbner bases theory. The author presented a brief list of subjects that were not yet fully resolved but that could play an essential role in applying Gröbner bases theory for solving the problem of IK. The question about the possibility of determining singular configurations using a Gröbner basis is highlighted in this list.

Kendricks\(^1\) used a Gröbner basis to determine the occurrence of a singularity of the workspace boundary, configuration when the manipulator is fully extended to the outer boundary or fully retracted to the inner boundary of its workspace. Her analysis focused on the robot Fanuc A-510, a SCARA type robot with three degrees of freedom, analyzing the discriminant of quadratic polynomials on the Gröbner basis generated to solve the IK of this serial robot. In this case, the author determined intervals of angles that produce a single real solution.
whose discriminants in the quadratic equations are equal to zero.

The anthropomorphic robot Unimation PUMA 560, used as a case study in this work, has the last three axes intersecting at a point. As a consequence, the position of the end-effector is only a function of $q_1, q_2, \text{ and } q_3$. In this way, the following three nonlinear equations in three unknowns are used

\[ x = c_1(a_2c_2 + a_3c_23 - d_3s_23) - d_3s_1 \]  \hspace{1cm} (41)
\[ y = s_1(a_2c_2 + a_3c_23 - d_3s_23) + d_3c_1 \]  \hspace{1cm} (42)
\[ z = -a_3s_23 - a_2s_2 - d_4c_23 \]  \hspace{1cm} (43)

To check whether it is possible to determine from a Gröbner base whether there is a singularity, equations (41) to (43) are rewritten by substituting $x, y, \text{ and } z$ by the respective values of the matrix (27). These three polynomials, added to the three polynomials obtained from the fundamental trigonometric relation for each joint variable, are used to generate a Gröbner basis.

The selected lexicographic order for this robot is the one shown in equation (44). This order represents the same sequence in which the first three joint variables were solved while applying Paul’s method presented in the third section

\[ c_2 > s_2 > c_3 > s_3 > c_1 > s_1 \]  \hspace{1cm} (44)

The following Gröbner basis consisting of six polynomials is obtained using Faugère’s F4 algorithm\(^\text{20}\)

\[ g_1 = (x^2 + y^2)s_1^2 + 250x_1s_1 - y^2 + 15625 \]  \hspace{1cm} (45)
\[ g_2 = c_1y - s_1x - 125 \]  \hspace{1cm} (46)
\[ g_3 = 138325708800s_3^2 + (743040\alpha^2 + 743040\beta^2 + 743040\gamma^2 - 287964408960)s_3 + x^4 + 2x^2y^2 + 2\gamma^2x^2 + \gamma^4 + 2\gamma^2z^2 + z^4 - 775098x^2 - 775098y^2 - 775098z^2 + 149895629001 \]  \hspace{1cm} (47)
\[ g_4 = -x^2 - y^2 - z^2 + 17280c_3 - 371520s_3 + 387549 \]  \hspace{1cm} (48)

Figure 9. Comparative analysis of inverse kinematics solutions ($\theta_1$ to $\theta_6$): (a) $\theta_1$, (b) $\theta_2$, (c) $\theta_3$, (d) $\theta_4$, (e) $\theta_5$, and (f) $\theta_6$. 

Silva et al.\(^\text{11}\)
Figure 10. Histograms of the RMS errors in the computations of all positions and rotation angles of the end-effector. RMS: root mean square.

Figure 11. (a–f) Results of simulation of positioning and orientation errors.
Figure 12. (a–f) Comparative analysis of forward kinematics solutions.

\[ g_5 = (16009920x^2 + 16009920y^2)s_1s_3 + (43x^4 + 86x^2y^2 + 43x^2z^2 + 43y^4 + 43y^2z^2 - 16664607x^2 - 16664607y^2)s_1 + (1728x^2y + 1728y^3 + 1728yz^2 - 27000000y)s_2 + 2001240000xy + 2x^3y + 2y^3z + 2yz^3 + 5375x^3 + 5375y^2x - 28602y^2z - 2083075875x \]

(49)

\[ g_6 = (1728x^2y + 1728y^3 + 1728yz^2 - 27000000y)c_2 + (-2x^4 - 4x^2y^2 - 2x^2z^2 - 2y^4 - 2y^2z^2 + 28602x^2 + 28602y^2)s_1 + 16009920x^3yz + 43x^2yz + 43y^3z + 43yz^2 - 250x^3 - 250x^2y - 16664607yz + 3575250x \]

(50)

Note that the first element in the basis is a quadratic polynomial in terms of \( s_1 \). By examining the discriminant of this equation, it can be observed that there will be no real solutions for \( s_1 \), when

\[ 4y^2(x^2 + y^2 - 15625) < 0 \]

(51)

Moreover, by analyzing the discriminant of the third element in the basis, there will be no real solutions to \( s_3 \) when

\[-x^4 - 2x^2y^2 - 2x^2z^2 + 775098x^2 - y^4 - 2y^2z^2 + 775098y^2 - 2 + 775098z^2 - 11868518601 < 0 \]

(52)

To obtain a quadratic polynomial in terms of \( s_2 \), a new lexicographic order for this robot is used: \( c_3 > s_3 > c_2 > s_2 > c_1 > s_1 \). The following Gröbner basis consisting of six polynomials is obtained using Faugère’s F4 algorithm

\[ g_1 = (x^2 + y^2)s_1^2 + 250x^3 - y^2 + 15625 \]

(53)

\[ g_2 = c_1y - s_1x - 125 \]

(54)

\[ g_3 = (746496x^2 + 746496y^2 + 746496z^2 - 11664000000)s_2^2 + (1728x^2 + 1728y^2z + 1728z^2 - 24712128)x + 2x^4 + 2x^2y^2 + 2x^2z^2 + 4y^4 + 2y^2z^2 + 2z^4 = 775098x^2 - 775098y^2 - 28602x^2 + 11868518601 \]

(55)

\[ g_4 = (864x^2y + 864y^3 - 13500000y)c_3 + (-864x^2y - 864y^3)c_3s_2 + (-x^4 - 2x^2y^2 - x^2z^2 - y^4 - y^2z^2 + 14301x^2 + 14301y^2)s_1 + 108000x^2z_2 - 125x^3 - 125y^3 - 125z^2x + 1787625x \]

(56)

\[ g_5 = (1728x^4 + 3456x^2y^2 + 1728x^2z^2 + 1728y^4 + 1728y^2z^2 - 27000000x^2 - 27000000y^2)s_1s_2 + (2x^4 + 2x^2y^2 + 2x^2z^2 + 2y^4 + 2y^2z^2 - 28602x^2 - 28602y^2)x_1 + (216000x^2 + 216000y^2 + 216000z^2 + 3375000000x_2 + 16009920y^2 + 16009920z^2 - 250155000000y_s_3 + 43x^4 + 43y^2 + 43x^2z^2 + 250x^3 + 250y^2 + 250z^3 - 17336482x^2y - 17336482z^2 + 671875y^2 - 3575250xz + 26038448375y \]

(57)

\[ g_6 = (16009920x^2y + 16009920yz^2 - 250155000000y)c_3 + (37152x^4 + 74304x^2y^2 + 37152x^2z^2 + 37152y^4 + 37152y^2z^2 - 5805000000x^2 - 5805000000y^2)x_1s_2 + (43x^4 + 86x^2y^2 + 43x^2z^2 + 43y^4 + 43y^2z^2 - 61494x^2z + 61494y^2)s_1 + (4644000x^3 + 4644000000y^3 + 4644000000z^2 - 72562500000x_2 - 2x^4y - 2x^2y^2 - 2y^4 - 2y^2z^2 - 2z^4 - 5375xz^2 + 5375y^2z + 5375x^2 - 806348x^2y + 806348y^2 + 31250y^2 - 768678758x - 12110906250y \]

(58)
Note that the third element in the basis is a quadratic polynomial in terms of $s_2$. By examining the discriminant of this equation, it can be observed that there will be no real solutions for $s_2$, when

\[
(2985984(x^2 + y^2 - 15625))(x^4 + (2y^2 + 2z^2 - 775098)x^2 + y^2 + (2z^2 - 775098)y^2 + z^2 - 775098z^2 + 11868518601) < 0
\]

(59)

When it is possible to determine an analytical solution for the IK of a manipulator robot using Gröbner bases theory, given a point $(x;y;z)$, it can determine the maximum number of geometric configurations to position the end-effector of the robot, and, more importantly, it is possible to discover all the real solutions that produce these settings.

However, it is essential to highlight that when all six joint variables are considered in the model, the 6R manipulator presents more than four solutions; however, the Gröbner bases method does not produce analytical solutions, thus requiring numerical methods to find the final solution. In many industry fields, the determination of all rotation angles of the end-effector is indispensable; thus, it justifies the importance of determining all joint angles of a robotic manipulator.

Furthermore, although it is possible to determine angular intervals that produce points that are external to the manipulator’s workspace, singularities such as boundary singularity or inner singularity remain to be further studied.

**Conclusions**

When using the Gröbner bases method for the solution of IK of anthropomorphic robots, one notes in the tests performed during this research that it is computationally more efficient than Paul’s method since the equations produced for determination of six variables are mathematically simpler to be solved. Since through retroactive substitutions, all joint solutions can be determined to precisely position and orient the robotic manipulator end-effector in space.

According to the performance data obtained (number of floating-point operations) in all the configurations analyzed, it can be stated that the method proposed in this work for the solution of IK of anthropomorphic robots is more efficient computationally than Paul’s method.

In future work, it is intended to demonstrate that this method can also be computationally efficient compared to the other methods for solving the IK problem of redundant manipulators.

**Code availability**

The MAPLE source code data used to support the findings of this study are available from the corresponding author upon request.

**Data availability**

The data that support the findings of this study are available from the corresponding author upon request.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES) (Finance Code 001).

**ORCID iD**

Sérgio Ricardo Xavier da Silva  
https://orcid.org/0000-0001-6644-055X

**References**

1. Daya B, Khawandi S, and Akoum M. Applying neural network architecture for inverse kinematics problem in robotics. *J Softw Eng Appl* 2010; III: 230–239.

2. Paul RP, Shimano B, and Mayer G. Kinematic control equations for simple manipulators. *IEEE Trans Syst Man Cybern* 1981; 11: 449–455.

3. Lee CSG. Robot arm kinematics, dynamics, and control. *Computer* 1982; 15: 62–80.

4. Yih TC and Youm Y. Geometrical modeling of lower-pairs based on spherical-Euler geometry. *Trend Develop Mech Mach Robot* 1988; 15: 435–440.

5. Huang B and Milenkovic V. Kinematics of major robot linkages. In: *Proceedings of the 13th International Symposium on Industrial Robots*; 1983; Chicago: Robotics International of SME, pp. 16–31.

6. Manseur R and Doty KL. A fast algorithm for inverse kinematic analysis of robot manipulators. *Int J Robot Res* 1988; 7: 52–63.

7. Denavit J and Hartenberg RS. *Description and displacement analysis of mechanics based on the 2 × 2 dual matrices*. Evanston: University of Northwestern, 1955.

8. Hartenberg RS and Denavit J. *Kinematic synthesis of linkages*. Col: McGraw-Hill series in mechanical engineering. New York: McGraw-Hill, 1965.

9. Paul RP. *Robot manipulators: mathematics, programming, and control (artificial intelligence)*. Cambridge: MIT Press, 1981.

10. Buchberger B. Bruno Buchberger’s PhD thesis 1965: an algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *J Symbol Comput* 2006; 41: 475–511.

11. Kendricks KD. Solving the inverse kinematic robotics problem for the GMF-A510 robot: a comparison study of the denavit-hartenberg matrix and groebner basis theory. PhD Thesis, The Graduate School of Auburn University; Auburn University Libraries, Auburn, USA, 2007.

12. Guzmán-Giménez J, Fernández ÁV, Amela VM, et al. Synthesis of the inverse kinematic model of non-redundant open-chain robotic systems using Gröbner basis theory. *Appl Sci* 2020; 10: 2781.
13. Ni Z and Wu R. 6R robot inverse solution algorithm based on quaternion matrix and Gröbner base. *Adv Linear Algebra Matrix Theory* 2018; VIII: 33–40.

14. Wang Y, Hang L, and Yang T. Inverse kinematics analysis of general 6r serial robot mechanism based on Gröbner base. *Front Mech Eng China* 2006; 1(I): 115–124.

15. Duffy J and Crane CD. Displacement analysis of the general 7-link 7R mechanism. *Mech Mach Theory* 1980; XV: 153–169.

16. El-Sherbiny A, Elhosseini M, and Haikal AY. A comparative study of soft computing methods to solve inverse kinematics problem. *Ain Shams Eng J* 2017; 9. DOI: 10.1016/j.asej.2017.08.001.

17. Cox D, Little A, and O’Shea D. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. 4th ed. New York: Springer, 2015.

18. Sturmfels B. What is . . . a Gröbner basis? *Not Am Math Soc* 2005; 52(10): 1199–1200.

19. Stifter S. Algebraic methods for computing inverse kinematics. *J Intell Robot Syst* 1994; 11: 79–89.

20. Faugère JC. A new efficient algorithm for computing Gröbner bases (F4). *J Pure Appl Algebra* 1999; 139: 61–88.

21. Ghosal A. *Robotics: fundamental concepts and analysis*, New Delhi: Oxford University Press, 2006.

22. Chakravarti M and Sivakumar AL. A GA-integrated adaptive model reference controller in robot tracking applications. *J Multidiscip Eng Sci Studies* 2016; 2: 283–291.

23. Craig JJ. *Introduction to robotics: mechanics and control*. (4th ed). Boston: Addison-Wesley, 2017.