IVR Assignment Report

Humbarwadi Srihari and Thanakrit Anutrakulchai

November 27, 2021

1 Contributions

2 Notation

For this report, we use the symbols ϑ, ν, ϕ, ψ to denote the value in radians of the angles of joints 1, 2, 3, and 4, respectively. We denote the position in 3D space of the i^{th} joint with symbols x_i, y_i, z_i, and x_e, y_e, z_e for the position of the end-effector, from any arbitrary left-handeded co-ordinate frame. (We only use the difference between these values so the position of the frame does not matter. We choose a left-hand frame as that made getting positional data from the cameras more straightforward.) We also use the symbols L_1, L_2, L_3, L_4 for the lengths of the respective links. We use short-hand notation for sine and cosine, as in s_γ is to be read as $\sin(\gamma)$ and c_γ is to be read as $\cos(\gamma)$ for angles γ.

3 Joint State Estimation

3.1 Part I —Fixing Joint 1

To calculate the angles of the joints, we first use blob detection to find where the joints are in 3D space. We use thresholding to filter out the positions of the green, yellow, blue, and red blobs which correspond to joints 1, 2, 3, 4 and the end-effector, respectively. As we have two cameras, one facing the yz-plane, and the other facing the xz-plane, we perform this operation on the images received from both camera, and combine their results.

Once we have the necessary joints positions, we then consider the orientation of the links between them, specifically link 3 and link 4. To do so, we calculate unit-length vectors $\mathbf{r}_3(\mathbf{q})$, $\mathbf{r}_4(\mathbf{q})$ pointing from the center of joint 2 to joint 3, and joint 3 to joint 4, respectively. We choose two right-handed co-ordinate frames positioned at the base of the vectors so when all joints are at angle 0, the unit vectors have co-ordinates [0 0 1] in those frames. These vectors are rotated as the joint angles change as so:

$$\mathbf{r}_3(\mathbf{q}) := \begin{pmatrix} r_{3x} \\ r_{3y} \\ r_{3z} \end{pmatrix} = \begin{pmatrix} c_\phi s_\nu \\ -s_\phi \\ c_\phi c_\nu \end{pmatrix} = \frac{1}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2 + (z_3 - z_4)^2}} \begin{pmatrix} x_4 - x_3 \\ y_4 - y_3 \\ z_3 - z_4 \end{pmatrix}$$

$$\mathbf{r}_4(\mathbf{q}) := \begin{pmatrix} r_{4x} \\ r_{4y} \\ r_{4z} \end{pmatrix} = \begin{pmatrix} c_\nu s_\psi + s_\nu c_\phi c_\psi \\ -s_\phi c_\psi \\ -s_\nu s_\psi + c_\nu c_\phi c_\psi \end{pmatrix} = \frac{1}{\sqrt{(x_e - x_4)^2 + (y_e - y_4)^2 + (z_4 - z_e)^2}} \begin{pmatrix} x_e - x_4 \\ y_e - y_4 \\ z_4 - z_e \end{pmatrix}$$

The formulae above can be derived from geometry (e.g. as in the derivation of the spherical co-ordinates in SVCDE) or from multiplying the relevant rotation matrices to transform co-ordinate frames (translations are unnecessary as we are only concerned with orientation). Note that we negate the z component as we switch from a left-handed frame to right-handed frames.

With these values, we can calculate the joint angles as so:

$$\begin{aligned} \nu &= \arctan 2(r_{3x}, r_{3z}) \\ \phi &= \arcsin(-r_{3y}) \\ \psi &= \arcsin(r_{4x} c_\nu - r_{4z} s_\nu) \end{aligned}$$

where we calculate ν before ψ. However, when ϕ is near $\pm\frac{\pi}{2}$, r_{3x} and r_{3z} are near zero, so $\arctan 2(r_{3x}, r_{3z})$ oscillates quickly and rapidly over a short time. In such a case, we switch to calculating $\nu = \arctan 2(r_{4x}c_\phi c_\psi - r_{4z}s_\psi,\ \ r_{4x}s_\psi + r_{4z}c_\phi c_\psi)$, where we use the value of ψ computed from the last iteration. Additionally, when ψ is near 0, as well, we stop calculating ν and keep the angle published constant. This is because we have a gimbal lock situation, any value of ν will produce the same observed set of positions, so we have no information on what ν can be, except that it cannot physically change from its previous value too much.

3.2 Part II —Fixing Joint 2

We follow much of the same methodology from Part I, but with a slight modification as most observed sets of positions of the joints now correspond with two possible set of angles. We calculate the unit-vectors now as:

$$\mathbf{r}_3(\mathbf{q}) := \begin{pmatrix} r_{3x} \\ r_{3y} \\ r_{3z} \end{pmatrix} = \begin{pmatrix} s_\vartheta s_\phi \\ -c_\vartheta s_\phi \\ c_\phi \end{pmatrix} = \frac{1}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2 + (z_3 - z_4)^2}} \begin{pmatrix} x_4 - x_3 \\ y_4 - y_3 \\ z_3 - z_4 \end{pmatrix}$$

$$\mathbf{r}_4(\mathbf{q}) := \begin{pmatrix} r_{4x} \\ r_{4y} \\ r_{4z} \end{pmatrix} = \begin{pmatrix} c_\vartheta s_\psi + s_\vartheta s_\phi c_\psi \\ s_\vartheta s_\psi - c_\vartheta s_\phi c_\psi \\ c_\phi c_\psi \end{pmatrix} = \frac{1}{\sqrt{(x_e - x_4)^2 + (y_e - y_4)^2 + (z_4 - z_e)^2}} \begin{pmatrix} x_e - x_4 \\ y_e - y_4 \\ z_4 - z_e \end{pmatrix}$$

and the angles as:

$$\begin{aligned} \vartheta &= \arctan 2(r_{3x} sgn(\phi), -r_{3y} sgn(\phi)) \\ \phi &= sgn(\phi) \arccos(\phi) \\ \psi &= \arcsin(r_{4x}c_\vartheta + r_{4z}s_\vartheta) \end{aligned}$$

where the sign function $sgn(\phi)$ is -1 if $\phi < 0$, and 1 otherwise. We use the sign of ϕ from the previous iteration for computing ϕ. Once again, we have problems with $\arctan 2$ when ϕ, and hence r_{3x}, r_{3y}, is near 0. But, now we also have problems with $sgn(\phi)$ giving the wrong value (i.e. when ϕ crosses 0 or oscillates near 0). In such cases, we compute $\phi = \arcsin(r_{3x}s_\vartheta - r_{3y}c_\vartheta)$, and $\vartheta = \arctan 2(r_{4x}s_\phi c_\psi + r_{4y}s_\psi,\ \ r_{4x}s_\psi - r_{4y}s_\phi c_\psi)$. Additionally, when ψ is also near zero, we have aa gimbal lock situation, so we keep ϑ, and thus also ϕ, constant during these periods.

Earlier, we noted that a set of joint positions, in general, corresponds to two possible sets of angles. The choice of $sgn(\phi)$ being -1 or 1 at the start picks which two of these sets we first compute. Once we have a set, the set computed in the next iteration is the one consistent with the previously computed set, in that the angle values do not jump discontinuously. There is an exception to this: when ϑ crosses between π and $-\pi$. In this scenario, we realize that we started with the wrong set of angles, and so we switch to the other set by adding or subtracting π from ϑ as appropriate, and negating the other angles.

4 Control

4.1 Forward Kinematics

We derived Forward Kinematics for the robot using the D-H convention.

link	α	a	d	θ
1	$-\frac{\pi}{2}$	0	L_1	ϑ
2	$-\frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
3	$\frac{\pi}{2}$	L_3	0	ϕ
4	0	L_4	0	ψ

$$K(\mathbf{q}) = \begin{pmatrix} c_\vartheta s_\psi L_4 + s_\vartheta s_\phi c_\psi L_4 + s_\vartheta s_\phi L_3 \\ s_\vartheta s_\psi L_4 - c_\vartheta s_\phi c_\psi L_4 - c_\vartheta s_\phi L_3 \\ c_\phi c_\psi L_4 + c_\phi L_3 + L_1 \end{pmatrix}$$

4.2 Inverse Kinematics

$$J(\mathbf{q}) = \begin{pmatrix} -s_\vartheta s_\psi L_4 + c_\vartheta s_\phi c_\psi L_4 + c_\vartheta s_\phi L_3 & s_\vartheta c_\phi c_\psi L_4 + s_\vartheta c_\phi L_3 & c_\vartheta c_\psi L_4 - s_\vartheta s_\phi s_\psi L_4 \\ c_\vartheta s_\psi L_4 + s_\vartheta s_\phi c_\psi L_4 + s_\vartheta s_\phi L_3 & -c_\vartheta c_\phi c_\psi L_4 - c_\vartheta c_\phi L_3 & s_\vartheta c_\psi L_4 + c_\vartheta s_\phi s_\psi L_4 \\ 0 & -s_\phi c_\psi L_4 - s_\phi L_3 & -c_\phi s_\psi L_4 \end{pmatrix}$$