# Inverse Kinematics Report

Valmik Prabhu

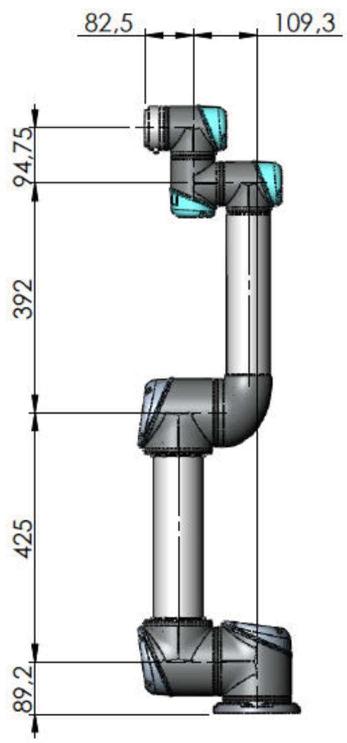
December 16, 2016

## 1 Objective

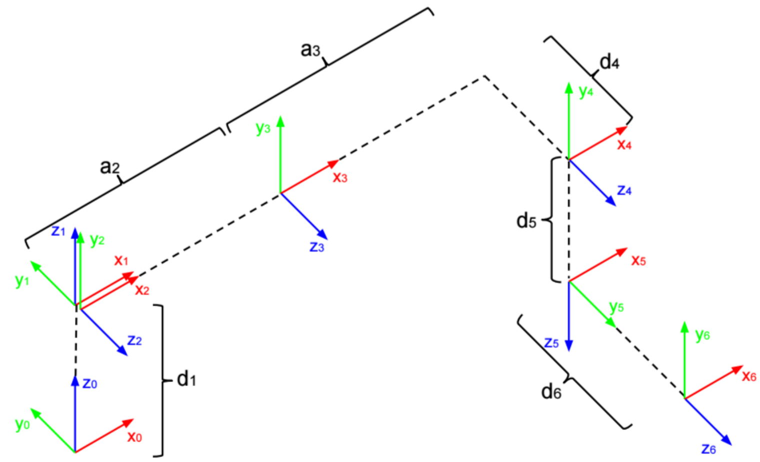
We want to accurately control the position of the end effector, and maintain the location of its $Z$ axis (ensure no rotation about $X$ or $Y$).

## 2 Basics

The UR5 is a 6-axis manipulator with two shoulder joints, an elbow, and three wrist joints.




However, as you can see below, initial state is not ideal for our application, and the end effector orientation is different from that of the base frame.




Therefore, we define the initial state as

$$\theta = [0, 0, 0, -90°, -90°, 0]$$

And

$$g_st(0) = \begin{bmatrix} 0 & 1 & 0 & -0.9118 \\ 1 & 0 & 0 & -0.1093 \\ 0 & 0 & -1 & 0.0067 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We elected to use Paden Kahan subproblems, rather than dialytical elimination because we know the former. However, because the axes of the three wrist joints $\omega_4, \omega_5, \omega_6$ don't intersect at a point, this problem isn't realizable with pure PK subproblems.

However, because we don't need to change joints 5 or 6 at all, we can technically solve for only the first four joints. This will solve for the angles precisely if given the proper end effector configuration.

## 3 End Effector Configuration

The end effector configuration is unfortunately not easy to find. Because we aren't changing joint 6, the coordinate frame of the end effector will change in relation to the base frame during a motion in $Y$. We attempt to account for this rotation by manufacturing it.

Given a desired end effector position $[x_1, y_1, z_1]$ and an initial end effector position $[x_0, y_0, z_0]$, we define a rotation about $Z$

$$\theta = \arctan 2(y_1, x_1) - \arctan 2(y_0, x_0)$$

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$g_R = \begin{bmatrix} \boldsymbol{R} & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

$$g_d = g_st(0)g_R$$

The output configuration is quite similar, but still distinctly different, and leads to wrong values of $\theta_4$ (and therefore the wrong orientation). As $\theta_1$ increased, the approximation grew worse and worse. Unfortunately

I'm writing this on my personal computer, and therefore can't run the kinematics code. We ended up running with this, and used the tactics below to repair it.

# 4 The Math

We ignore joints 5 and 6.

We have 4 points of interest:

- $p_f$ end effector
- $p_w$ wrist
- $p_e$ elbow
- $p_s$ shoulder

We start with

$$g_d = g_{st}(\theta) = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}e^{\hat{\xi}_4\theta_4}g_{st}(0) \tag{1}$$

Now multiply by $g_{st}^{-1}(0)$ to get

$$g_1 = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}e^{\hat{\xi}_4\theta_4} = g_d g_{st}^{-1}(0) \tag{2}$$

Now let's multiply the the equation by $p_w$

$$g_1 p_{w2} = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}e^{\hat{\xi}_4\theta_4}p_w \tag{3}$$

Because $p_w$ is on the axes of $\xi_4$, this simplifies to

$$g_1 p_w = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}p_w \tag{4}$$

Now we subtract $p_s$ or the shoulder point. We get

$$g_1 p_w - p_s = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3}p_w - p_s \tag{5}$$

Since $p_s$ is on the intersection of $\xi_1$ and $\xi_2$, we can factor it like so

$$g_1 p_w - p_s = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}\left(e^{\hat{\xi}_3\theta_3}p_w - p_s\right) \tag{6}$$

Since the distance between points is preserved by a rigid body transform, we can take the magnitude to get

$$\|g_1 p_w - p_s\| = \left\|e^{\hat{\xi}_3\theta_3}p_w - p_s\right\| \tag{7}$$

This is a PK3 problem. We have

$$\begin{gathered} p = p_w \\ q = p_s \\ \delta = \|g_1 p_w - p_s\| \\ \omega \text{ comes from } \xi_3 \\ r = p_e \end{gathered}$$

Now that we've solved $\theta_3$, we can solve for $\theta_1$ and $\theta_2$. We add back the location of the base to get back to

$$g_1 p_w = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}\left(e^{\hat{\xi}_3\theta_3}p_w\right) \tag{8}$$

Since we know know $\theta_3$, we can set do a PK2 problem with

$$p = e^{\hat{\xi}_3\theta_3} p_w$$
$$q = g_1 p_w$$
$$\omega_1 \text{ comes from } \xi_1$$
$$\omega_2 \text{ comes from } \xi_2$$
$$r = p_s$$

Now we're left with

$$e^{\hat{\xi}_4\theta_4} = e^{-\hat{\xi}_3\theta_3} e^{-\hat{\xi}_2\theta_2} e^{-\hat{\xi}_1\theta_1} g_1 = g_2 \tag{9}$$

We only have one point left, so we can choose any point (let's choose $p_f$) and get

$$e^{\hat{\xi}_4\theta_4} p_f = g_2 p_f \tag{10}$$

This is a PK1 problem. We have

$$p = p_f$$
$$q = g_2 p_f$$
$$\omega \text{ comes from } \xi_4$$
$$r = p_w$$

# 5 Fixes

Because our approximation didn't actually work, we manually corrected $\theta_4 = -\theta_2 - \theta_3$. This fixed the orientation but rendered the position incorrect. To fix this, we ran the kinematics in a while loop and adjusted the parameters until the end result worked. The pseudocode is below:

```
given q_desired
q_ik = q_desired
while true:
    theta = ik(q_ik)
    q_current = fk(theta)
    if q_current - q_desired < 0.001
        break
    else
        q_ik = q_ik - q_current + q_desired
end
```

With this, we were able to solve inverse kinematics with millimeter precision without resorting to more intensive methods like dialytical elimanation.