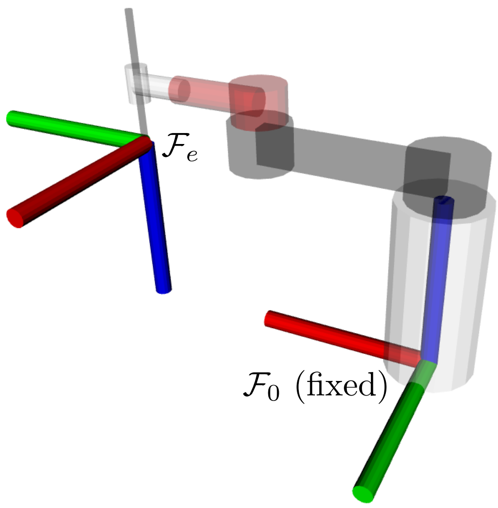
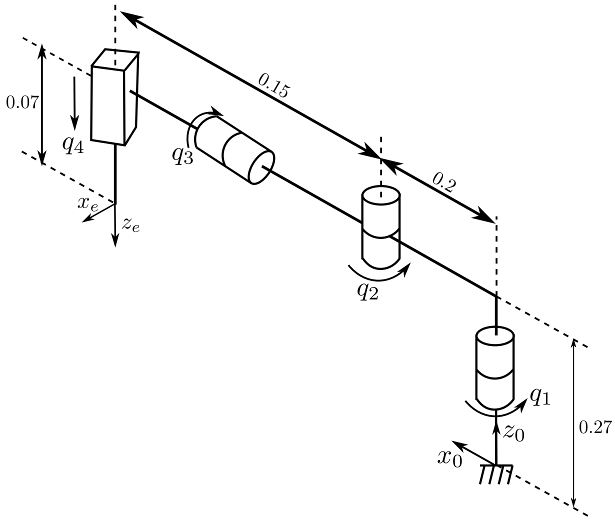
Manipulator Modeling & Control

Example for the RRRP robot

1 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	α_i	a_i	θ_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 ${}^{f}\mathbf{M}_e = {}^{f}\mathbf{M}_w {}^{w}\mathbf{M}_e$

2 Solving the inverse model

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

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

Assuming that the right-hand side matrix is given with numerical values.
In order to do so, we have to identify remarquable 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 ${}^f\mathbf{M}_w^* = {}^f\mathbf{M}_e^{*e}\mathbf{M}_w$, 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:

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

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 $q_{12} = q_1 + q_2$

The system (y_x, y_y) will give the solutions of $q_1 + q_2$ from a Type 3 equation. This solution will be used later to get q_2 when we know q_1:

```
for(auto q12: solveType3(0, 1, yx, 1, 0, yy)
 {
    // q12 is q1+q2, let us carry on in this loop to get q1 and q4

 }
```

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. 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 unkowns (q_1, q_4). Similarly, this only works when $\sin(q_3) \neq 0$ as in the other case q_4 would disappear from the equation.
We thus have to pick the best choice: rely c_3 or s_3. They cannot be both close to 0, so a simple choice is to use the one having the greatest absolute value.

2.0.1 If $|\cos(q_3)| > |\sin(q_3)|$: solve q_4 first

In this case we can use t_z directly:

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

Then q_1 is now the only unknown in (t_x, t_y), which makes it a Type 3 system for q_1.

2.0.2 If $|\cos(q_3)| < |\sin(q_3)|$: solve (q_1, q_4) together

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 (t_x, t_y) a Type 5 equation:

$$\left\{\begin{array}{l} X_1 s_1 = Y_1 + Z_1 q_4 \\ X_2 c_1 = Y_2 + Z_2 q_4 \end{array}\right. \quad \text{with} \left\{\begin{array}{lll} X_1 = a_2, & Y_1 = t_y - r_3 s_{12}, & Z_1 = -c_{12} s_3 \\ X_2 = a_2, & Y_2 = t_x - r_3 c_{12}, & Z_2 = s_{12} s_3 \end{array}\right.$$

This is solved in practice with:

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

for(auto [q1,q4]: solveType5(a2, ty-r3*yy, -yx*s3, a2, tx-r3*yx, yy*s3))
{
    // continue with this solution for q1 and q4

}
```

Solving for $\mathbf{q_2}$ and adding the candidate solution

At this point we have a candidate for (q_1, q_3, q_4) and we have the value of $q_{12} = q_1 + q_2$. q_2 can this easily be computed, that makes a valid candidate solution to consider:

```
const auto q2 = q12 - q1;
addCandidate({q1, q2, q3, q4});
```