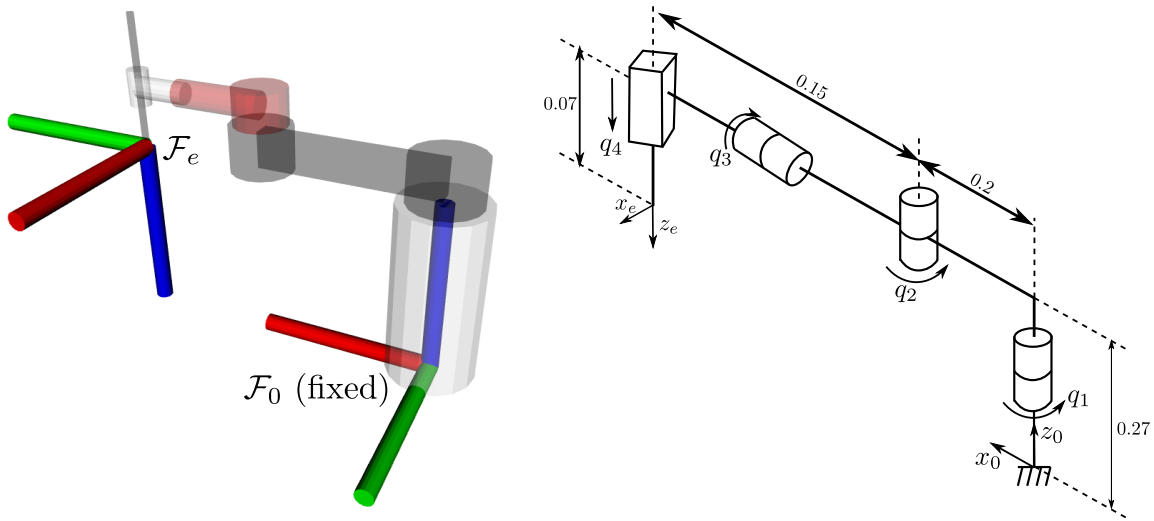


# Manipulator Modeling & Control

## Example for the RRRP robot

### Description of the robot

The considered robot is the one seen during the lectures:



As we saw in class, the MDH table is as follows:

Joint	$\alpha_i$	$a_i$	$\theta_i$	$r_i$
1	0	0	$q_1$	$r_1$
2	0	$a_2$	$q_2 + \pi/2$	0
3	$\pi/2$	0	$q_3$	$r_3$
4	$\pi/2$	0	0	$q_4$
e	0	0	0	$r_4$

with values:

$r_1$	0.27
$a_2$	0.2
$r_3$	0.15
$r_4$	0.07

The wrist-to-end effector transform is:

$${}^w\mathbf{M}_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And the Direct Geometric model is:

$${}^0\mathbf{M}_4 = {}^f\mathbf{M}_w = \begin{bmatrix} -s_{12}c_3 & c_{12} & -s_3s_{12} & a_2c_1 - q_4s_3s_{12} + r_3c_{12} \\ c_3c_{12} & s_{12} & s_3c_{12} & a_2s_1 + q_4s_3c_{12} + r_3s_{12} \\ s_3 & 0 & -c_3 & -q_4c_3 + r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We of course have  ${}^f\mathbf{M}_e = {}^f\mathbf{M}_w {}^w\mathbf{M}_e$

## Solving the inverse model

When solving the inverse model, we want to find  $(q_1, q_2, q_3, q_4)$  that solve:

$$\begin{bmatrix} -s_{12}c_3 & c_{12} & -s_3s_{12} & a_2c_1 - q_4s_3s_{12} + r_3c_{12} \\ c_3c_{12} & s_{12} & s_3c_{12} & a_2s_1 + q_4s_3c_{12} + r_3s_{12} \\ s_3 & 0 & -c_3 & -q_4c_3 + r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_x & y_x & z_x & t_x \\ x_y & y_y & z_y & t_y \\ x_z & y_z & z_z & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^* = {}^f\mathbf{M}_w^* \quad (1)$$

Assuming that the right-hand side matrix is given with numerical values.

In order to do so, we have to identify remarkable equations types.

Terms of equation (1) can be obtained in C++ through:

```
const auto [xx,xy,xz,yx,yy,yz,zx,zy,zz,tx,ty,tz] = explodeMatrix(fMe_des);
```

where the `fMe_des` is the numerical value of  ${}^f\mathbf{M}_e^*$ .

It will first apply  ${}^f\mathbf{M}_w^* = {}^f\mathbf{M}_e^* \mathbf{M}_w$ , then decompose it into all 12 values.

### Solving for $q_3$

In (1) we notice:  $\begin{cases} s_3 = x_z \\ -c_3 = z_z \end{cases}$

This can be written as a Type 3 equation:

$$\begin{cases} X_1s_3 + Y_1c_3 = Z_1 \\ X_2s_3 + Y_2c_3 = Z_2 \end{cases} \quad \text{with} \quad \begin{cases} X_1 = 1, & Y_1 = 0, & Z_1 = xz \\ X_2 = 0, & Y_2 = -1, & Z_2 = zz \end{cases}$$

We can thus solve it for  $q_3$  with the following syntax:

```
for(auto q3: solveType3(1, 0, xz, 0, -1, zz))
{
    // q3 is a valid solution, can be used to find other joints
}
```

### Solving for $q_1$ and $q_4$

From a valid value for  $q_3$ , it is tempting to use  $t_z$  to solve  $q_4$ . Indeed it would write:

$$q_4 = \frac{r_1 - t_z}{\cos(q_3)}$$

Unfortunately this only works if  $\cos(q_3) \neq 0$ .

On the opposite, we notice that  $(t_x, t_y)$  form a system of two unknowns  $(q_1, q_4)$ :

$$\begin{cases} a_2c_1 - q_4s_3s_{12} + r_3c_{12} = t_x \\ a_2s_1 + q_4s_3c_{12} + r_3s_{12} = t_y \end{cases} \quad (2)$$

While we do not know the values of  $q_1$  and  $q_2$ , we know from (1) the values of  $s_{12}$  and  $c_{12}$ .

This makes (2) a Type 5 equation:

$$\begin{cases} X_1s_1 = Y_1 + Z_1q_4 \\ X_2c_1 = Y_2 + Z_2q_4 \end{cases} \quad \text{with} \quad \begin{cases} X_1 = a_2, & Y_1 = t_y - r_3s_{12}, & Z_1 = -c_{12}s_3 \\ X_2 = a_2, & Y_2 = t_x - r_3c_{12}, & Z_2 = s_{12}s_3 \end{cases}$$

This is solved in practice with:

```
const auto s3 = sin(q3); // we are inside the q3 loop so we know sin(q3)

for(auto q14: solveType5(a2, ty-r3*yy, -yx*s3, a2, tx-r3*yx, yy*s3))
{
    auto q1 = q14.qi; // extract joint i = 1
    auto q4 = q14.qj; // extract joint j = 4
}
```

### Solving for $q_2$ and adding the candidate solution

The system  $(y_x, y_y)$  will give the solutions of  $q_1 + q_2$  from a Type 3 equation. Then  $q_2$  can easily be computed as we now have  $q_1$ . As we have a full candidate, we can add it to the potential solutions.

```
for(auto q12: solveType3(0, 1, yx, 1, 0, yy))
{
    auto q2 = q12 - q1;
    addCandidate({q1, q2, q3, q4});
}
```