robot arm documentation

thomas-hiemstra Hiemstra

June 2017

1 kinematics

First off some notations which are used for the DH matrices. Since there are a lot of sines and cosines of the angles $q_1, ..., q_6$ a shorthand is used for the sine and cosine of these angles. Namely: $\cos(q_i) = c_i$, $\sin(q_i) = s_i$ and $\cos(q_i + q_j) = c_{ij}$.

1.1 Forward kinematics

Let's start with the forward kinematics of the robot arm of which I will give a brief overview. The method used for this is the Denavit-Hartenberg convention. In this convention coordinate frames are attached to links of the robot arm. Then a coordinate transformation between these 2 frames (in the form of a matrix) is then defined according to the convention. A good visually guided explanation is given here [2]. for a thorough explanation see [4].

In short:

- the z-axis is the direction of the joint axis
- the x-axis is parallel to the common normal $x_n = z_n \times z_{n-1}$
- the y-axis follow from a right handed coordinate system

Then the parameters are:

- d: offset along previous z to the common normal
- θ: angle about previous z, from old x to new x
- a: length of the common normal (in the video called r)
- α: angle about common normal, from old z-axis to new z-axis

These 4 parameters are then plugged into the DH matrix which then in turn gives the full rotation and translation between the 2 links:

$$A_i^{i-1}(q_i) = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Here the upper left 3x3 matrix represents the rotation and the right 1x3 column vector represents the translation.

The forward (and inverse) kinematics of my robot arm can be split up into 2 parts. One for the anthropomorphic arm and 1one for the spherical wrist (with some minor but extremely important subtleties which I will go over later). Let's start with the arm:

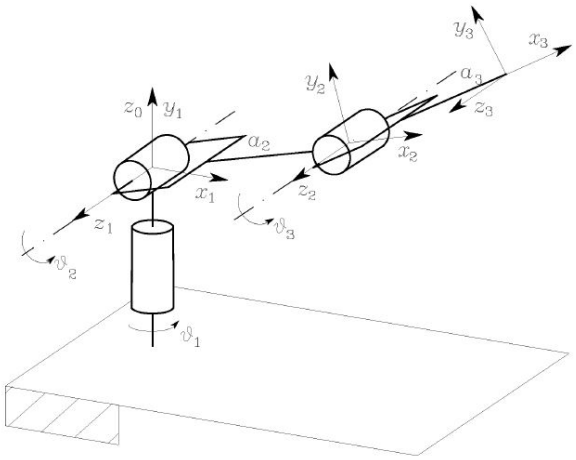


Figure 1: anthropomorphic arm

The parameters of the arm are given by:

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

Notice that frame 3 here is left hanging since it is not possible to directly attatch something like a gripper to it in it's current configuration. Because the Z-axis of a gripper would be in the gripping direction whereas the Z-axis of the final frame is pointing sideways, this will become important later on.

Next the wrist and it's parameters:

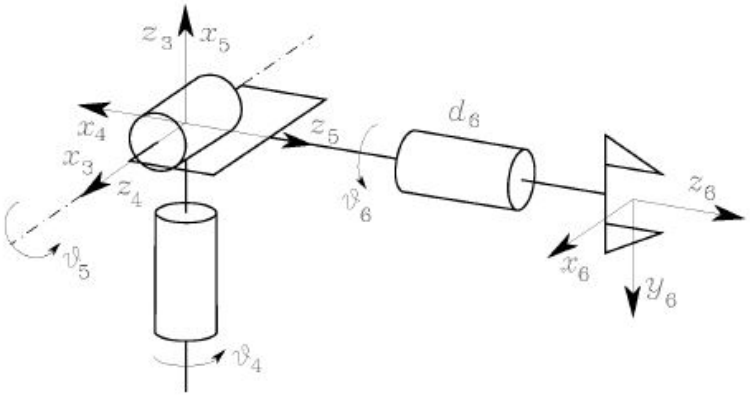


Figure 2: Spherical wrist

Link	a_i	α_i	d_i	ϑ_i
4	0	$-\pi/2$	0	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6

Notice again here that the wrist cannot be directly connected to the arm since the Z-axis of frame 3 does not coincide with frame 3 of the arm. Also notice the orientation of frame 5 with respect to frame 4 which determines the behaviour of θ_5. In figure 2 the wrist is drawn for $\theta_5 = \pi/2$.

Finally the full arm:

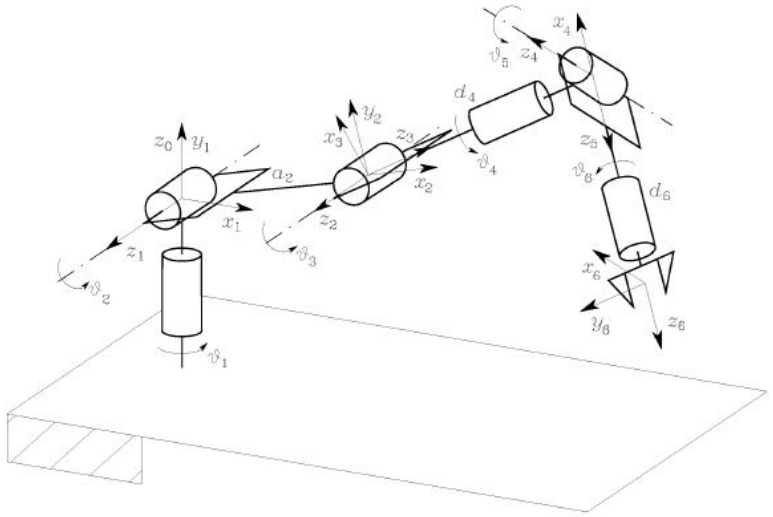


Figure 3: anthropomorphic arm

The parameters of the full arm are given by:

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	0	$\pi/2$	0	ϑ_3
4	0	$-\pi/2$	d_4	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6

Notice here that the parameters for frame 3 and 4 are different than for the arm and wrist given above. The reason as stated before is that the wrist cannot be directly connected to the arm since frame 3 of the arm and frame 3 of the wrist do not coincide. Changing α fixes that.
IMPORTANT: this has the consequence that θ_3 behaves in a weird way, when the arm is stretched out θ_3 would have been 0 for the case without the wrist, now $\theta_3 = \pi/2$!! This explains the "angles[3] += pi/2" in the code.
The DH matrices for the full arm are worked out in the mathematica notebook.

1.2 Inverse kinematics

The general inverse kinematics problem is generally impossible to solve analytically. However the case of a 6 axis robotic arm where the last 3 joints are intersection at a point is a special case. Here it is possible to decouple the inverse kinematics into 2 simpler problems. Namely inverse position kinematics and inverse orientation kinematics.
The input for a given position should counting 6 parameters. 3 for position and 3 for orientation. From the orientation parameters the wrist center can be computed since distance from center to gripper is a constant. Next the position of the arm is calculated using the position inverse kinematics. Then using the (now calculated) values of the first 3 joints the orientation of the arm is computed. Finally we need to go from the orientation of the arm to the desired orientation of the gripper which is done by calculating the values of the last 3 joints.

The input for our location are an origin O and a rotation matrix R. From this we have:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d6r_{13} \\ o_y - d6r_{23} \\ o_z - d6r_{33} \end{bmatrix} \tag{1}$$

where the subscript c denotes the origin of the wrist and r are the elements of the input rotation matrix. Using the position as the wrist for the desired location of the arm we can perform the inverse position kinematics which are worked out in [1]. The resulting formulas are:

$$\theta_1 = Atan2(y_c, x_c) \tag{2}$$

$$\theta_3 = Atan2(-\sqrt{1 - D^2}, D) \tag{3}$$

$$\theta_2 = Atan2(z_c - d1, \sqrt{x_c^2 + y_c^2}) - Atan2(k2, k1) \tag{4}$$

With D,k1, k2 given in [1]
Again notice that in this reference the arm is slightly different since there is no wrist attached to it so θ_3 has to be modified by $\pi/2$. I'm also picking the elbow up configuration since elbow down would make no sense for my robot.
Now all that is left is the orientation part. Since we have the values of the first 3 angles we can compute the rotation matrix of the arm R_3^0. The final orientation is given by $R = R_3^0 R_6^3$ where R is the input rotation matrix. Since R and R_3^0 are known we can compute R_6^3. Noting that the inverse of a rotation matrix is it's transpose we get:

$$R_6^3 = (R_6^3)^T R \tag{5}$$

R_6^3 is obtained from the DH matrix of the full robot arm. This is a matrix in terms of $\theta_4, \theta_5, \theta_6$ and has the form of the ZYZ euler matrix. So now we are left with the matrix equation:

$$R_6^3 = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_x & a_x \\ n_z & s_x & a_x \end{bmatrix} \tag{6}$$

Where the left hand side contains the unknown angles and the right hand side is $(R_6^3)^T R$ which is known (AKA just a bunch of numbers). (n,s and a stand for normal, sliding and approach direction respectively). Since R_6^3 is the ZYZ euler matrix the solution for the angles can be found in the literature and it reads:

$$\theta_4 = Atan2(a_y, a_x) \tag{7}$$

$$\theta_5 = Atan2(\sqrt{(a_x)^2 + (a_y)^2}, a_z) \tag{8}$$

$$\theta_6 = Atan2(s_z, -n_z) \tag{9}$$

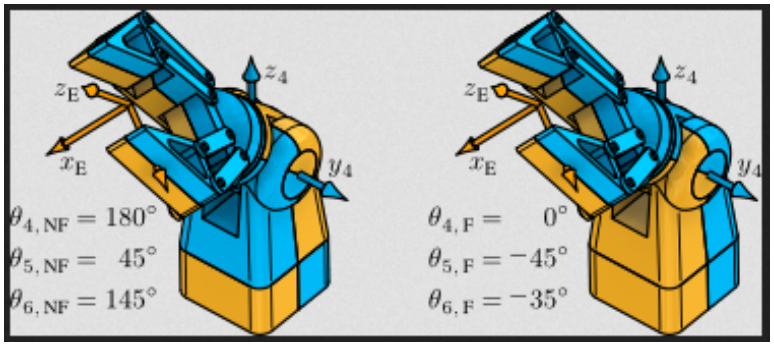
for $\theta_5 \in (0, \pi)$ and

$$\theta_4 = Atan2(-a_y, -a_x) \tag{10}$$

$$\theta_5 = Atan2(-\sqrt{(a_x)^2 + (a_y)^2}, a_z) \tag{11}$$

$$\theta_6 = Atan2(-s_z, n_z) \tag{12}$$

for $\theta_5 \in (-\pi, 0)$.
This means the target orientation can be reached in 2 ways. With the wrist up or down which corresponds to having the wrist flipped or not as illustrated in this picture:



As can be seen flipping the wrist corresponds to adding π to θ_4, changing the sign of θ_5 and adding π to θ_6. Having the wrist in the correct flip can prevent a so called configuration flip. This happens for example when the gripper

moves in a line through x=0. Since $\theta_4 \in (-\pi, \pi)$ it cannot go to 181 degrees, in order to achieve this the entire configuration of the robot arm will flip. This is why moving the gripper in straight lines can be problematic since it is very hard to figure out where the configuration flips might occur. In order to avoid this you have to either pick the correct flip (and flip the wrist at points where it causes minimal arm movement) or you simply let the joints go smoothly from their starting configuration to their ending configuration, this is done in the pointToPoint function.

1.3 singularities

a quick note on singularities, when $\theta_5 = 0$ the Atan2 function becomes undefined (divide by 0 error) which is called a wrist singularity. I avoid this by adding epsilon to θ_5 when this happens but this still causes numeric instability. There are other singularities such as the robot having to move extremely fast to make the gripper move a constant rate, many of such things are shown in this video: [5]

1.4 practical implantation

Here I will go over the inverseKinematicsRaw function in the code [3]. I have used an array size of 7 because I wanted to label the angles just like the book does (not calling θ_1 angles[0] in the code, that would have confused me too much). So angles[0] does nothing. The function takes a position and orientation as input and returns the values of the angles. A conversion is needed since the servo motors are not aligned with the DH frames. the inverseKinematics function has this conversion built in.
The eulermatrix function is a bit special because of the fact that the z-axis of the gripper is it's approach direction, not it's x-direction. So a unit matrix would cause it to point upwards since this matrix is defined with respect to the base frame. Permuting the columns of the matrix causes the gripper to behave as if it's approach direction was it's x-axis.
getServoTick is used to convert an angle to the correct servo signal. The servos take a PWM signal as input and ideally you'd only have to know the signal for 0 and 180 and then interpolate between these 2 to go from angle to PWM. However the servo response is not quite linear so I used a piecewise linear function and measured for each servo the correct PWM signal ever 45 degrees. (this has to be done by hand and is awe-full).

2 robot vision

References

[1] *Dr. Rainer Hessmer. Kinematics for Lynxmotion Robot Arm.* URL: http://www.hessmer.org/uploads/RobotArm/Inverse%2520Kinematics%2520for%2520Robot%2520Arm.pdf.

[2] *Inverse kinematics video.* URL: https://www.youtube.com/watch?v=rA9tm0gTln8.

[3] *robot arm with vision.* URL: https://github.com/thomashiemstra/robot_arm_with_vision.

[4] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.

[5] *Singularities of a six-axis robot.* URL: https://www.youtube.com/watch?v=zlGCurgsqg8.