
Table of Contents

Micro Arm Kinematics	1
Dimensions	1
Inverse Kinematics formula	1
Lines of Code in total	3

Micro Arm Kinematics

The Micro Arm robot has 4 rotary joints and a gripper end effector. The joints are as such:

1. Base (rotates about z)
2. Shoulder (rotates about x)
3. Elbow (rotates about x)
4. Wrist (rotates about x)

```
% DH matrix function, for forward kinematics
% t is z rotation
% d is z height
% r is x length
% a is x rotation
DH = @(t, d, r, a) [
    cos(t), -sin(t)*cos(a), sin(t)*sin(a), r*cos(t);
    sin(t), cos(t)*cos(a), -cos(t)*sin(a), r*sin(t);
    0, sin(a), cos(a), d;
    0, 0, 0, 1;
];
```

Dimensions

All dimensions in mm

```
shoulder_height = 37.6; % Distance from z=0 to shoulder joint
upper_arm_len = 60; % from shoulder to elbow
forearm_len = 60; % from elbow to wrist
wrist_len = 45; % from wrist to end effector center
```

Inverse Kinematics formula

We will assume that the angle the wrist makes with the floor is always known, as part of the "function arguments"

We will also assume that the 2-link upper arm and forearm system will always be in the elbow-up configuration, because the elbow down configuration is highly likely to cause collisions between robot geometry. This also helps mitigate the issue of limited micro servo actuation (180 degrees).

```
% Inputs
```

```

% x, y, z in mm from origin
% theta is the angle in radians from the xy-plane to the gripper's horizontal
plane
x = 100; % mm
y = 100; % mm
z = 100; % mm
theta = 0; % radians

% end effector position relative to wrist joint
end_effector_x = 0;
end_effector_y = wrist_len;
end_effector_z = 0;

% Outputs
% q1, q2, q3, q4 are joint angles in radians,
% measured from a zero angle position,
% where qn corresponds to joint n (see first section).

```

Step 1

Find wrist position, which will later allow us to determine q4 by the angle between the elbow and wrist.

The wrist will always be "facing" from the target x, y, z to the origin

```

heading = atan2(y, x); % which direction the robot is facing in the xy plane,
in radians
q1 = heading;
% theta = pitch

% TODO: allow the end effector to impact this reading, since the end
effector will not always be at the same location

wrist_delta_x = -wrist_len*cos(heading)*cos(theta);
wrist_delta_y = -wrist_len*sin(heading)*cos(theta);
wrist_delta_z = -wrist_len*sin(theta);
wrist_x = x+wrist_delta_x;
wrist_y = y+wrist_delta_y;
wrist_z = z+wrist_delta_z;

```

Step 2

Now that we have the wrist position, we can solve the standard two-link ik for the forearm and upper arm.

```

shoulder_x = 0;
shoulder_y = 0;
shoulder_z = shoulder_height;

% calculate "x" and "y" in vertical plane that goes from wrist to shoulder
two_link_x = sqrt((wrist_x-shoulder_x)^2 + (wrist_y-shoulder_y)^2);
two_link_y = wrist_z - shoulder_z;
D = (two_link_x^2 + two_link_y^2 - upper_arm_len^2 - forearm_len^2) ...
    / (2*upper_arm_len*forearm_len);
if D > 1
    fprintf("IK is unsolvable: D=%f", D);
end

```

```
q3 = atan2(+sqrt(1-D^2),D); % review order of Atan2 with code from last
semester
q2 = atan2(two_link_y, two_link_x) - atan2(forearm_len*sin(q3), upper_arm_len
+ forearm_len*cos(q3));
q4 = theta-q3-q2;

joint_positions = [q1 q2 q3 q4]

joint_positions =

    0.7854    0.2804    0.5880   -0.8684
```

Lines of Code in total

After finishing the project, I counted approximately how many lines of code are in each file which was used in the final project. This estimate is inflated, but not by much.

```
loc =
130+108+83+105+51+154+135+25+141+38+73+118+47+30+635+816+499+29+22+78+95+135+14+18+101

loc =

3680
```

Published with MATLAB® R2022b