Kinematics of minibot-7R

Aykut C. Satici *Member, IEEE*

***Abstract*—This technical report contains the kinematics of the minibot-7R robot.**

***Index Terms*—kinematics**

I. Introduction

We want to derive the kinematics of the 7-degree-of-freedom robotic manipulator, minibot-7R. A kinematic diagram of the robot is provided in Figure 1.

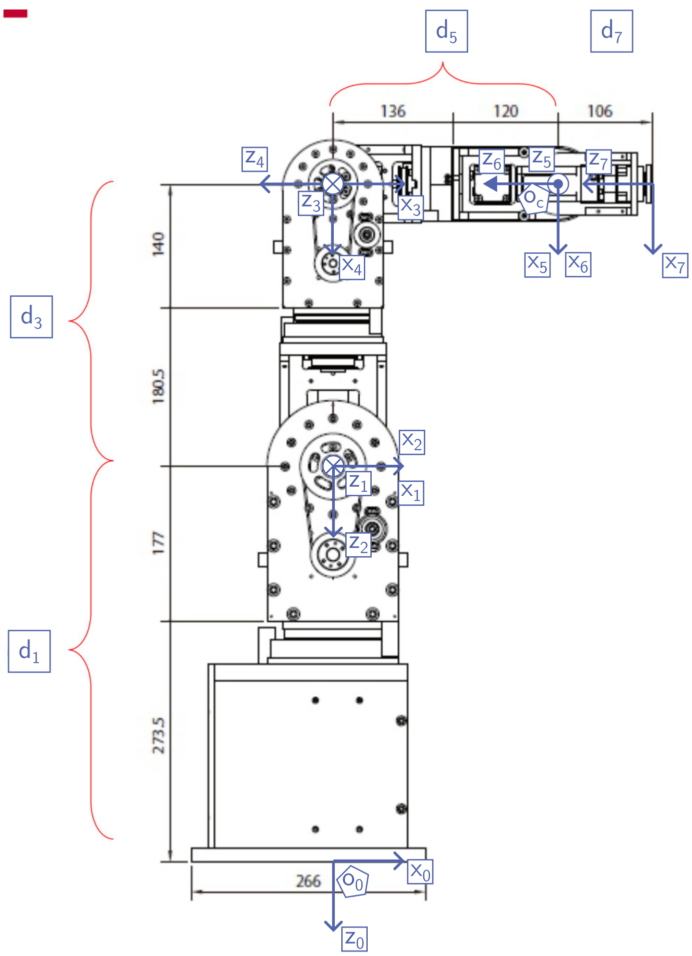

Fig. 1: Schematic of minibot-7R.

We will follow the classic Denavit-Hartenberg convention as presented in [1].

II. Denavit-Hartenberg Formulation

The transformation matrices between consecutive frames are given as follows.

TABLE I: DH Table for minibot-7R

Link	α_i [°]	a_i	d_i [mm]	θ_i [°]
1	90	0	-450.5	θ_1^*
2	-90	0	0	θ_2^*
3	90	0	-320.5	θ_3^*
4	-90	0	0	θ_4^*
5	-90	0	-256	θ_5^*
6	90	0	0	θ_6^*
7	0	0	-106	θ_7^*
Home:	$\theta_i = 0°$,	$i \neq 4$;		$\theta_4 = 90°$

$$
\begin{aligned}
A_1 &= \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
A_3 &= \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
A_5 &= \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_6 = \begin{bmatrix} c_6 & 0 & s_6 & 0 \\ s_6 & 0 & -c_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
A_7 &= \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{1}
$$

The forward kinematics map is then given by

$$
f : \mathcal{Q} \triangleq \prod_{1}^{7} \mathbb{S}^1 \to \mathrm{SE}(3), \quad f(\boldsymbol{q}) = \prod_{1}^{7} A_i(\theta_i),
$$

III. Velocity Kinematics

We derive the Jacobian $\boldsymbol{J} : \mathbb{R}^7 \to \mathfrak{se}(3)$, that maps the rate of change of the joint variables to the end effector twist. This map can be represented in its matrix form by the concatenation of two submatrices, $\boldsymbol{J}_v$ and $\boldsymbol{J}_\omega$, as

$$
\boldsymbol{J} = \begin{pmatrix} \boldsymbol{J}_v \\ \boldsymbol{J}_\omega \end{pmatrix}, \tag{2}
$$

where $\boldsymbol{J}_v$ and $\boldsymbol{J}_\omega$ are both elements of $\mathbb{R}^{3\times 7}$. The construction of these matrices are subsequently shown, starting with $\boldsymbol{J}_v$.

$$
\begin{aligned}
\boldsymbol{J}_v &= \begin{bmatrix} \boldsymbol{J}_{v_1} & \boldsymbol{J}_{v_2} & \cdots & \boldsymbol{J}_{v_7} \end{bmatrix}, \\
\boldsymbol{J}_{v_i} &= \boldsymbol{z}_{i-1} \times (\boldsymbol{o}_c - \boldsymbol{o}_{i-1}).
\end{aligned} \tag{3}
$$

The second part of the Jacobian matrix is constructed as

$$\boldsymbol{J}_\omega = \begin{bmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_6 \end{bmatrix}. \tag{4}$$

With the Jacobian constructed as in (2), the end-effector twist $\boldsymbol{\xi} \in \mathfrak{se}(3)$ is related to the joint rates $\dot{\boldsymbol{\theta}} \in \mathbb{R}^7$ as

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \boldsymbol{\xi}.$$

IV. Inverse Kinematics (IK)

Suppose that we are given a target pose $H \in \mathrm{SE}(3)$:

$$\boldsymbol{H}_t = \begin{bmatrix} \boldsymbol{R}_t & \boldsymbol{o}_t \\ 0 & 1 \end{bmatrix}.$$

We want to find a $\boldsymbol{\theta} \in \mathcal{Q}$ such that $f(\boldsymbol{\theta}) = \boldsymbol{H}_t$. We describe two methods to achieve this, which would yield equivalent solutions for a fully-actuated robot, but the differences get amplified when the robot is redundant such as minibot-7R.

We will be using gradient-based optimization to solve for the joint angles. This is a recursive algorithm that improves on the current guess $\boldsymbol{\theta}^{(k)}$ to a solution $\boldsymbol{\theta}^d$ by performing the following iteration.

$$\begin{aligned} \bar{\boldsymbol{\theta}}^{(k+1)} &= \boldsymbol{\theta}^{(k)} - \alpha_k \boldsymbol{M}\left(\boldsymbol{\theta}^{(k)}\right)\delta\boldsymbol{x}, \\ \boldsymbol{\theta}^{(k+1)} &= \mathrm{clamp}\left(\bar{\boldsymbol{\theta}}^{(k+1)}, \boldsymbol{\theta}_{\mathrm{lb}}, \boldsymbol{\theta}_{\mathrm{ub}}\right). \end{aligned} \tag{5}$$

The matrix $\boldsymbol{M}$ is some sort of Jacobian, $\delta\boldsymbol{x}$ is the pertinent error that the current guess produces, and $(\boldsymbol{\theta}_{\mathrm{lb}}, \boldsymbol{\theta}_{\mathrm{ub}})$ is the lower and upper bounds on the joint angles.

A. IK by Decoupling Position and Orientation

We first consider the inverse position kinematics problem, which relies on the fact that the inverse orientation kinematics may be solved by using the final three joints due to the presence of a spherical wrist. Indeed, whatever the rotation matrix ${}^0\boldsymbol{R}_4$ is, there exist θ_5, θ_6 and θ_7, such that ${}^0\boldsymbol{R}_4{}^4\boldsymbol{R}_7 = \boldsymbol{R}_t$. Since ${}^0\boldsymbol{R}_4$ will be determined once the inverse position kinematics problem is solved, one can then use θ_5 through θ_7 to yield ${}^4\boldsymbol{R}_7 = {}^0\boldsymbol{R}_4^\top \boldsymbol{R}_t$ following the steps outlined in Chapter 5.4 of [1] (pg. 151).

The inverse position kinematics problem is to find q_1 through q_4 such that the product $\prod_1^4 \boldsymbol{A}_i$ has as its translation vector the wrist center location, $\boldsymbol{o}_c \in \mathbb{R}^3$.

In order to solve the inverse position problem, we first find the location of the wrist center $\boldsymbol{o}_c$ in the base coordinate system, Σ_0:

$$\boldsymbol{o}_c = \boldsymbol{o} - d_7 \boldsymbol{R}_t \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top.$$

The gradient descent optimization in equation (5) is then performed with $\delta\boldsymbol{x} = \left(f_c\left(\boldsymbol{\theta}_{1:4}^{(k)}\right) - \boldsymbol{o}_c\right)$; where $f_c : \prod_{i=1}^4 \mathbb{S}^1 \to \mathbb{R}^3$ maps the first four joint angles to the position of the wrist center. We can take the matrix $\boldsymbol{M}$ as the pseudo-inverse or the transpose of the Jacobian, $\boldsymbol{J}_w$, that maps the rates of changes $\dot{q}_1$ through $\dot{q}_4$ to the rate of change of the wrist center, i.e., $\boldsymbol{J}_w\dot{\boldsymbol{\theta}}_{1:4} = \dot{\boldsymbol{o}}_c$. This matrix $\boldsymbol{J}_w$ corresponds to the first three-by-four block of the matrix $\boldsymbol{J}_v$ from equation (3). Notice that the step size α_k may be orders of magnitude different depending on whether $\boldsymbol{M} = \boldsymbol{J}_w^\top$ or $\boldsymbol{M} = \boldsymbol{J}_w^\dagger$.

B. Fully Coupled IK

Instead of decoupling the position and orientation, we can perform a gradient step to update all the joint angles in response to both the position and orientation error at the end-effector (as opposed to the wrist center). The gradient descent update due to the error in the end-effector position is very similar to the one due to the wrist center IV-A, except that we use the Jacobian and the error at the end-effector rather than at the wrist center. Furthermore, in this case, we use descent on the full set of angles $\boldsymbol{q}$ rather than only the first four.

The gradient descent update due to the orientation error is a bit more tricky to derive due to the curved nature of the space of orientations. We can perform the gradient descent by reasoning directly on $\mathrm{SO}(3)$, but we let us see how to do this in the space of unit quaternions $\mathbb{H}$. This space is topologically a double cover space of $\mathrm{SO}(3)$, and its tangent space is isomorphic to $\mathfrak{so}(3)$.

We start by converting the target rotation $\boldsymbol{R}_t$ to its representation in $\mathbb{H}$. This can be achieved by following standard procedures; one is provided in [2]. Suppose that the target orientation is denoted by the unit quaternion $\boldsymbol{q}_t$ and the current guess at the joint angles $\boldsymbol{\theta}^{(k)}$ results in the orientation $\boldsymbol{q}_e$. The error this guess makes is given by the residual unit quaternion $\boldsymbol{q}_r$ that satisfies

$$\boldsymbol{q}_e \circ \boldsymbol{q}_r = \boldsymbol{q}_t \quad \Rightarrow \quad \boldsymbol{q}_r = \boldsymbol{q}_e^* \circ \boldsymbol{q}_t.$$

This residual may be expressed as the exponential of an element $\boldsymbol{d} \in \mathbb{R}^3$, identified with the space of pure quaternions $\mathbb{H}_p$ (the tangent space at identity of unit quaternions, isomorphic to $\mathfrak{so}(3)$). We can express $\boldsymbol{d} = \phi\boldsymbol{u}$, where ϕ is the magnitude and $\boldsymbol{u}$ is a unit vector in $\mathbb{R}^3$. Hence, we have

$$\boldsymbol{q}_e^* \circ \boldsymbol{q}_t = e^{\phi\boldsymbol{u}} \quad \Rightarrow \quad \phi\boldsymbol{u} = \log\left(\boldsymbol{q}_e^* \circ \boldsymbol{q}_t\right).$$

Note that this is the exponential map acting from the Lie algebra $\boldsymbol{H}_p$ and mapping to the space of unit quaternions $\boldsymbol{H}$, which is defined as

$$\exp\left(\phi\boldsymbol{u}\right) \triangleq \cos\phi \; \langle \sin\left(\phi\right)\boldsymbol{u} \rangle$$

and the logarithm map is its inverse.

We can now perform the gradient descent update for all the joint angles $\boldsymbol{\theta}$ as in equation (5), by choosing $\boldsymbol{M}$ as the Jacobian transpose of the mapping $\boldsymbol{\theta} \mapsto \phi\boldsymbol{u}$ and $\delta\boldsymbol{x} = \phi\boldsymbol{u}$.

References

[1] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2020.

[2] Wikipedia, "Quaternions and spatial rotation — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Quaternions%20and%20spatial%20rotation&oldid=1226083493, 2024. [Online; accessed 11-June-2024].