FMU Library Catalogue

Serhat Kahraman

18.06.2025

v0.3

Model Name: UR10e Cobot

- FMU Name: UR10e.fmu
- FMI Standard: v2.0.2

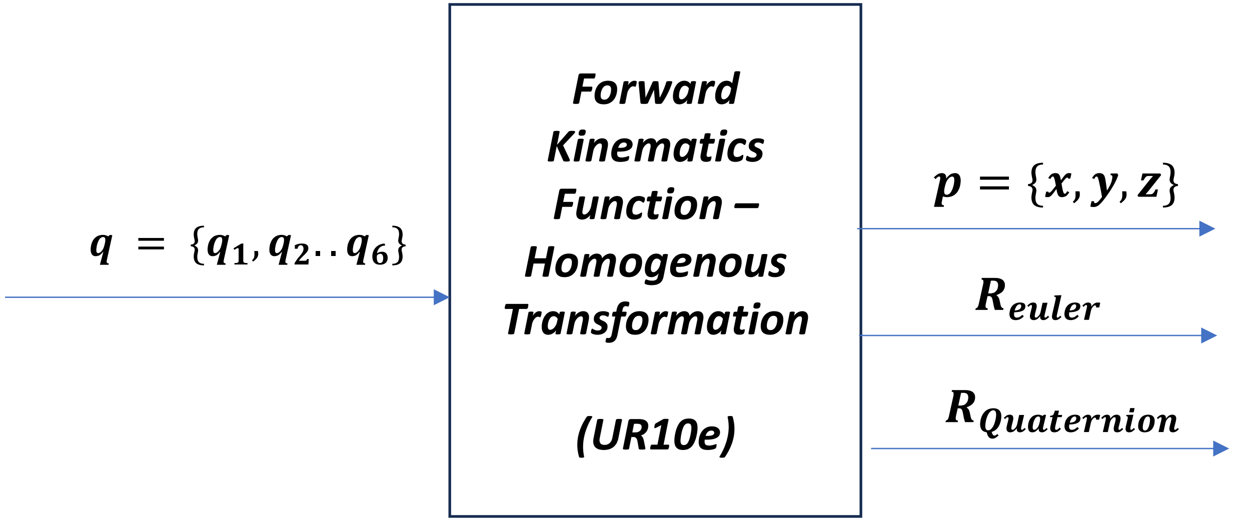
Source:UR10e

- FMU Types - UR10e Cobot:
  - Forward Kinematics
  - Inverse Kinematics
  - Dynamics

Model Name: UR10e Robot Forward Kinematics

Model Block Diagram

UR10eRobotManipulatorFKinematics.fmu



Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

Table 7. Six-dof Universal Robot URe-Series DH Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	90°	0	0	$\theta_2 + 90^\circ$
3	0	a_2	0	θ_3
4	0	a_3	d_4	$\theta_4 - 90^\circ$
5	-90°	0	d_5	θ_5
6	90°	0	0	θ_6

Table 8. Specific DH Lengths in the Six-dof URe-Series (mm)

Parameter	UR3e	UR5e	UR10e	UR16e
L_B	152	163	181	170
a_2	244	425	613	476
a_3	213	392	572	361
d_4	131	133	174	194
d_5	85	100	120	120
L_{TP}	92	100	117	112

Source:
R.L. Williams II, “Universal Robot URe-Series Kinematics”, Internet Publication, https://www.ohio.edu/mechanical-faculty/williams/html/pdf/UniversalKinematics.pdf, January 2024.

Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

$$\left[{}^0_1T\right]=\begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left[{}^1_2T\right]=\begin{bmatrix} -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left[{}^2_3T\right]=\begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^3_4T\right]=\begin{bmatrix} s_4 & c_4 & 0 & a_3 \\ -c_4 & s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left[{}^4_5T\right]=\begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \left[{}^5_6T\right]=\begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used: $c_i=\cos\theta_i$, $s_i=\sin\theta_i$, for $i=1,2,\cdots,6$.

The basic Universal Cobot FPK solution is found from the following homogeneous transform equation to derive the active-joints FPK result.

$$\left[{}^0_6T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\left[{}^0_1T(\theta_1)\right]\left[{}^1_2T(\theta_2)\right]\left[{}^2_3T(\theta_3)\right]\left[{}^3_4T(\theta_4)\right]\left[{}^4_5T(\theta_5)\right]\left[{}^5_6T(\theta_6)\right]$$

Since Cartesian coordinate frames {2}, {3}, and {4} have parallel Z axes, the basic Universal Cobot FPK solution should be grouped as follows.

$$\left[{}^0_6T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\left[{}^0_1T(\theta_1)\right]\left[{}^1_4T(\theta_2,\theta_3,\theta_4)\right]\left[{}^4_6T(\theta_5,\theta_6)\right]$$

where $\left[{}^0_1T(\theta_1)\right]$ was given above, and the other two matrices are found by matrix multiplication and simplification:

$$\left[{}^1_4T(\theta_2,\theta_3,\theta_4)\right]=\begin{bmatrix} c_{234} & -s_{234} & 0 & -a_2s_2-a_3s_{23} \\ 0 & 0 & -1 & -d_4 \\ s_{234} & c_{234} & 0 & a_2c_2+a_3c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^4_6T(\theta_5,\theta_6)\right]=\begin{bmatrix} c_5c_6 & -c_5s_6 & s_5 & 0 \\ s_6 & c_6 & 0 & d_5 \\ -s_5c_6 & s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used:

$$c_{23}=\cos(\theta_2+\theta_3) \qquad c_{234}=\cos(\theta_2+\theta_3+\theta_4)$$
$$s_{23}=\sin(\theta_2+\theta_3) \qquad s_{234}=\sin(\theta_2+\theta_3+\theta_4)$$

Source:
R.L. Williams II, “Universal Robot URe-Series Kinematics”, Internet Publication, https://www.ohio.edu/mechanical-faculty/williams/html/pdf/UniversalKinematics.pdf, January 2024.

Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

Any time consecutive Z axes are parallel in a serial robot we can expect such sum-of-angles simplifications. Now we can find the basic active-joints FPK solution $\left[{}^0_6T\right]$ by more matrix multiplications (but no more trigonometric simplifications):

$$\left[{}^0_6T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^0x_6 \\ r_{21} & r_{22} & r_{23} & {}^0y_6 \\ r_{31} & r_{32} & r_{33} & {}^0z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The orthonormal rotation matrix elements for this FPK result are:

$$\begin{aligned} r_{11} &= -s_1 s_5 c_6 + c_1(-s_{234}s_6 + c_{234}c_5c_6) \\ r_{21} &= c_1 s_5 c_6 + s_1(-s_{234}s_6 + c_{234}c_5c_6) \\ r_{31} &= c_{234}s_6 + s_{234}c_5c_6 \end{aligned}$$

$$\begin{aligned} r_{12} &= s_1 s_5 s_6 - c_1(s_{234}c_6 + c_{234}c_5s_6) \\ r_{22} &= -c_1 s_5 s_6 - s_1(s_{234}c_6 + c_{234}c_5s_6) \\ r_{32} &= c_{234}c_6 - s_{234}c_5s_6 \end{aligned}$$

$$\begin{aligned} r_{13} &= s_1c_5 + c_1c_{234}s_5 \\ r_{23} &= -c_1c_5 + s_1c_{234}s_5 \\ r_{33} &= s_{234}s_5 \end{aligned}$$

And the FPK translational vector components giving the position of the origin of {6} with respect to the origin of {0}, expressed in the basis of {0} are:

$$\begin{aligned} {}^0x_6 &= d_4s_1 - c_1(a_2s_2 + a_3s_{23} + d_5s_{234}) \\ {}^0y_6 &= -d_4c_1 - s_1(a_2s_2 + a_3s_{23} + d_5s_{234}) \\ {}^0z_6 &= a_2c_2 + a_3c_{23} + d_5c_{234} \end{aligned}$$

Source:
R.L. Williams II, “Universal Robot URe-Series Kinematics”, Internet Publication, https://www.ohio.edu/mechanical-faculty/williams/html/pdf/UniversalKinematics.pdf, January 2024.

Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

Determining yaw, pitch, and roll from a rotation matrix

It is often convenient to determine the α, β, and γ parameters directly from a given rotation matrix. Suppose an arbitrary rotation matrix

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{3.43}$$

is given. By setting each entry equal to its corresponding entry in (3.42), equations are obtained that must be solved for α, β, and γ. Note that $r_{21}/r_{11} = \tan\alpha$ and $r_{32}/r_{33} = \tan\gamma$. Also, $r_{31} = -\sin\beta$ and $\sqrt{r_{32}^2 + r_{33}^2} = \cos\beta$. Solving for each angle yields

$$\alpha = \tan^{-1}(r_{21}/r_{11}), \tag{3.44}$$

$$\beta = \tan^{-1}\left(-r_{31}/\sqrt{r_{32}^2 + r_{33}^2}\right), \tag{3.45}$$

and

$$\gamma = \tan^{-1}(r_{32}/r_{33}). \tag{3.46}$$

Source:[Determining yaw, pitch, and roll from a rotation matrix]

Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

Convert Rotation Matrix to Quaternion

Given the rotation matrix R:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{8}$$

We can find the equivalent quaternion using two steps.

Step 1: Find the magnitude of each quaternion component. This leaves the sign of each component undefined:

$$|q_0| = \sqrt{\frac{1 + r_{11} + r_{22} + r_{33}}{4}} \tag{9a}$$

$$|q_1| = \sqrt{\frac{1 + r_{11} - r_{22} - r_{33}}{4}} \tag{9b}$$

$$|q_2| = \sqrt{\frac{1 - r_{11} + r_{22} - r_{33}}{4}} \tag{9c}$$

$$|q_3| = \sqrt{\frac{1 - r_{11} - r_{22} + r_{33}}{4}} \tag{9d}$$

Step 2: To resolve the signs, find the largest of q_0, q_1, q_2, q_3 and assume its sign is positive. Then compute the remaining components as shown in the table below. Taking the largest magnitude avoids division by small numbers, which would reduce numerical accuracy.

If q_0 is largest:	If q_1 is largest:	If q_2 is largest:	If q_3 is largest:
$q_1 = \frac{r_{32} - r_{23}}{4q_0}$	$q_0 = \frac{r_{32} - r_{23}}{4q_1}$	$q_0 = \frac{r_{13} - r_{31}}{4q_2}$	$q_0 = \frac{r_{21} - r_{12}}{4q_3}$
$q_2 = \frac{r_{13} - r_{31}}{4q_0}$	$q_2 = \frac{r_{12} + r_{21}}{4q_1}$	$q_1 = \frac{r_{12} + r_{21}}{4q_2}$	$q_1 = \frac{r_{13} + r_{31}}{4q_3}$
$q_3 = \frac{r_{21} - r_{12}}{4q_0}$	$q_3 = \frac{r_{13} + r_{31}}{4q_1}$	$q_3 = \frac{r_{23} + r_{32}}{4q_2}$	$q_2 = \frac{r_{23} + r_{32}}{4q_3}$

The reason the sign is ambiguous is that any given rotation has two possible quaternion representations. If one is known, the other can be found by taking the negative of all four terms. This has the effect of reversing both the rotation angle and the axis of rotation. So for all rotation quaternions, (q_0, q_1, q_2, q_3) and $(-q_0, -q_1, -q_2, -q_3)$ produce identical rotations. To convert from a rotation matrix to a quaternion, we must arbitrarily pick one of the two possible answers as described in steps 1 and 2.

Source:[Quaternions]

Model Name: UR10e Robot Forward Kinematics (FK)

UR10e FK Model

1. Unit quaternions

A quaternion, customarily denoted by q, is a 4-tuple of real numbers. It can be decomposed into a scalar component $q_0 \in \mathbb{R}$ and a vector component $\boldsymbol{q} \in \mathbb{R}^3$, i.e.

$$q = (q_0, \boldsymbol{q}) = (q_0, (q_i, q_j, q_k))$$

A unit quaternion is a quaternion with unit norm ($\|q\|^2 = q_0^2 + \|\boldsymbol{q}\|^2 = 1$). Rotations can be parametrized by unit quaternions. Euler's rotation theorem states that any rotation in $\mathbb{R}^3$ can be represented as a rotation of angle θ about an axis $\boldsymbol{n}$ ($\boldsymbol{n}$ is the **unit vector** along the rotation axis). Therefore, a unit quaternion can also be written in terms of θ and $\boldsymbol{n}$:

$$q = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\boldsymbol{n}\right)$$

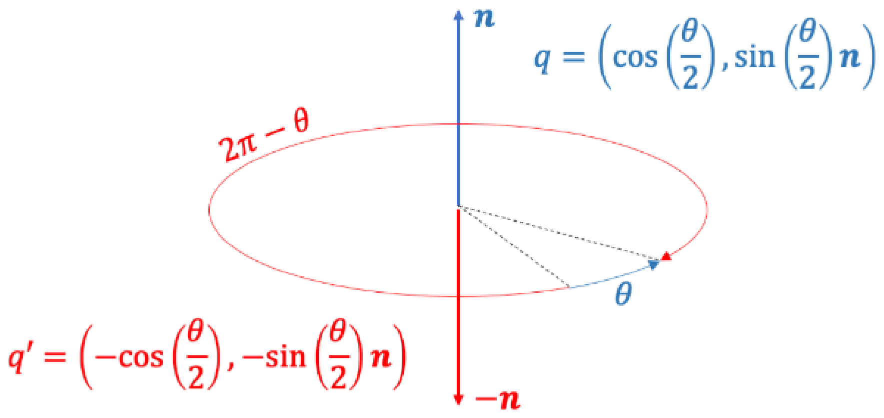
1.1 Double coverage

Note that an rotation of $2\pi - \theta$ about axis $-\boldsymbol{n}$ has the same effect as a rotation of θ about $\boldsymbol{n}$, in terms of quaternions, this observation can be written as

$$\begin{aligned} q' &= \left(\cos\frac{2\pi-\theta}{2}, \sin\frac{2\pi-\theta}{2}(-\boldsymbol{n})\right) \\ &= \left(\cos\pi - \frac{\theta}{2}, \sin\pi - \frac{\theta}{2}(-\boldsymbol{n})\right) \\ &= \left(-\cos\frac{\theta}{2}, -\sin\frac{\theta}{2}\boldsymbol{n}\right) \\ &= -q \end{aligned}$$

Therefore, q and $-q$ represent the same rotation. This is sometimes called the **double coverage** property of quaternions. A quaternion q with a non-negative scalar part ($q_0 \geq 0$) is called **canonical**. Although representing the same rotation, canonical quaternions take the shorter path ($\theta \leq \pi$) compared to their non-canonical counterparts.

Figure : double coverage of quaternions: q and $-q$ represent the same rotation. The blue quaternion is canonical.



Source: pset5_rotation

UR10e								
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	**Dynamics**	Mass [kg]	Center of Mass [m]	Inertia Matrix
Joint 1	0	0	0.1807	π/2	Link 1	7.369	[0.021, 0.000, 0.027]	[0.0341, 0.0000, -0.0043, 0.0000, 0.0353, 0.0001, -0.0043, 0.0001, 0.0216]
Joint 2	0	-0.6127	0	0	Link 2	13.051	[0.38, 0.000, 0.158]	[0.0281, 0.0001, -0.0156, 0.0001, 0.7707, 0.0000, -0.0156, 0.0000, 0.7694]
Joint 3	0	-0.57155	0	0	Link 3	3.989	[0.24, 0.000, 0.068]	[0.0101, 0.0001, 0.0092, 0.0001, 0.3093, 0.0000, 0.0092, 0.0000, 0.3065]
Joint 4	0	0	0.17415	π/2	Link 4	2.1	[0.000, 0.007, 0.018]	[0.0030, -0.0000, -0.0000, -0.0000, 0.0022, -0.0002, -0.0000, -0.0002, 0.0026]
Joint 5	0	0	0.11985	-π/2	Link 5	1.98	[0.000, 0.007, 0.018]	[0.0030, -0.0000, -0.0000, -0.0000, 0.0022, -0.0002, -0.0000, -0.0002, 0.0026]
Joint 6	0	0	0.11655	0	Link 6	0.615	[0, 0, -0.026]	[0.0000, 0.0000, -0.0000, 0.0000, 0.0004, 0.0000, -0.0000, 0.0000, 0.0003]

Link: https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/?ysclid=m9jvxr13yl422421043

REFERANSLAR

- Kaynak: https://github.com/ESOGU-SRLAB