Inverse Kinematics of 6R Manipulator by Gröbner Basis Computation

Viktor Korotynskiy

December 2024

I. Symbolic Formulation of Inverse Kinematics

Mathematical formulation of inverse kinematics

Let $\left\{\mathbf{M}_i^{i-1} \mid i \in [6]\right\}$ be the 4×4 transformation matrices between the coordinate frames assigned to the adjacent links (according to the DH notation) and $\mathbf{M}_e$ be the transformation from the end-effector frame to the base frame (we use the notation $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$). Then

$$\underbrace{\prod_{i=1}^{6} \mathbf{M}_i^{i-1}(\theta_i + \underbrace{\theta_{i_{\text{offset}}}, d_i, a_i, \alpha_i}_{\text{DH parameters}})}_{\mathbf{M}(\boldsymbol{\theta})} = \underbrace{\begin{bmatrix} \mathbf{R}_e & \mathbf{t}_e \\ \mathbf{0}^\top & 1 \end{bmatrix}}_{\mathbf{M}_e},$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)$. If we focus on the non-constant elements of $\mathbf{M}(\boldsymbol{\theta})$, then we get a set of 12 functions

$$\left\{\mathbf{M}^{(i,j)}(\boldsymbol{\theta}) - \mathbf{M}_e^{(i,j)} \,\middle|\, i \in [3], j \in [4]\right\},$$

whose zero set defines the solutions to IKT.

Symbolic formulation of inverse kinematics

Every matrix $\mathbf{M}_i^{i-1}(\theta_i)$ can be decomposed as

$$\begin{bmatrix} \cos(\theta_i + \theta_{i_{\text{offset}}}) & -\sin(\theta_i + \theta_{i_{\text{offset}}}) & 0 & 0 \\ \sin(\theta_i + \theta_{i_{\text{offset}}}) & \cos(\theta_i + \theta_{i_{\text{offset}}}) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

New variables:

$$c_i = \cos(\theta_i + \theta_{i_{\text{offset}}}), \quad s_i = \sin(\theta_i + \theta_{i_{\text{offset}}})$$

Symbolic formulation:

$$\begin{cases} \prod_{i=1}^{6} \mathbf{M}_i^{i-1}(c_i, s_i) - \mathbf{M}_e = \mathbf{O} \\ {c_i}^2 + {s_i}^2 - 1 = 0, \quad i = 1, \ldots, 6 \end{cases}$$

Computing θ_i from c_i and s_i:

$$\theta_i = \operatorname{atan2}(s_i, c_i) - \theta_{i_{\text{offset}}}$$

Reducing degree of polynomials

The inverse matrix $\left(\mathbf{M}_i^{i-1}\right)^{-1}$ has the form:

$$\begin{bmatrix} 1 & 0 & 0 & -a_i \\ 0 & \cos\alpha_i & \sin\alpha_i & 0 \\ 0 & -\sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_i & s_i & 0 & 0 \\ -s_i & c_i & 0 & 0 \\ 0 & 0 & 1 & -d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since it is linear in c_i and s_i, the former polynomial equations of IKT of degree at most 6 can be simplified to the following ones of degree at most 3:

$$\prod_{i=4}^{6} \mathbf{M}_i^{i-1}(c_i, s_i) - \left(\prod_{i=1}^{3} \mathbf{M}_i^{i-1}(c_i, s_i)\right)^{-1} \mathbf{M}_e = \mathbf{O},$$

$$c_i{}^2 + s_i{}^2 - 1 = 0, \quad i = 1, \ldots, 6$$

Notice that this is not the only way to reduce the degree: we can multiply $\mathbf{M}_e$ by the inverses of $\mathbf{M}_i^{i-1}$ either from left or right, which gives in total $2^3 = 8$ various reduced formulations.

Solvability of equations by Gröbner basis method

1. Gröbner Basis computation is done in exact arithmetics over the rational numbers and therefore the rational input must be provided.
2. At the same time, the input must satisfy all identities on sines, cosines and rotations, otherwise the equations would have no solution.

That's why,

1. the parameters d_i, a_i, $\mathbf{t}_e$ must be given as rational numbers;
2. $\cos\alpha_i$ and $\sin\alpha_i$ must be given as a tuple of rational numbers such that the sum of their squares is exactly 1
3. the rotation matrix $\mathbf{R}_e$ must be given as a rational matrix such that $\mathbf{R}_e^\top \mathbf{R}_e = \mathbf{I}$ and $\det \mathbf{R}_e = 1$ hold exactly.

In Python 3, rational numbers can be represented by `Fraction` type from `fractions` module.

II. Naive Approach to Rationalization of Inverse Kinematics

Rational approximation of a floating point number

Algorithm 1: Rational approximation of a floating point number

Input: float number n, positive tolerance ε

Output: rational number $(a, b) \in \mathbb{Z}^2$ such that

$$\left|\frac{a}{b} - n\right| < \varepsilon$$

if $\varepsilon \geq 1$ **then**
  return $(\lfloor n \rfloor, 1)$

represent $\varepsilon = m \cdot 10^e$ for $m \in [1, 10)$, $e \in \mathbb{Z}_{\leq 0}$

return $(\lfloor n \cdot 10^{-e} \rfloor, 10^{-e})$

Rational approximation of a floating point number

Example

Let the floating point number be

$$n = 10.123456789$$

and the tolerance of the rational approximation be given by another floating point number

$$\varepsilon = 0.000025932 = 2.5932 \times 10^{-5} \Rightarrow e = -5$$

A rational number which approximates n is

$$(a, b) = (\lfloor 10.123456789 \cdot 10^5 \rfloor, 10^5) = (1012345, 10^5)$$

$$\left| \frac{a}{b} - n \right| = |10.12345 - 10.123456789| = 0.000006789 < \varepsilon$$

Parametrization of rational cosines and sines

Rational parametrization of the unit circle:

$$\cos\theta = \frac{1-t^2}{1+t^2}, \;\; \sin\theta = \frac{2t}{1+t^2}, \;\; t \in \mathbb{Q}$$

Trigonometric meaning of t:

$$\tan\frac{\theta}{2} = \frac{1-\cos\theta}{\sin\theta} = \frac{1-\frac{1-t^2}{1+t^2}}{\frac{2t}{1+t^2}} = \frac{2t^2}{2t} = t$$

Rational cosine and sine

Algorithm 1: Rational cosine and sine

Input: $\theta \in (-\pi, \pi]$, positive tolerance ε

Output: Rational numbers c, s such that

$$c^2 + s^2 = 1 \ \text{(exactly)} \quad \& \quad |c - \cos\theta| < \varepsilon,\ |s - \sin\theta| < \varepsilon$$

if $1 + \cos\theta < \varepsilon$ **and** $|\sin\theta| < \varepsilon$ **then**
  return $-1, 0$
$\varepsilon_t \leftarrow \varepsilon$
while *True* **do**
  $t \leftarrow \texttt{rational\_approx}(\tan\frac{\theta}{2}, \varepsilon_t)$
  $c, s \leftarrow \frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}$
  if $|c - \cos(\theta)| < \varepsilon$ **and** $|s - \sin(\theta)| < \varepsilon$ **then**
    return c, s
  else
    $\varepsilon_t \leftarrow \frac{\varepsilon_t}{10}$

Rational cosine and sine

Example

Let the angle and the tolerance be given by

$$\theta = 1.2345, \quad \varepsilon = 0.0023$$

The output of the above algorithm is

$$c = \frac{1-t^2}{1+t^2} = \frac{497319}{1502681} \approx \cos\theta, \quad s = \frac{2t}{1+t^2} = \frac{1418000}{1502681} \approx \sin\theta$$

which is obtained from the rational approximation of $\tan\frac{\theta}{2}$

$$t = \frac{709}{1000} \approx \tan\frac{\theta}{2}$$

Parametrization of rational 3×3 rotation matrices

For $q_1, q_2, q_3, q_4 \in \mathbb{Q}$ we have a rational rotation matrix

$$\mathbf{R} = \frac{1}{\sum_{i=1}^{4} q_i^2} \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 + q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}$$

In the next slide we are using the function `q2r` which is defined by the above formula.

Rational 3×3 rotation matrix

Algorithm 1: Rational 3×3 rotation matrix

Input: 4 float numbers $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^\top$ with $\|\mathbf{q}\| \approx 1$, positive tolerance ε

Output: 3×3 rational matrix $\mathbf{R}$ such that

$$\mathbf{R}^\top \mathbf{R} = \mathbf{I}, \;\; \det \mathbf{R} = 1 \;\; \text{(exactly)} \quad \& \quad \|\mathbf{R} - \mathtt{q2r}(\mathbf{q})\|_{\mathrm{F}} < \varepsilon$$

$\varepsilon_q \leftarrow \varepsilon$
while *True* **do**
    $\mathbf{q}_r \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$
    for **(** $k \leftarrow 1; k \leq 4; k \leftarrow k + 1$ **)**
        $\mathbf{q}_r[k] \leftarrow \mathtt{rational\_approx}(q_k, \varepsilon_q)$
    $\mathbf{R} \leftarrow \mathtt{q2r}(\mathbf{q}_r)$
    if $\|\mathbf{R} - \mathtt{q2r}(\mathbf{q})\|_{\mathrm{F}} < \varepsilon$ **then**
        return R
    else
        $\varepsilon_q \leftarrow \frac{\varepsilon_q}{10}$

Rational 3×3 rotation matrix

Example

Let the quaternion and the tolerance be given by

$$\mathbf{q} = \begin{bmatrix} 0.748 & 0.654 & 0.108 & 0.012 \end{bmatrix}, \quad \varepsilon = 0.0011$$

The output of the above algorithm gives the rational rotation matrix

$$\mathbf{R} = \begin{bmatrix} \frac{243853}{249757} & \frac{30828}{249757} & \frac{44316}{249757} \\ \frac{39804}{249757} & \frac{35827}{249757} & -\frac{243948}{249757} \\ -\frac{36468}{249757} & \frac{245244}{249757} & \frac{30067}{249757} \end{bmatrix}$$

which comes from the (non-unit) rational quaternion

$$\mathbf{q}_r = \begin{bmatrix} \frac{187}{250} & \frac{327}{500} & \frac{27}{250} & \frac{3}{250} \end{bmatrix} \approx \mathbf{q}$$

III. Solving Inverse Kinematics by Gröbner Basis Computation

Reduced lexicographic Gröbner basis of IKT equations

IKT equations:

$$\prod_{i=4}^{6} \mathbf{M}_i^{i-1}(c_i, s_i) - \left(\prod_{i=1}^{3} \mathbf{M}_i^{i-1}(c_i, s_i)\right)^{-1} \mathbf{M}_e = \mathbf{O},$$

$$c_i^{\,2} + s_i^{\,2} - 1 = 0, \quad i = 1, \ldots, 6$$

For the variable ordering

$$c_1 > s_1 > \cdots > c_6 > s_6$$

the reduced lexicographic Gröbner basis of IKT equations has the form:

$$s_6^{\,16} + a_{1,15} \cdot s_6^{\,15} + a_{1,14} \cdot s_6^{\,14} + \cdots + a_{1,1} \cdot s_6 + a_{1,0}$$

$$c_6 + a_{2,15} \cdot s_6^{\,15} + a_{2,14} \cdot s_6^{\,14} + \cdots + a_{2,1} \cdot s_6 + a_{2,0}$$

$$\vdots$$

$$c_1 + a_{12,15} \cdot s_6^{\,15} + a_{12,14} \cdot s_6^{\,14} + \cdots + a_{12,1} \cdot s_6 + a_{12,0}$$

where $a_{i,j} \in \mathbb{Q}$.