Universal Robot URe-Series Cobot Kinematics & Dynamics
© 2024 Dr. Bob Productions
Robert L. Williams II, Ph.D., williar4@ohio.edu

Mechanical Engineering, Ohio University, January 2024

www.universal-robots.com

For referencing this document, please use:

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

Universal Robot URe-Series Cobot Kinematics & Dynamics

© 2024 Dr. Bob Productions

Robert L. Williams II, Ph.D., williar4@ohio.edu
Mechanical Engineering, Ohio University, Athens, Ohio, USA

Table of Contents

1. Introduction

Universal Robots was founded in 2005 in Denmark, with 2 goals: provide robotic automation to small and medium companies, and produce Cobots (**Co**llaborative Ro**bots**) that are safe to operate in the same workspace as human workers. To date, 46,000 Universal robots have been sold. Their basic serial robot arm product is the URe-Series (Universal Robot) as shown in Figure 1. The URe-Series of serial robot arms is also available (**e**mpowering, **e**ase of use, **e**veryone and **e**volution). Improvements of the URe-Series include better positioning repeatability, built-in force/torque sensor, and more safety features. This document presents kinematics and dynamics equations for control of the URe-Series cobot. The series of 4 Universal cobots all share the same kinematics and joint design (the differences are in size and strength).

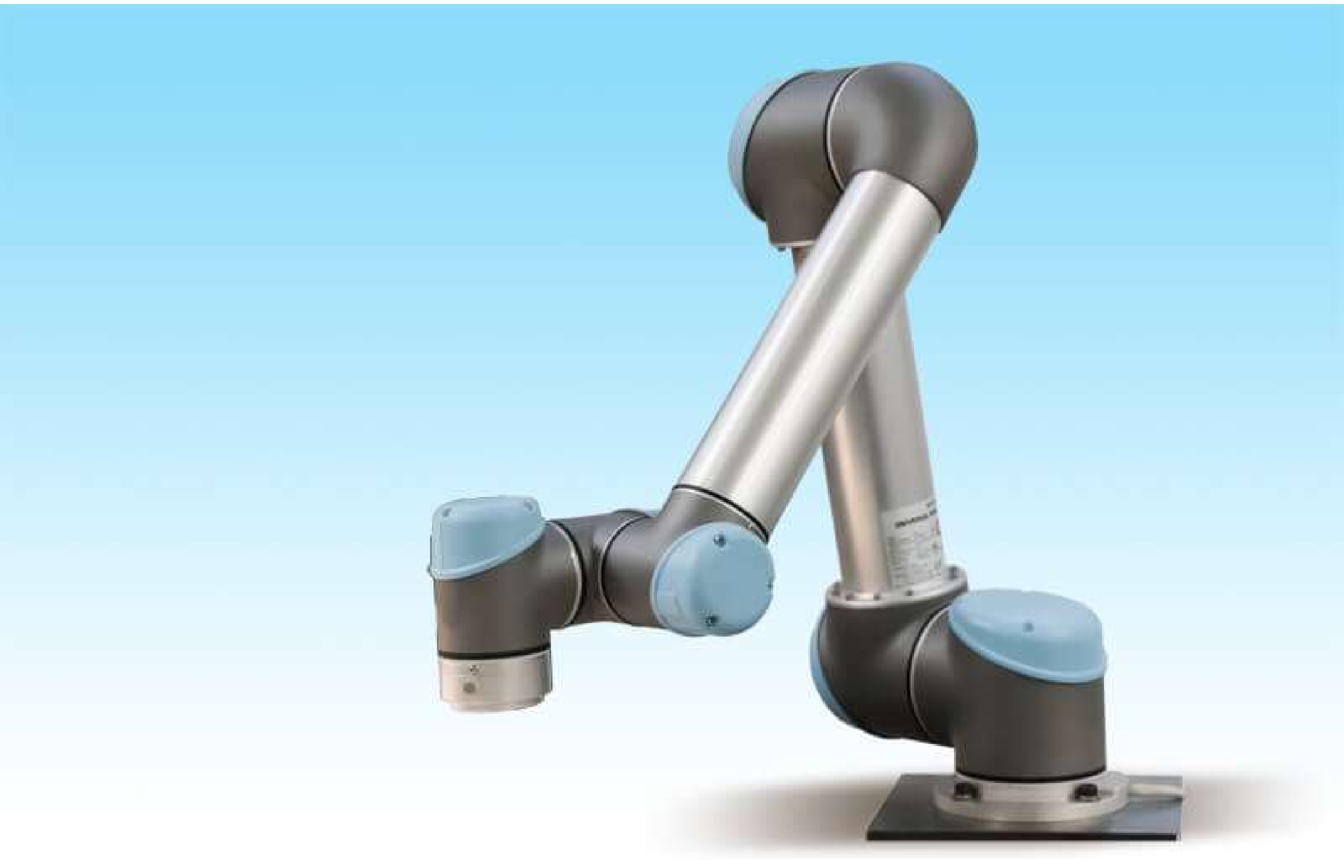


Figure 1. Universal Robot UR3 Cobot

www.universal-robots.com

The Universal Robots website lists the following industrial/automation/manufacturing applications for Cobots: assembly, dispensing, finishing, gluing, injection molding, lab analysis, machine tending, material handling, material removal, packaging, palletizing, pick-and-place, quality inspection, screw-driving, and welding. Cobots are a relatively new paradigm in industrial and service robots where the robot is designed and programmed to safely interact with humans directly in their workspace. Traditional autonomous robots usually work alone, behind safety fencing and interlocks, without human supervision. Cobots are intended to assist and guide humans in manufacturing tasks, responding directly to and moving with human actions, pictured in Figure 2.

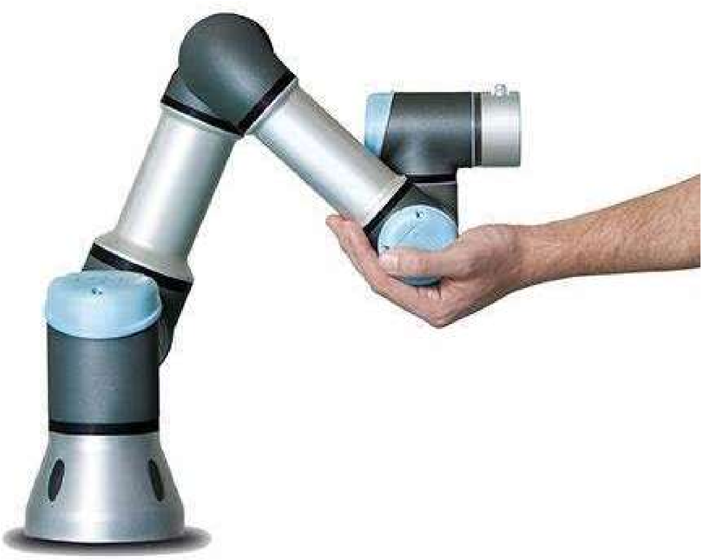


Figure 2. Cobot Intended to Work with Humans

www.universal-robots.com

Presented is a description of the Universal Robot URe-Series of Cobots, followed by kinematics analysis and equations including Forward Pose Kinematics (FPK) and Inverse Pose Kinematics (IPK) expressions and solutions. Numerical examples are given for both FPK and IPK with both snapshots and trajectories. The workspace of this serial robot arm is presented. The velocity equations and Jacobian matrix are also derived and used in a resolved-rate control scheme which has many advantages over IPK-based control. Singularity analysis is also presented. Then Newton-Euler numerical recursion is presented to solve the inverse dynamics problem.

2. URe-Series Cobot Description

As seen in Figure 3, the kinematic structure of all 4 members of the Universal Robot URe-Series (also the original URe-Series) is identical, composed of six R-joints in series. At first glance the URe-Series of cobots appear rather traditional in design, similar to many previous 6-dof serial robot arms: it consists of a waist-shoulder-elbow arm followed by an offset pitch-yaw-roll wrist.

Here are some unique aspects of the URe-Series designs. The workspace is large, even for a serial robot, relative to its size, due to significant $\pm 360^\circ$ motion range on all six joints (plus unlimited bidirectional rotation for the sixth joint of the UR3e). The joints are offset in order to allow as much of the generous joint limits motion as possible. Also, the joints are modular (more on this later). There is an unfortunate offset in the 3-dof wrist, so that all three wrist joints do not share a common origin. However, an advantage in the design involves the 3 parallel Z axes in the consecutive 2nd, 3rd, and 4th revolute joints (this advantage is not unique to URe-Series robots).

Table 1 presents the reach, payload, footprint, mass, and payload-to-mass ratio for each member of the UR3-series of cobots (**UR3e**, **UR5e**, **UR10e**, and **UR16e**). We see that the size, reach, maximum payload capacity, and robot mass increase for the **UR3e**, **UR5e**, and **UR10e** robots. However, the **UR16e** robot can handle a significantly higher payload, though it is smaller than the **UR10e** robot, using the identical motors to the **UR10e** robot. Evidently the number in the series name is the maximum payload mass in kg.

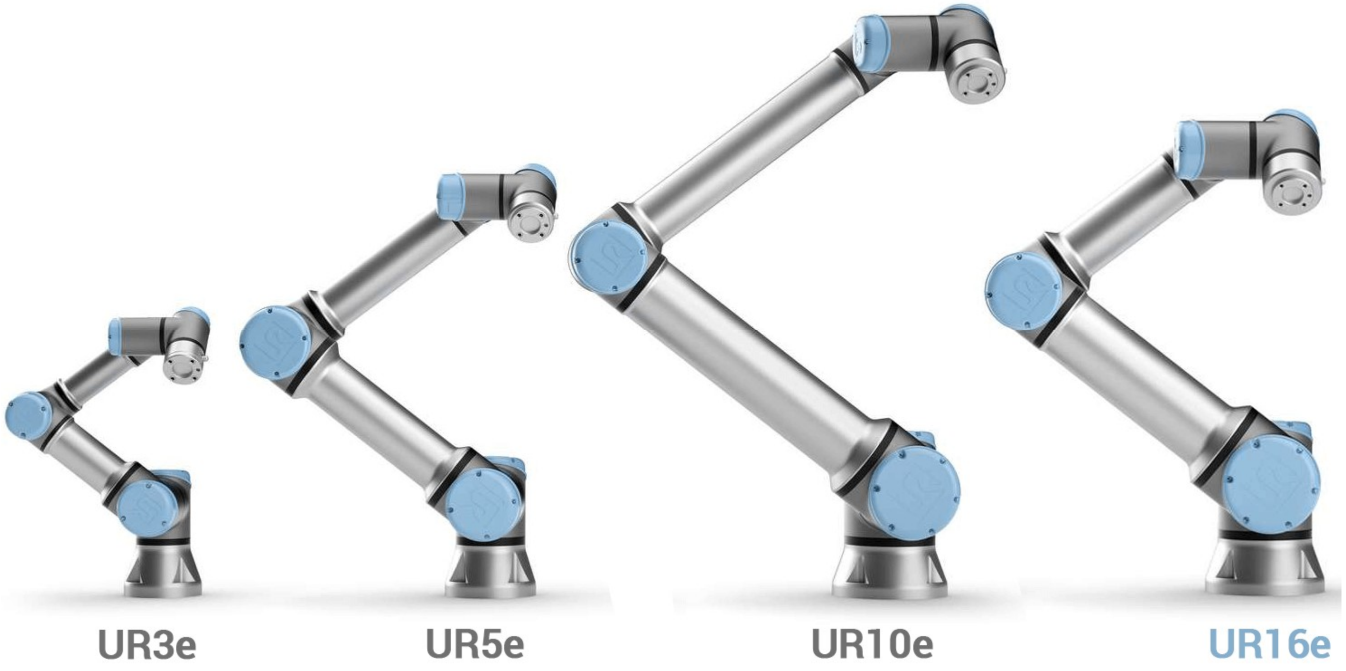


Figure 3. Universal Robot URe-Series of Cobots

www.universal-robots.com

Table 1: Universal Robot Ure-series Data

model	UR3e	UR5e	UR10e	UR16e
reach (mm)	500	850	1300	900
payload (kg)	3	5	10	16
fooprint (mm - dia)	128	149	190	190
mass (kg)	11.2	20.6	33.5	33.1
payload-to-mass ratio	0.27	0.24	0.30	0.48
power consumption (Watts)	100	200	350	350

We see the payload-to-mass ratio in the URe-Series is like average serial robot arms, i.e. poor. But the **UR16e** model is respectable at 0.48.

The required power supply is 100 – 240 VAC at 50 – 60 Hz. The I/O power supply is 24 V and 2 A in the control box, and 12/24 V and 600 mA in the tool. Universal Robot URe-Series cobots are built to handle a temperature range of 0 – 50 $^\circ C$ and a non-condensing relative humidity of 90%. The robots may be mounted in any orientation and meet ISO clear room categories of 5. The noise is stated to be less than 65 dB. The materials used in these cobots are Aluminum, plastics, and steel. The typical endpoint speed is 1 m/sec. On-line programming is accomplished through a teach pendant / GUI with a 12" touch screen.

The Universal Robots naming convention for the six revolute (**R**) joints, base-to-tool-plate, is: Base, Shoulder, Elbow, Wrist 1, Wrist 2, Wrist 3 (see Figure 4). Table 2 presents the joint numbers, name, motion type, angle, and joint limits.

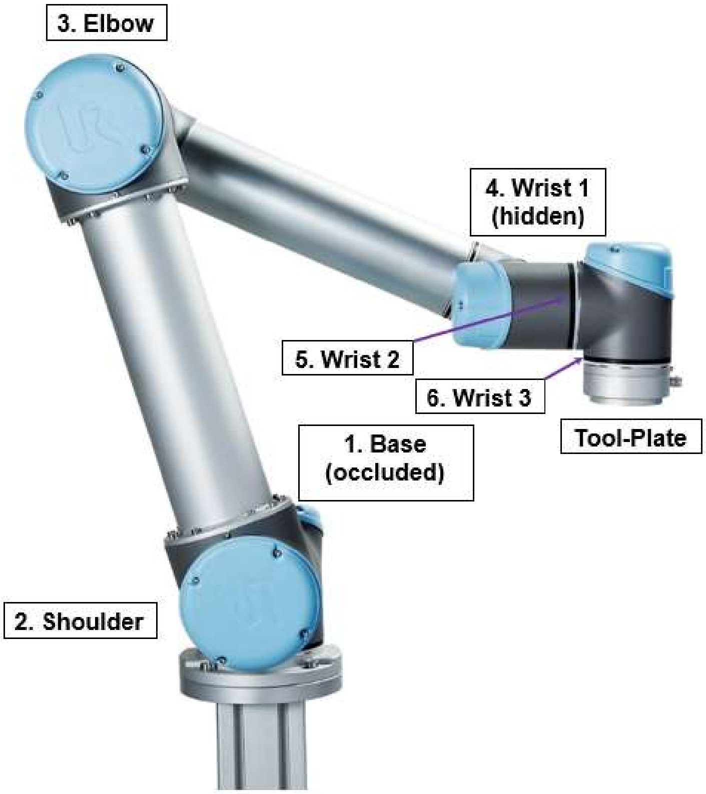


Figure 4. URe-Series Joint-Naming Convention

Table 2. Six-dof URe-Series Arm R Joints Naming Convention

Joint Number	Joint Name	Joint Motion	Joint Angle	Joint Limits
1	Base	waist yaw	θ_1	$\pm 360^\circ$
2	Shoulder	shoulder pitch	θ_2	$\pm 360^\circ$
3	Elbow	elbow pitch	θ_3	$\pm 360^\circ$
4	Wrist 1	wrist pitch	θ_4	$\pm 360^\circ$
5	Wrist 2	wrist yaw	θ_5	$\pm 360^\circ$
6	Wrist 3	wrist roll	θ_6	$\pm 360^{\circ *}$

*unlimited and bidirectional for the UR3e

Table 3 shows the maximum joint speeds for the URe-Series cobots, in units of rad/sec.

Table 3. Six-dof URe-Series Joints Maximum Speed (rad/sec)

Joint Number	UR3e	UR5e	UR10e
1	$\pm\pi$	$\pm\pi$	$\pm 2\pi/3$
2	$\pm\pi$	$\pm\pi$	$\pm 2\pi/3$
3	$\pm\pi$	$\pm\pi$	$\pm\pi$
4	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
5	$\pm 2\pi$	$\pm\pi$	$\pm\pi$
6	$\pm 2\pi$	$\pm\pi$	$\pm\pi$

As mentioned previously, the joints used throughout the URe-Series of serial cobots are modular (and thus replaceable), consisting of five different sizes as shown in Table 4. Table 5 then shows the types of these five joints used in the various sizes of URe arms.

Table 4. Five Sizes of URe-Series Modular Joints

Joint Size	Joint Torque (Nm)	Torque Constant (Nm/Amp)
Size 0	±12	
Size 1	±28	0.092
Size 2	±56	
Size 3	±150	0.125
Size 4	±330	

Note: the joint torque values shown in Table 4 are the maximum possible torque; due to cobot safety, not all of this torque is available in practice. Further, for gravity-mounted robots (in the vertical plane), the torque is effectively reduced more as the robot must lift its own weight against gravity.

Each joint in the URe-Series has a harmonic gear with ratio $n = 101$.

Table 5. Modular Joints used in the Six-dof URe-Series

Joint Number	Joint Name	UR3e	UR5e	UR10e	UR16e
1	Base	Size 2	Size 3	Size 4	Size 4
2	Shoulder	Size 2	Size 3	Size 4	Size 4
3	Elbow	Size 1	Size 3	Size 3	Size 3
4	Wrist 1	Size 0	Size 1	Size 2	Size 2
5	Wrist 2	Size 0	Size 1	Size 2	Size 2
6	Wrist 3	Size 0	Size 1	Size 2	Size 2

Clearly we see that, for a serial robot arm, the stronger motors must be towards the base, while towards the tool-plate, the motor may have less torque capacity. Table 6 presents the positioning repeatability as reported by Universal Robots. The **Ure-series** made a significant improvement in positioning repeatability.

Table 6. URe-Series Universal-Robot-Reported Positioning Repeatability

cobot	UR3	UR5	UR10
position repeatability (mm)	±0.1	±0.1	±0.1
cobot	**UR3e**	**UR5e**	**UR10e**
position repeatability (mm)	±0.03	±0.03	±0.05

Collaboration

Safety

Figure 5. Additional Human / Cobot Photographs

www.universal-robots.com

3. URe-Series Cobot Denavit-Hartenberg (DH) Parameters

The modified Denavit-Hartenberg (DH, Denavit and Hartenberg, 1955) parameters are presented in this section for the serial chain of the Universal Robots 6-dof URe-Series cobots. The Denavit-Hartenberg (DH) Parameters (1955) are used to describe the links/joints geometry of a serial-chain robot. DH parameters have been adopted for standard kinematics analysis in serial-chain robots (Craig, 2005). The community has come to call Craig-style DH Parameters as 'modified', with the original DH Parameters interpretation by Paul as 'standard'. The modified DH parameters have certain advantages over the standard (the main one being that a Craig coordinate frame rotates right at it joint, rather than distal from the joint as in Paul).

The Cartesian reference frame definitions for the URe-Series 6-dof serial arm are shown in Figure 6. Table 7 gives the associated DH parameters (Craig convention, known as 'modified DH parameters').

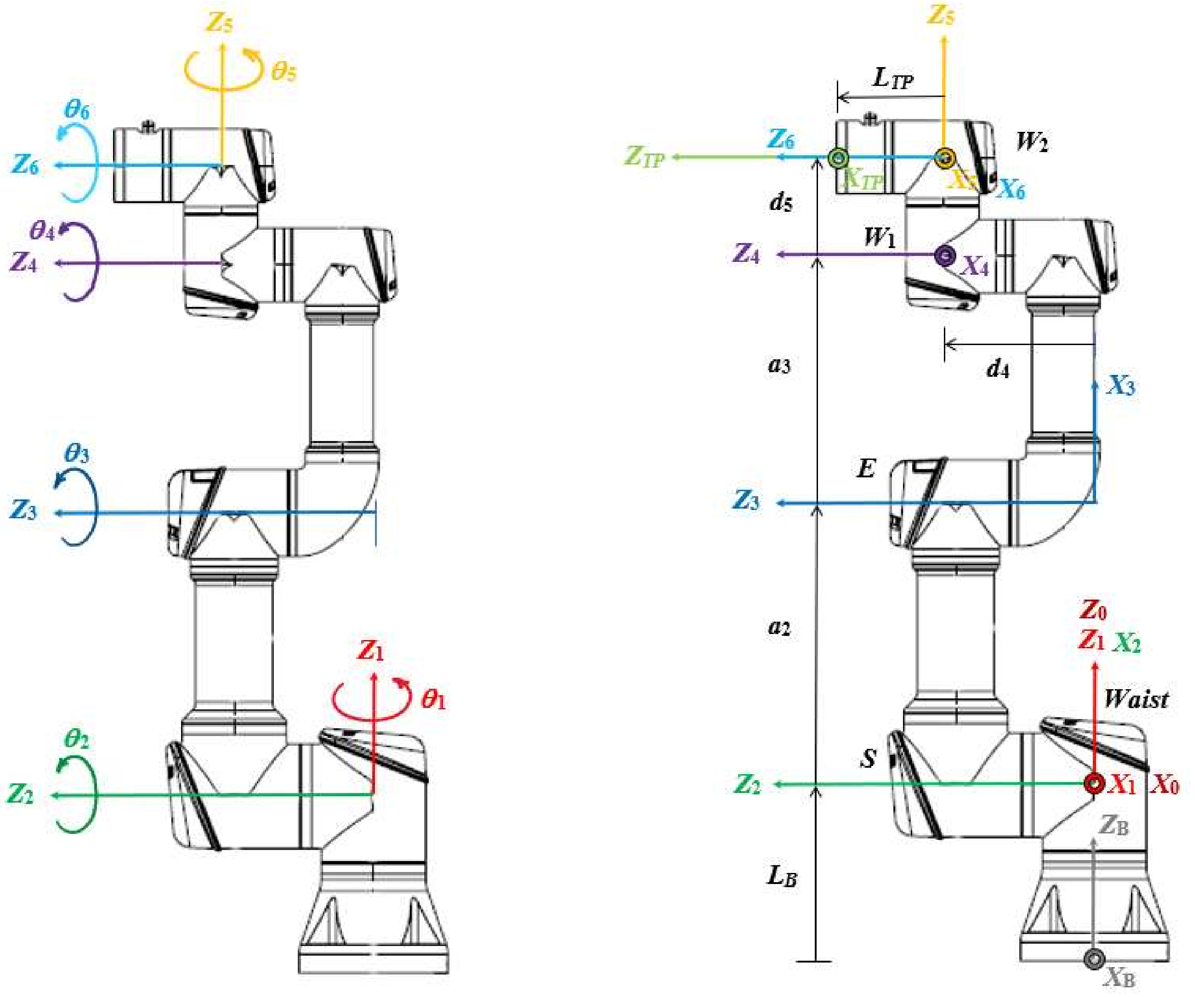


Figure 6. Six-dof Universal URe-Series Kinematic Diagram with Coordinate Frames

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

Kinematic Notation

See the right diagram in Figure 5 above; relative to Universal Robot's notation, this document prefers to refer to joint 1 as the Waist (instead of Base), since the Base commonly refers to a fixed Cartesian coordinate frame $\{B\}$ rather than a joint. The Shoulder joint notation (S), and the Elbow joint notation (E) is straight-forward. Point W_1 above is the origin for the Wrist 1 Cartesian coordinate frame $\{4\}$, and point W_2 above is the origin shared by the Cartesian coordinate frames $\{5\}$ and $\{6\}$ for the Wrist 2 and Wrist 3 joints.

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

4. URe-Series Cobot Forward Pose Kinematics

In general, the Forward Pose Kinematics (FPK) problem for a serial-chain robot is stated: Given the joint values, calculate the pose (position and orientation) of the end-effector frame of interest. For serial-chain robots, the FPK problem set up and solution is straight-forward. It is based on substituting each line of the Denavit-Hartenberg Parameters Table (Table 7) into the equation below (Craig, 2005), giving the pose of frame $\{i\}$ with respect to its nearest neighbor frame $\{i–1\}$ back along the serial chain:

$$\left[{}^{i-1}_{i}T\right]=\begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} \left[{}^{i-1}_{i}R\right] & & & \left\{{}^{i-1}P_i\right\} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used: $c\theta_i = \cos\theta_i$, $s\theta_i = \sin\theta_i$, $c\alpha_i = \cos\alpha_i$, and $s\alpha_i = \sin\alpha_i$.

The equation above represents pose (position and orientation) of frame $\{i\}$ with respect to frame $\{i–1\}$ by using a 4x4 homogeneous transformation matrix. The upper left 3x3 matrix is the rotation matrix $\left[{}^{i-1}_{i}R\right]$ giving the orientation of frame $\{i\}$ with respect to frame $\{i–1\}$, expressed in $\{ i–1\}$ coordinates. The upper right 3x1 vector $\left\{{}^{i-1}P_i\right\}$ is the position vector from the origin of $\{i–1\}$ to the origin of $\{i\}$, expressed in $\{ i–1\}$ coordinates.

Then homogeneous transformation equations are used to find the pose of the overall end-effector frame of interest with respect to the base reference frame, to complete the FPK solution for each serial chain.

4.1 Ure-series Analytical Six-dof FPK Expressions

The statement of the FPK problem for the 6-dof serial chain of the Universal Cobot URe-Series is:

$$\text{Given } (\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\text{, calculate } \left[{}^{0}_{6}T\right] \text{ and } \left[{}^{B}_{TP}T\right].$$

where $(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)$ are the six joint angles,$\{TP\}$ is the tool-plate frame and $\{B\}$ is the fixed robot base reference frame (see Figure 6). Dextral Cartesian coordinate frames are indicated by the curly brackets { }. There are also seven numbered Cartesian coordinate frames {0}, {1}, … {6}. {0} is a fixed frame, while {1} through {6} are the active moving joint frames.

Substitute each row of the DH parameters from Table 7 into the equation for $\left[{}^{i-1}_{i}T\right]$ to obtain the six neighboring homogeneous transformation matrices as a function of the joint angles for the six-dof arm.

$$\left[{}^{0}_{1}T\right]=\begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{1}_{2}T\right]=\begin{bmatrix} -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{2}_{3}T\right]=\begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{3}_{4}T\right]=\begin{bmatrix} s_4 & c_4 & 0 & a_3 \\ -c_4 & s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{4}_{5}T\right]=\begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{5}_{6}T\right]=\begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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.

$$\left[{}^{0}_{6}T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\left[{}^{0}_{1}T(\theta_1)\right]\left[{}^{1}_{2}T(\theta_2)\right]\left[{}^{2}_{3}T(\theta_3)\right]\left[{}^{3}_{4}T(\theta_4)\right]\left[{}^{4}_{5}T(\theta_5)\right]\left[{}^{5}_{6}T(\theta_6)\right]$$

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

$$\left[{}^{0}_{6}T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\left[{}^{0}_{1}T(\theta_1)\right]\left[{}^{1}_{4}T(\theta_2,\theta_3,\theta_4)\right]\left[{}^{4}_{6}T(\theta_5,\theta_6)\right]$$

where $\left[{}^{0}_{1}T(\theta_1)\right]$ was given above, and the other two matrices are found by matrix multiplication and simplification:

$$\left[{}^{1}_{4}T(\theta_2,\theta_3,\theta_4)\right]=\begin{bmatrix} c_{234} & -s_{234} & 0 & -a_2s_2-a_3s_{23} \\ 0 & 0 & -1 & -d_4 \\ s_{234} & c_{234} & 0 & a_2c_2+a_3c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{4}_{6}T(\theta_5,\theta_6)\right]=\begin{bmatrix} c_5c_6 & -c_5s_6 & s_5 & 0 \\ s_6 & c_6 & 0 & d_5 \\ -s_5c_6 & s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the following abbreviations were used:

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

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

This particular grouping in matrix multiplication was used to significantly simplify the matrix $\left[{}^{1}_{4}T(\theta_2,\theta_3,\theta_4)\right]$ using the standard sum of angles formulae (in two levels of simplification):

$$\begin{aligned}\cos(a\pm b) &= cacb \mp sasb \\ \sin(a\pm b) &= sacb \pm casb\end{aligned}$$

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 $\left[{}^{0}_{6}T\right]$ by more matrix multiplications (but no more trigonometric simplifications):

$$\left[{}^{0}_{6}T(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)\right]=\begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^{0}x_6 \\ r_{21} & r_{22} & r_{23} & {}^{0}y_6 \\ r_{31} & r_{32} & r_{33} & {}^{0}z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The orthonormal rotation matrix elements for this FPK result are:

$$\begin{aligned} r_{11} &= -s_1s_5c_6 + c_1(-s_{234}s_6 + c_{234}c_5c_6) \\ r_{21} &= c_1s_5c_6 + s_1(-s_{234}s_6 + c_{234}c_5c_6) \\ r_{31} &= \qquad\qquad\quad c_{234}s_6 + s_{234}c_5c_6 \end{aligned}$$

$$\begin{aligned} r_{12} &= s_1s_5s_6 - c_1(s_{234}c_6 + c_{234}c_5s_6) \\ r_{22} &= -c_1s_5s_6 - s_1(s_{234}c_6 + c_{234}c_5s_6) \\ r_{32} &= \qquad\qquad\quad c_{234}c_6 - s_{234}c_5s_6 \end{aligned}$$

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

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:

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

Note that, since the origins of Cartesian coordinate frames {5} and {6} are coincident at the wrist point, the translational terms above are only functions of the first four joint angles $(\theta_1,\theta_2,\theta_3,\theta_4)$; this will become an issue when we consider Inverse Pose Kinematics (IPK):

$$\left\{ {}^{0}P_6 \right\} = \left\{ {}^{0}P_6(\theta_1, \theta_2, \theta_3, \theta_4) \right\} = \begin{Bmatrix} {}^{0}x_6 \\ {}^{0}y_6 \\ {}^{0}z_6 \end{Bmatrix}$$

Additional, Fixed Transforms – Tool-Plate and Base Coordinate Frames

To complete the FPK solution we need to include the Cartesian coordinate frames $\{TP\}$ and $\{B\}$ to find $\left[{}^{B}_{TP}T \right]$:

$$\left[{}^{B}_{TP}T \right] = \left[{}^{B}_{0}T(L_B) \right] \left[{}^{0}_{6}T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \right] \left[{}^{6}_{TP}T(L_{TP}) \right]$$

Where L_B and L_{TP} are known constants. Note that these two additional matrices are not evaluated by any row in the DH parameter table (those were all used above), since there is no variable associated with these fixed homogeneous transformation matrices based on constant lengths. Instead, they are determined by inspection, using the rotation matrix and position vector components of the homogeneous transformation matrix definition.

$$\left[{}^{B}_{0}T \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{6}_{TP}T \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{TP} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note the origin of $\{TP\}$ is now a function of $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ but it is still not a function of θ_6 since the last Z_6 R-joint axis passes through the origin of $\{6\}$. Of course, the position of a general tool held in a gripper attached to the tool-plate would be a function of all six joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$.

The FPK solutions can be evaluated numerically or symbolically, or using a combination of these two methods.

4.2 Ure-series FPK Examples

Now we present three snapshot FPK examples for the Universal Robot UR3e, for all zero joint angles, a general case, and for the initial angles case. Each shows four MATLAB views of the Cobot pose. Then a FPK trajectory example is given. Translational units are m in all examples.

FPK Example 1: Zero Joint Angles

Given Zero Joint Angles: $\{\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6\} = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\}$

The FPK results are:

$$\left[{}_6^0T\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.131 \\ 0 & 1 & 0 & 0.542 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}_{TP}^{B}T\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.223 \\ 0 & 1 & 0 & 0.694 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

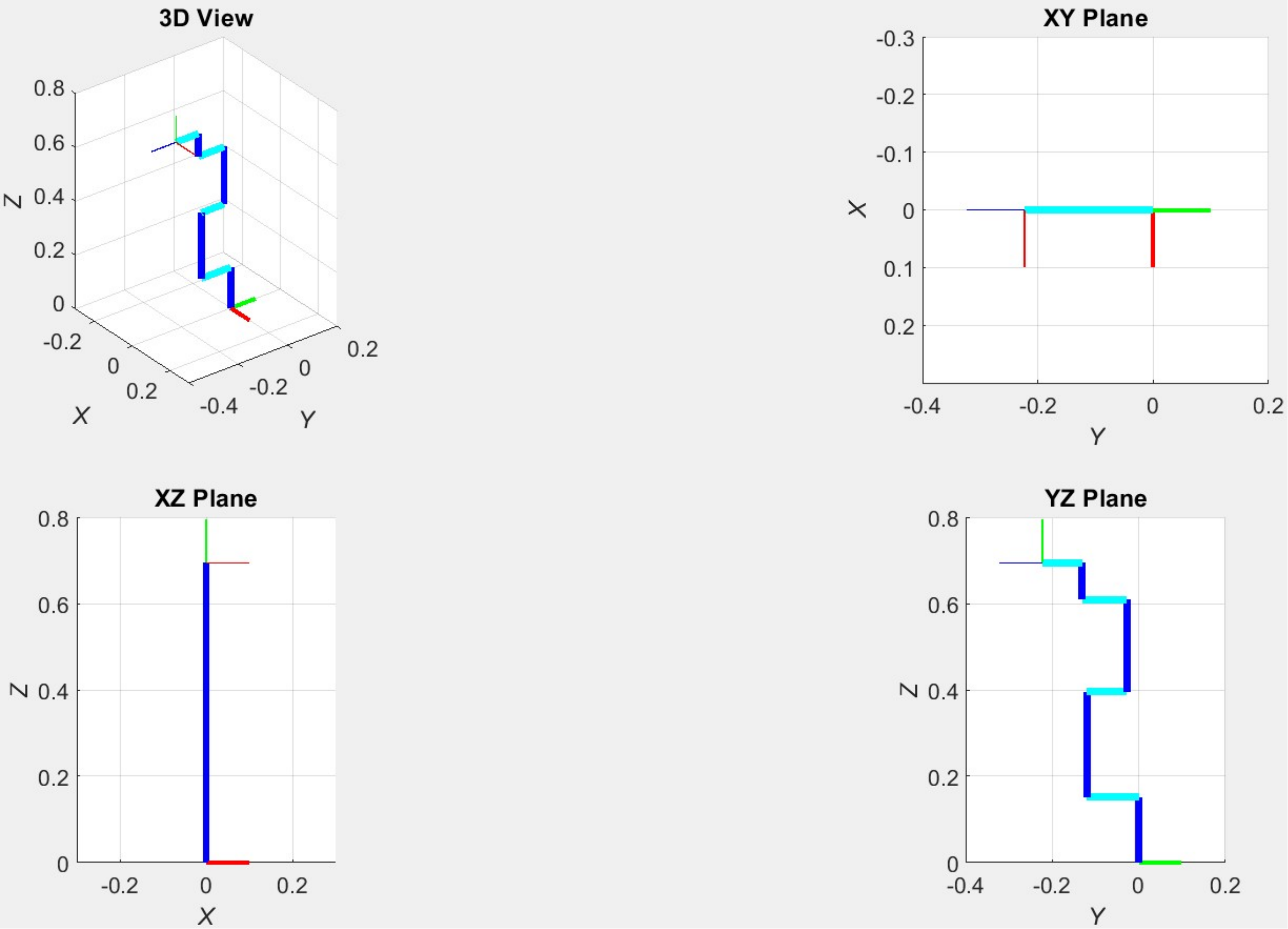


Figure 7. FPK Example 1, Zero Joint Angles

FPK Example 2: General Joint Angles

Given General Joint Angles: $\{\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6\} = \{20^\circ \quad 40^\circ \quad 60^\circ \quad 50^\circ \quad 70^\circ \quad 10^\circ\}$

The FPK results are:

$$\left[{}^{0}_{6}T\right] = \begin{bmatrix} -0.6722 & -0.3586 & -0.6477 & -0.3396 \\ 0.7401 & -0.3042 & -0.5997 & -0.2630 \\ 0.0180 & -0.8826 & 0.4698 & 0.0763 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{B}_{TP}T\right] = \begin{bmatrix} -0.6722 & -0.3586 & -0.6477 & -0.3992 \\ 0.7401 & -0.3042 & -0.5997 & -0.3182 \\ 0.0180 & -0.8826 & 0.4698 & 0.2715 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

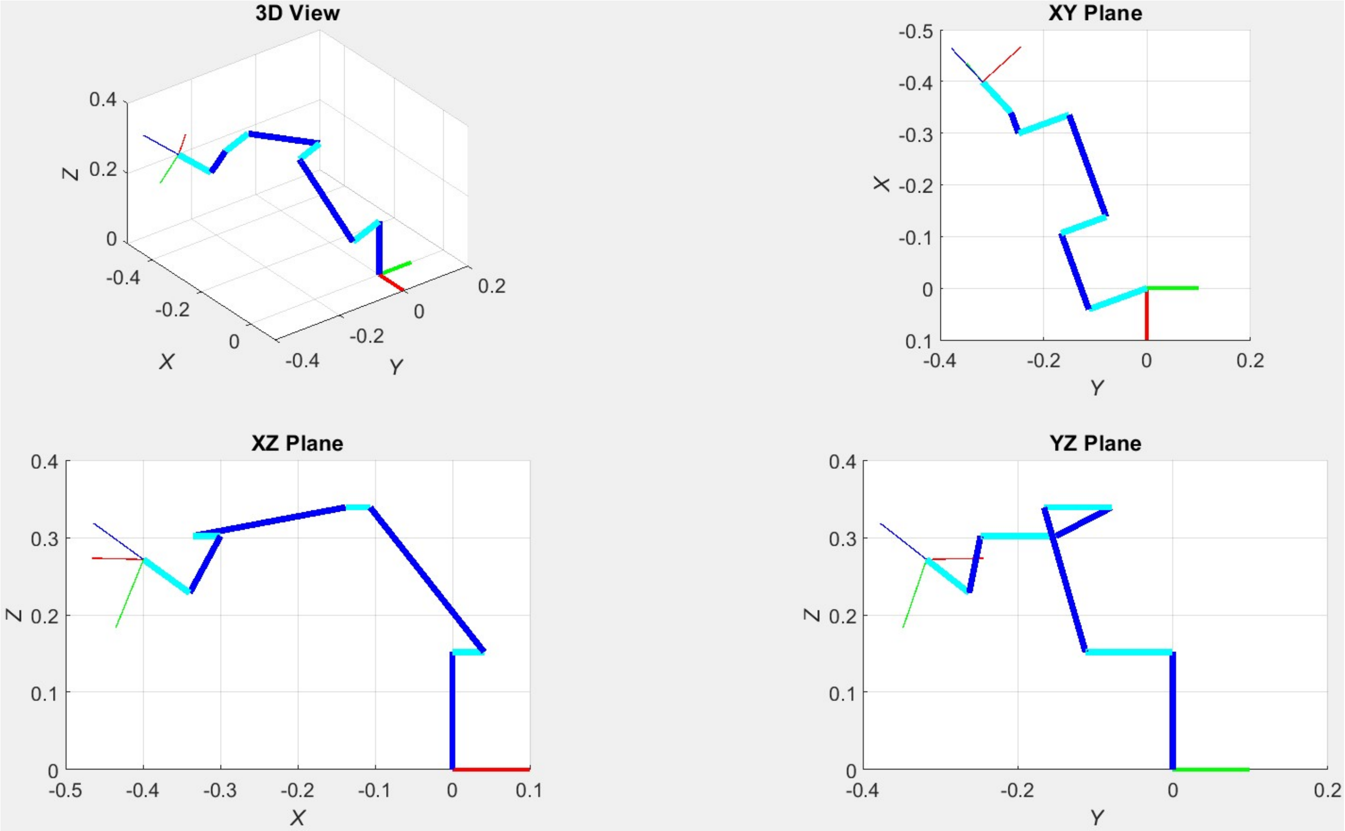


Figure 8. FPK Example 2, General Joint Angles

The cobot arm pose shown in Figures 6 and 7 are for all zero joint angles. The six joint angles for a suggested initial pose are given in Table 9. A third snapshot FPK solution and MATLAB graphic for this initial pose is then presented.

Table 9. Universal URe Initial Joint Angles

Joint Number	Joint Name	Joint Variable	Initial Angle (deg)
1	Base	θ_1	0
2	Shoulder	θ_2	45
3	Elbow	θ_3	90
4	Wrist 1	θ_4	45
5	Wrist 2	θ_5	90
6	Wrist 3	θ_6	0

FPK Example 3: Initial Joint Angles

Given Initial Joint Angles: $\{\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6\} = \{0 \quad 45^\circ \quad 90^\circ \quad 45^\circ \quad 90^\circ \quad 0\}$

The FPK results are:

$$\left[{}^{0}_{6}T\right] = \begin{bmatrix} 0 & 0 & -1 & -0.3231 \\ 1 & 0 & 0 & -0.1310 \\ 0 & -1 & 0 & -0.0631 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{B}_{TP}T\right] = \begin{bmatrix} 0 & 0 & -1 & -0.4151 \\ 1 & 0 & 0 & -0.1310 \\ 0 & -1 & 0 & 0.0889 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

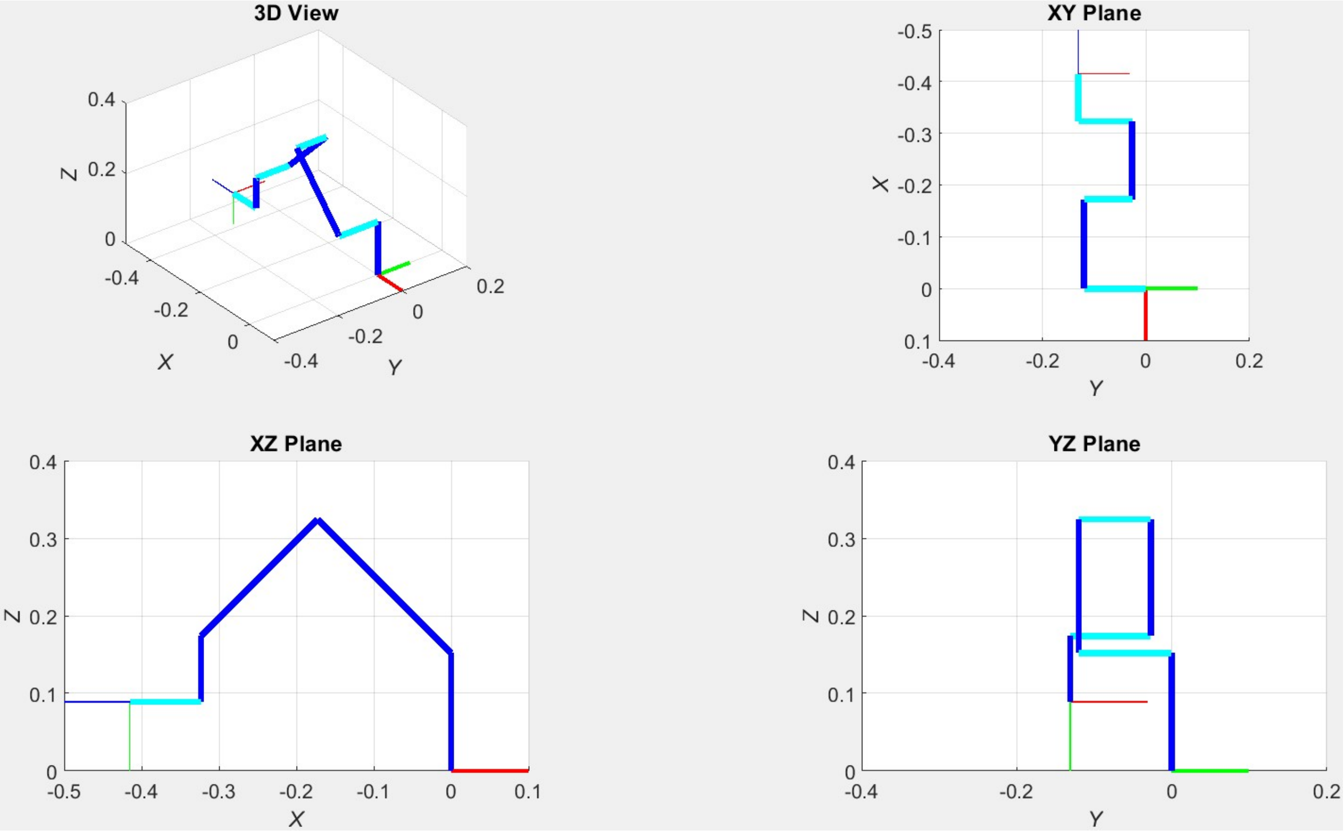


Figure 9. FPK Example 3, initial joint angles

The following FPK solution validations were performed successfully in all 3 snapshot examples:

- The numerical and symbolic solutions for $\left[{}^{0}_{6}T\right]$ and $\left[{}^{B}_{TP}T\right]$ agree.
- The FPK solutions were performed by inspection (verifying the $\{TP\}$ Cartesian position and the XYZ_{TP} pointing directions with respect to $\{B\}$) for Examples 1 and 3, which agreed with the above solutions.

All-Joints FPK Motion Simulation

In this MATLAB simulation we changed all six joints simultaneously and linearly, over their entire $\pm 360^\circ$ joint limits, in 100 steps. The FPK results plots repeat over half the joint limit range, so this simulation is for joint limit ranges $(0, +360^\circ)$ on all six joints. Figure 10 shows the six joint angle inputs (all are identical).

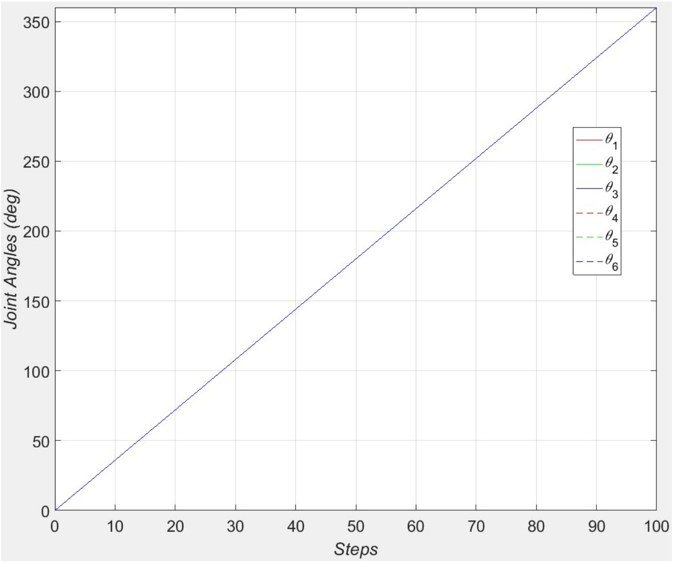
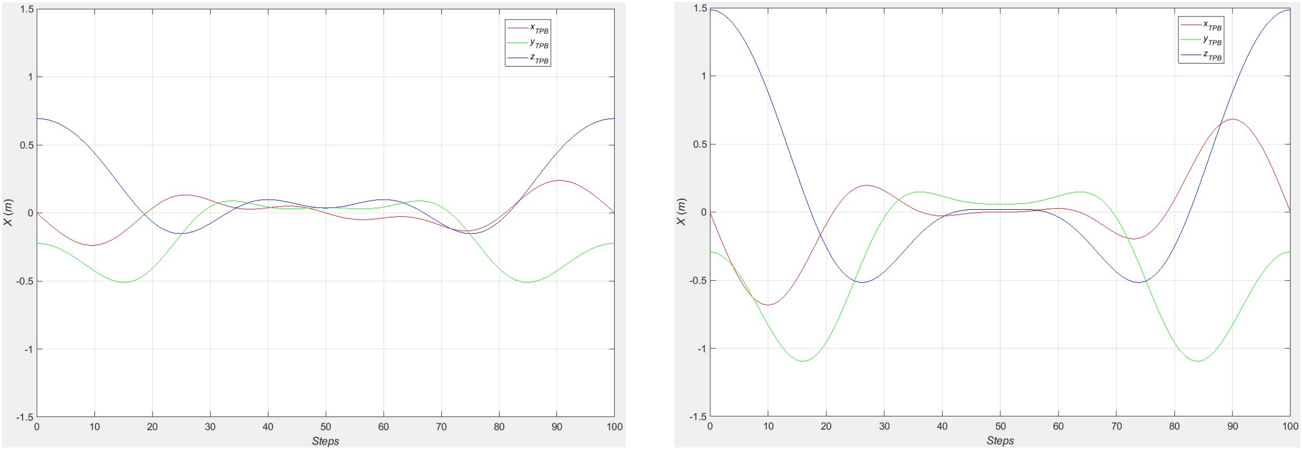


Figure 10. All-Joints FPK Inputs, both UR3e and UR10e

The plots of Figures 11 below show the resulting Cartesian translational portion of the FPK results, for the UR3e and UR10e Cobots. The UR10e is significantly larger than the UR3e, which these results verify. The published reaches are 500 vs. 1300 mm for the UR3e vs. UR10e cobots.



UR3e **UR10e**

Figure 11. All-Joints FPK Translational Results

Since the size differences of these two cobots do not affect the rotational FPK results, the α, β, γ Euler angles plots are identical, shown in Figure 12.

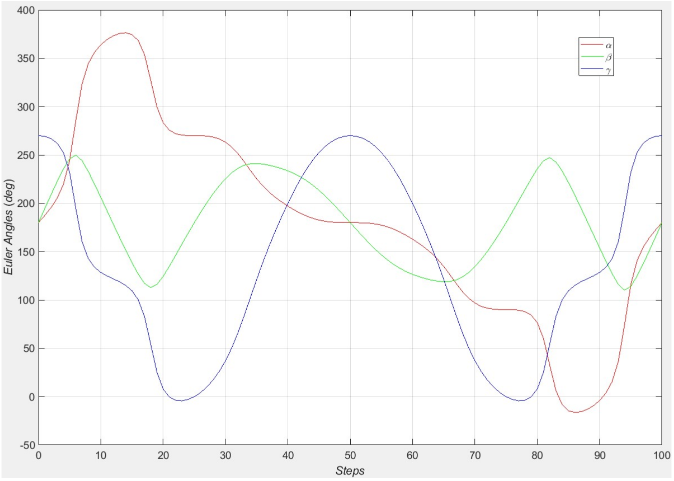
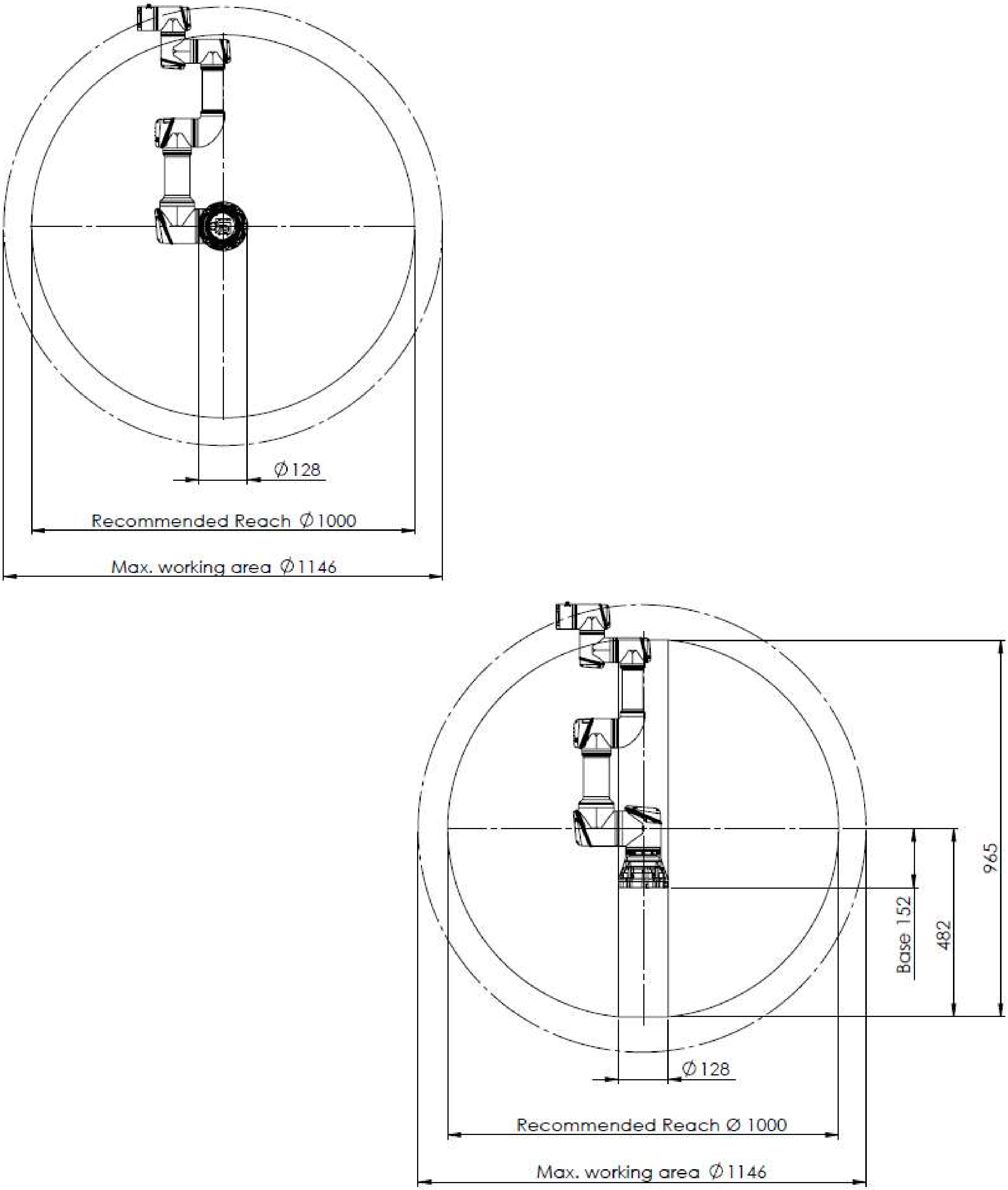


Figure 12. All-Joints FPK Rotational Results, UR3e and UR10e (identical)

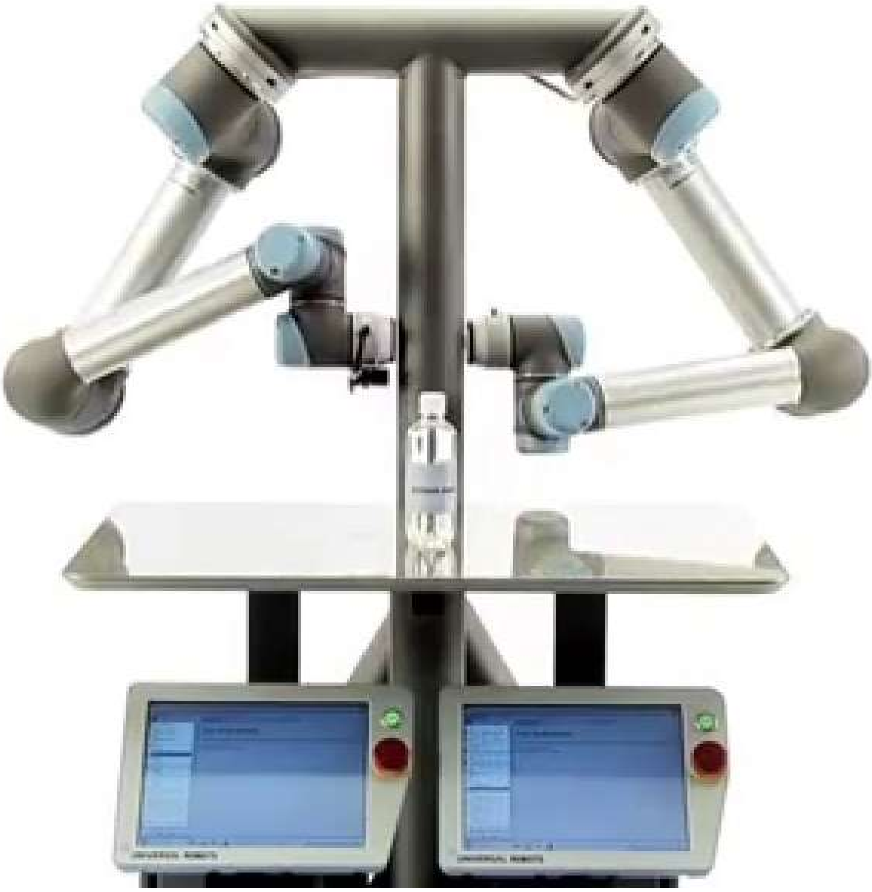
5. URe-Series Cobot Workspace



www.universal-robots.com

Figure 13. Universal UR3e 6-dof Cobot Reachable Workspace, top view and front view

Considering its size, this robot has a large reachable workspace, even for a serial robot, since all six R joints have generous ±360° joint limits. As the figures show, this is a full spherical reachable workspace! Of course, mounted on a table, this would reduce to a hemisphere. With its base suspended from a pedestal, as shown in the figure below, the spherical workspace is approached. The same would be achieved for a single arm mounted on a floor-based pedestal (not shown).



www.universal-robots.com

Figure 13. Dual Universal UR3e 6-dof Cobots mounted to a pedestal

During our FPK motion simulations, it was discovered that some joints' motion for the published ±360° joint limits lead to physical interference between links. This is obviously an essential issue which limits the effective reachable workspace, especially interior to the claimed spherical workspace.

6. URe-Series Cobot Inverse Pose Kinematics (IPK)

According to Pieper's principle, if a 6-dof serial robot has 3 consecutive coordinate frames meeting at the same origin, then an analytical solution is guaranteed to exist for the coupled nonlinear inverse pose kinematics problem. This does not occur for the URe Cobots. There is no spherical wrist where three consecutive wrist frame origins share the same point: instead, the offset d_5 disrupts this desired attribute. But Pieper's principle guarantees an analytical solution if his condition is met; it doesn't say that there is no analytical solution in the absence of this condition. So let us pursue an analytical solution for the Inverse Pose Kinematics (IPK) problem of the URe Cobots, despite the lack of a spherical wrist.

In general, the Inverse Pose Kinematics (IPK) problem for a serial-chain robot is stated: Given the pose (position and orientation) of the end frame of interest, calculate the joint values to obtain that pose. For serial-chain robots, the IPK solution starts with the FPK equations. The solution of coupled nonlinear algebraic equations is required and multiple solution sets generally result.

6.1 Ure-series Analytical Six-dof IPK Solution

The specific statement of the IPK problem for the 6-dof URe serial cobots (the entire series shares the identical kinematic design shown in Figure 6) is:

Given: the constant DH Parameters

and the required end-effector pose $\left[{}^0_6T\right]=\begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^0x_6 \\ r_{21} & r_{22} & r_{23} & {}^0y_6 \\ r_{31} & r_{32} & r_{33} & {}^0z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Calculate: the joint angles $(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6)$ to achieve this pose

Actually, in the real world, a more general pose input $\left[{}^B_{TP}T\right]$ must be given. Then the associated required IPK input $\left[{}^0_6T\right]$ is calculated from known constant homogeneous transformation matrices as follows:

$$\left[{}^B_{TP}T\right]=\left[{}^B_0T\right]\left[{}^0_6T\right]\left[{}^6_{TP}T\right]$$

$$\left[{}^0_6T\right]=\left[{}^B_0T\right]^{-1}\left[{}^B_{TP}T\right]\left[{}^6_{TP}T\right]^{-1}$$

where the constant homogeneous transformation matrices $\left[{}^B_0T\right],\left[{}^6_{TP}T\right]$ were given in Section 4, Forward Pose Kinematics (FPK). Remember these two matrices do not come from the DH Parameters table, but were found by inspection. The IPK equations come from the FPK expressions; but with the six joint angles now unknown, coupled nonlinear (transcendental) equations result, very difficult to solve compared to FPK.

Here are the form of the FPK equations; remember the LHS is a (consistent) given set of 16 numbers representing the desired pose of {6} with respect to {0}:

$$\left[{}^{0}_{6}T\right]=\left[{}^{0}_{1}T(\theta_1)\right]\left[{}^{1}_{2}T(\theta_2)\right]\left[{}^{2}_{3}T(\theta_3)\right]\left[{}^{3}_{4}T(\theta_4)\right]\left[{}^{4}_{5}T(\theta_5)\right]\left[{}^{5}_{6}T(\theta_6)\right]$$

So in principle 16 equations may be written to solve the IPK problem. But 4 of these are useless (the last row [0 0 0 1]). All three translational equations are useful and independent. The remaining nine equations come from the rotation matrix terms, only three of which are independent.

A classic IPK solution approach is to invert some of the consecutive $\left[{}^{i}_{i+1}T(\theta_{i+1})\right]$ homogeneous transformation matrices from the RHS and multiply them to the given numbers $\left[{}^{0}_{6}T\right]$ on the appropriate sides of the LHS. For the URe Cobots, use the following homogeneous transformation equation to separate two unknowns θ_1,θ_6 from the other four unknown joint angles $\theta_2,\theta_3,\theta_4,\theta_5$:

$$\left[{}^{0}_{1}T(\theta_1)\right]^{-1}\left[{}^{0}_{6}T\right]\left[{}^{5}_{6}T(\theta_6)\right]^{-1}=\left[{}^{1}_{2}T(\theta_2)\right]\left[{}^{2}_{3}T(\theta_3)\right]\left[{}^{3}_{4}T(\theta_4)\right]\left[{}^{4}_{5}T(\theta_5)\right]$$

Now we must inspect the resulting equations in order to find equations that may be solved for all 6 unknown joint angles in some do-able order.

$$\left[{}^{0}_{1}T(\theta_1)\right]^{-1}\left[{}^{0}_{6}T\right]\left[{}^{5}_{6}T(\theta_6)\right]^{-1}=$$

$$\begin{bmatrix} (r_{11}c_1+r_{21}s_1)c_6-(r_{12}c_1+r_{22}s_1)s_6 & -r_{13}c_1-r_{23}s_1 & (r_{12}c_1+r_{22}s_1)c_6+(r_{11}c_1+r_{21}s_1)s_6 & {}^{0}x_6c_1+{}^{0}y_6s_1 \\ (r_{21}c_1-r_{11}s_1)c_6-(r_{22}c_1-r_{12}s_1)s_6 & -r_{23}c_1+r_{13}s_1 & (r_{22}c_1-r_{12}s_1)c_6+(r_{21}c_1-r_{11}s_1)s_6 & {}^{0}y_6c_1-{}^{0}x_6s_1 \\ r_{31}c_6-r_{32}s_6 & -r_{33} & r_{32}c_6+r_{31}s_6 & {}^{0}z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{1}_{2}T(\theta_2)\right]\left[{}^{2}_{3}T(\theta_3)\right]\left[{}^{3}_{4}T(\theta_4)\right]\left[{}^{4}_{5}T(\theta_5)\right]=\begin{bmatrix} c_{234}c_5 & -c_{234}s_5 & -s_{234} & -a_2s_2-a_3s_{23}-d_5s_{234} \\ s_5 & c_5 & 0 & -d_4 \\ s_{234}c_5 & -s_{234}s_5 & c_{234} & a_2c_2+a_3c_{23}+d_5c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

First, the (2,4) (y-translational) terms involve only one unknown, θ_1:

$${}^{0}y_6c_1-{}^{0}x_6s_1=-d_4$$

Re-writing into classical form yields a well-known equation:

$$E_1\cos\theta_1+F_1\sin\theta_1+G_1=0$$

where:

$$E_1 = \ {}^0y_6$$
$$F_1 = -{}^0x_6$$
$$G_1 = \ d_4$$

This equation may be solved for the unknown θ_1 by applying the well-known and perennial-favourite tangent half-angle substitution.

If we define $t = \tan\left(\frac{\theta_1}{2}\right)$ then $\cos\theta_1 = \frac{1-t^2}{1+t^2}$ and $\sin\theta_1 = \frac{2t}{1+t^2}$

Substitute this **Tangent Half-Angle Substitution** into the *EFG* equation:

$$E_1\left(\frac{1-t^2}{1+t^2}\right) + F_1\left(\frac{2t}{1+t^2}\right) + G_1 = 0$$

$$E_1(1-t^2) + F_1(2t) + G_1(1+t^2) = 0$$

$$(G_1 - E_1)t^2 + (2F_1)t + (G_1 + E_1) = 0$$

So we see this converts the original first-order trigonometric equation into a quadratic polynomial. Using the quadratic formula, we can solve for the intermediate parameter *t*:

$$t_{1,2} = \frac{-F_1 \pm \sqrt{E_1^2 + F_1^2 - G_1^2}}{G_1 - E_1}$$

Then solve for θ_1 by inverting the original Tangent Half-Angle Substitution definition:

$$\theta_{1_{1,2}} = 2\tan^{-1}(t_{1,2})$$

Note that we do not need to use the quadrant-specific **atan2** function in the above solution, since the multiplier 2 takes care of possible the trigonometric uncertainty (dual values) of inverse trigonometric functions. There are two valid solutions for θ_1, from the ± in the quadratic formula.

<u>Second</u>, the (2,3) (rotational) terms now comprise one unknown θ_6, since θ_1 has been found:

$$(r_{22}c_1 - r_{12}s_1)c_6 + (r_{21}c_1 - r_{11}s_1)s_6 = 0$$

The solution for θ_6 does not require the Tangent Half-Angle Substitution since the new *G* is zero.

$$\frac{s_6}{c_6} = \tan\theta_6 = \frac{r_{12}s_1 - r_{22}c_1}{r_{21}c_1 - r_{11}s_1}$$

$$\theta_6 = \text{atan2}(r_{12}s_1 - r_{22}c_1, r_{21}c_1 - r_{11}s_1)$$

Note we must use the quadrant-specific inverse tangent function **`atan2`** for the θ_6 solution above. This yields a unique θ_6 for each of the two θ_1 results.

Third, since θ_1 and θ_6 are now known, a ratio of the (2,1) to (2,2) rotational equations can be used to solve for θ_5:

$$\begin{aligned}(r_{21}c_1 - r_{11}s_1)c_6 - (r_{22}c_1 - r_{12}s_1)s_6 &= s_5 \\ -r_{23}c_1 + r_{13}s_1 &= c_5\end{aligned}$$

$$\theta_5 = \text{atan2}((r_{21}c_1 - r_{11}s_1)c_6 + (r_{12}s_1 - r_{22}c_1)s_6, r_{13}s_1 - r_{23}c_1)$$

Note again we must use the quadrant-specific inverse tangent function **`atan2`** for the θ_5 solution above. This yields a unique θ_5 for each of the two θ_1, θ_6 results.

Now we are halfway home! Before step 4 we need to gather two intermediate equations, from the (3,1) and (3,3) rotational terms:

$$\begin{aligned}r_{31}c_6 - r_{32}s_6 &= s_{234}c_5 \\ r_{32}c_6 + r_{31}s_6 &= c_{234}\end{aligned} \qquad \text{so} \qquad \begin{aligned}s_{234} &= \frac{r_{31}c_6 - r_{32}s_6}{c_5} = A \\ c_{234} &= r_{32}c_6 + r_{31}s_6 = B\end{aligned}$$

Fourth, substitute these intermediate terms c_{234} and s_{234} into the *x* and *z* translational equations (1,4) and (3,4), which have not yet been used. This replaces the sum of three unknowns $(\theta_2 + \theta_3 + \theta_4)$ in these translational equations with two angles θ_5, θ_6 that are now known.

$$\begin{aligned}{}^0x_6c_1 + {}^0y_6s_1 &= -a_2s_2 - a_3s_{23} - d_5s_{234} = -a_2s_2 - a_3s_{23} - d_5A \\ {}^0z_6 &= \quad a_2c_2 + a_3c_{23} + d_5c_{234} = \quad a_2c_2 + a_3c_{23} + d_5B\end{aligned}$$

Rearrange these equations to isolate the $(\theta_2 + \theta_3)$ terms:

$$\begin{aligned}a_3s_{23} &= -a_2s_2 - {}^0x_6c_1 - {}^0y_6s_1 - d_5A \\ a_3c_{23} &= -a_2c_2 + {}^0z_6 - d_5B\end{aligned} \qquad \begin{aligned}a_3s_{23} &= -a_2s_2 + a \\ a_3c_{23} &= -a_2c_2 + b\end{aligned}$$

Where, for convenience, define:

$$\begin{aligned} a &= -{}^0x_6c_1 - {}^0y_6s_1 - d_5A \\ b &= {}^0z_6 - d_5B \end{aligned}$$

Square and add the two equations to eliminate the $(\theta_2 + \theta_3)$ terms; this yields the following, our second *EFG*-type equation:

$$E_2 \cos\theta_2 + F_2 \sin\theta_2 + G_2 = 0$$

where:

$$\begin{aligned} E_2 &= -2a_2b \\ F_2 &= -2a_2a \\ G_2 &= \quad a_2^2 + a^2 + b^2 - a_3^2 \end{aligned}$$

This equation may be solved for the unknown θ_2 by again applying the trusty tangent half-angle substitution, the same method used earlier for θ_1.

$$t_{2_{1,2}} = \frac{-F_2 \pm \sqrt{E_2^2 + F_2^2 - G_2^2}}{G_2 - E_2} \qquad\qquad \theta_{2_{1,2}} = 2\tan^{-1}(t_{2_{1,2}})$$

Again, there are two valid solutions for θ_2, due to the ± in the quadratic formula; there are two θ_2 solutions for each of the two valid θ_1 solutions, for a total of 4 valid θ_1, θ_2 solutions so far.

<u>Fifth</u>, return to the *x* and *z* (1,4) and (3,4) translational equations (repeated below).

$$\begin{aligned} a_3s_{23} &= -a_2s_2 + a \\ a_3c_{23} &= -a_2c_2 + b \end{aligned}$$

Since squaring-and-adding used one independence, we are free to use them again in a different way. Using a ratio of the equations:

$$\theta_{3_{1,2}} = \text{atan2}(a - a_2s_{2_{1,2}}, b - a_2c_{2_{1,2}}) - \theta_{2_{1,2}}$$

Using the quadrant-specific inverse tangent function **`atan2`** for the θ_3 solution above, there is a unique θ_3 for each of the two valid θ_2 solutions (so there is no additional ballooning of number of solution sets here).

<u>Sixth</u>, and last, return to the two intermediate equations above, from the (3,1) and (3,3) rotational terms. Since θ_2, θ_3 are now known (not to mention θ_6 is also known), the solution to θ_4 is now straight-forward:

$$\theta_{4_{1,2}} = \mathrm{atan2}(A, B) - \theta_{2_{1,2}} - \theta_{3_{1,2}}$$

Again, using the quadrant-specific inverse tangent function **`atan2`** for the θ_4 solution above, there is a unique θ_4 for each of the two valid θ_2, θ_3 solutions (so there is no additional ballooning of number of solution sets here).

<u>In Summary</u>, there are 4 overall IPK joint angles solution sets $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ to achieve the commanded pose, as detailed in Table 10.

Table 10. The Four Universal Ure-series IPK Solution Sets

Solution Set	***t*1 / *t*2 sign**	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	**elbow**
1	$+/-$	θ_{1a}	θ_{2a_1}	θ_{3a}	θ_{4a_1}	θ_{5a}	θ_{6a}	up
2	$+/+$	θ_{1a}	θ_{2a_2}	$-\theta_{3a}$	θ_{4a_2}	θ_{5a}	θ_{6a}	down
3	$-/-$	θ_{1b}	θ_{2b_1}	θ_{3b}	θ_{4b_1}	θ_{5b}	θ_{6b}	down
4	$-/+$	θ_{1b}	θ_{2b_2}	$-\theta_{3b}$	θ_{4b_2}	θ_{5b}	θ_{6b}	up

Here are the patterns seen in the four IPK solution sets: 1. For a given θ_1, the elbow joint angle θ_3 for elbow-up is negative of that for elbow-down, as expected. 2. For a given θ_1, the last two wrist joint angles θ_5, θ_6 are the same for both elbow-up and elbow-down, also as expected.

At first eight distinct solution sets were expected, due to the similarity to PUMA-type 6-dof 6R serial robots. However, that is for a spherical wrist, i.e. one in which the last three active (wrist) joint frames share a common origin. The wrist offset d_5 in the URe-series prevents this.

6.2 Ure-series IPK Examples

Now we present three Universal Robot UR3e snapshot analytical IPK examples, the same poses as in the FPK examples (for all zero joint angles, a general case, and for the initial angles case). Unlike the FPK examples, there are multiple IPK solution sets to each case, which are presented analytically, something a numerical IPK solution procedure cannot do. Translational units are m and angle units are in degrees, in all examples.

IPK Example 1: Zero Joint Angles

Given:

$$\left[{}^{B}_{TP}T\right]=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & -1 & -0.223\\ 0 & 1 & 0 & 0.694\\ 0 & 0 & 0 & 1\end{bmatrix}$$

we first use a homogeneous transformation equation $\left[{}^{0}_{6}T\right]=\left[{}^{B}_{0}T\right]^{-1}\left[{}^{B}_{TP}T\right]\left[{}^{6}_{TP}T\right]^{-1}$ with constant matrices $\left[{}^{B}_{0}T\right],\left[{}^{6}_{TP}T\right]$ to find:

$$\left[{}^{0}_{6}T\right]=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & -1 & -0.131\\ 0 & 1 & 0 & 0.542\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The IPK results are all identical, for all four solution sets:

$$\{\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6\}=\{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\}$$

and the robot pose is the same as shown earlier, in FPK Example 1. This is case is singular; however the IPK equations still work, for all four solution sets (identical). Actually, this case is triply singular (all three singularity conditions are met – see the Velocity Kinematics Section).

IPK Example 2: General Joint Angles

Given:

$$\left[{}^{B}_{TP}T\right]=\begin{bmatrix}-0.6722 & -0.3586 & -0.6477 & -0.3992\\ 0.7401 & -0.3042 & -0.5997 & -0.3182\\ 0.0180 & -0.8826 & 0.4698 & 0.2715\\ 0 & 0 & 0 & 1\end{bmatrix}$$

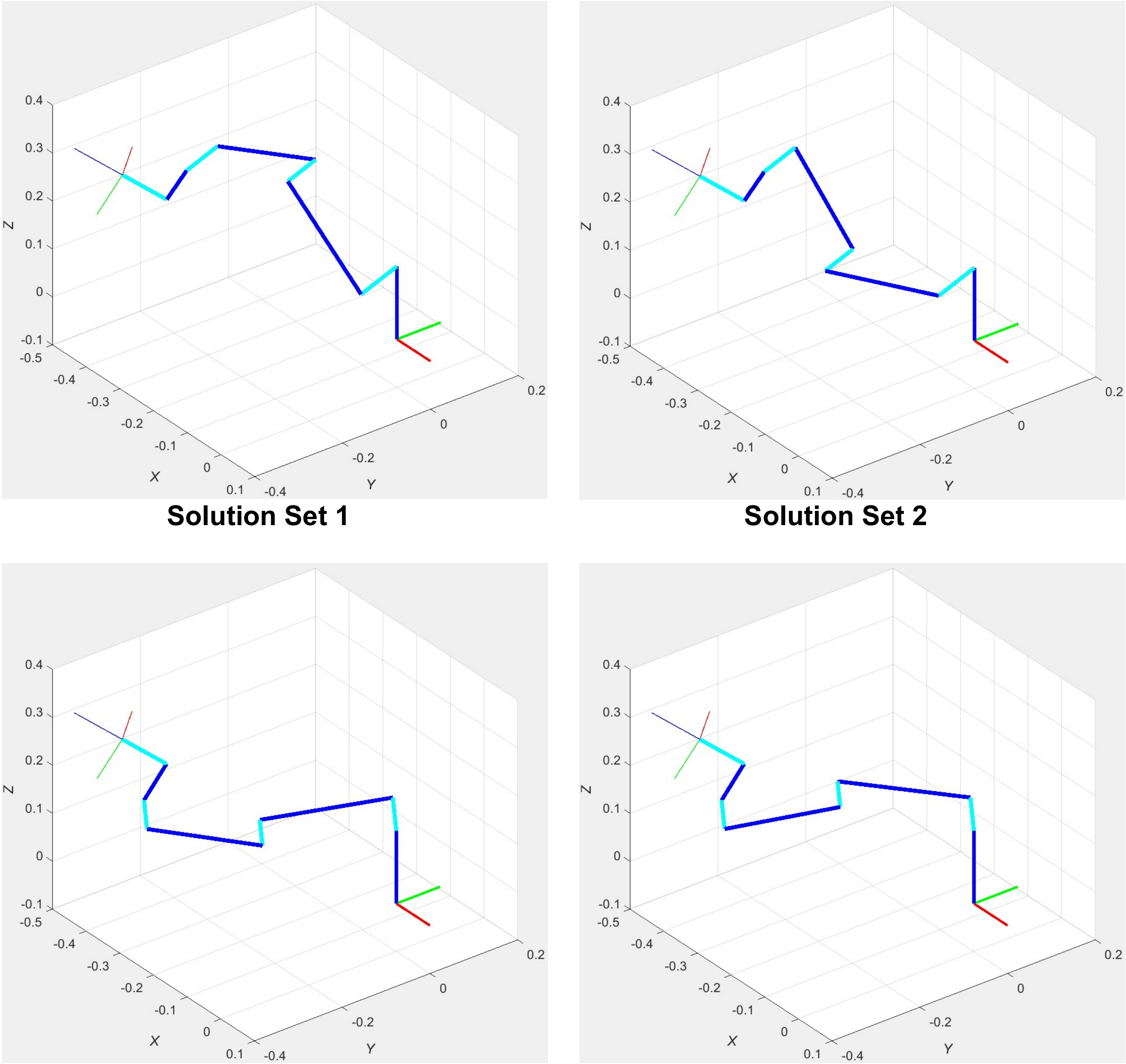
we first use $\left[{}^{0}_{6}T\right]=\left[{}^{B}_{0}T\right]^{-1}\left[{}^{B}_{TP}T\right]\left[{}^{6}_{TP}T\right]^{-1}$ with constant matrices $\left[{}^{B}_{0}T\right],\left[{}^{6}_{TP}T\right]$ to find:

$$\left[{}^{0}_{6}T\right]=\begin{bmatrix}-0.6722 & -0.3586 & -0.6477 & -0.3396\\ 0.7401 & -0.3042 & -0.5997 & -0.2630\\ 0.0180 & -0.8826 & 0.4698 & 0.0763\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The IPK results are:

Solution Set	t_1 / t_2 sign	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	elbow
1	+ / –	20	40	60	50	70	10	up
2	+ / +	20	95.5	–60	114.5	70	10	down
3	– / –	–124.5	–99.3	20.4	107.5	78.8	172.8	down
4	– / +	–124.5	–80.3	–20.4	129.3	78.8	172.8	up

As seen in the figures on the following page, all four solution sets achieve the same correct commanded Cartesian pose. The four solutions are formed by permutations of the $\pm$ in the solutions for joint angles θ_1 and θ_2. The second column in the table above reports the signs used in the quadratic formula of t_1 and t_2, leading to θ_1 and θ_2. The last column in the table above reports the elbow condition of each pose, i.e. elbow-up or elbow-down. These results are also clearly visible in the figures on the following page.



Solution Set 1

Solution Set 2

Solution Set 3

Solution Set 4

IPK Example 2, Four Possible Solution Sets Poses

IPK Example 3: Initial Joint Angles

This example is for the same Initial Joint Angles introduced in FPK Example 3.

Given:

$$\left[{}^{B}_{TP}T\right]=\begin{bmatrix}0 & 0 & -1 & -0.4151\\ 1 & 0 & 0 & -0.1310\\ 0 & -1 & 0 & 0.0889\\ 0 & 0 & 0 & 1\end{bmatrix}$$

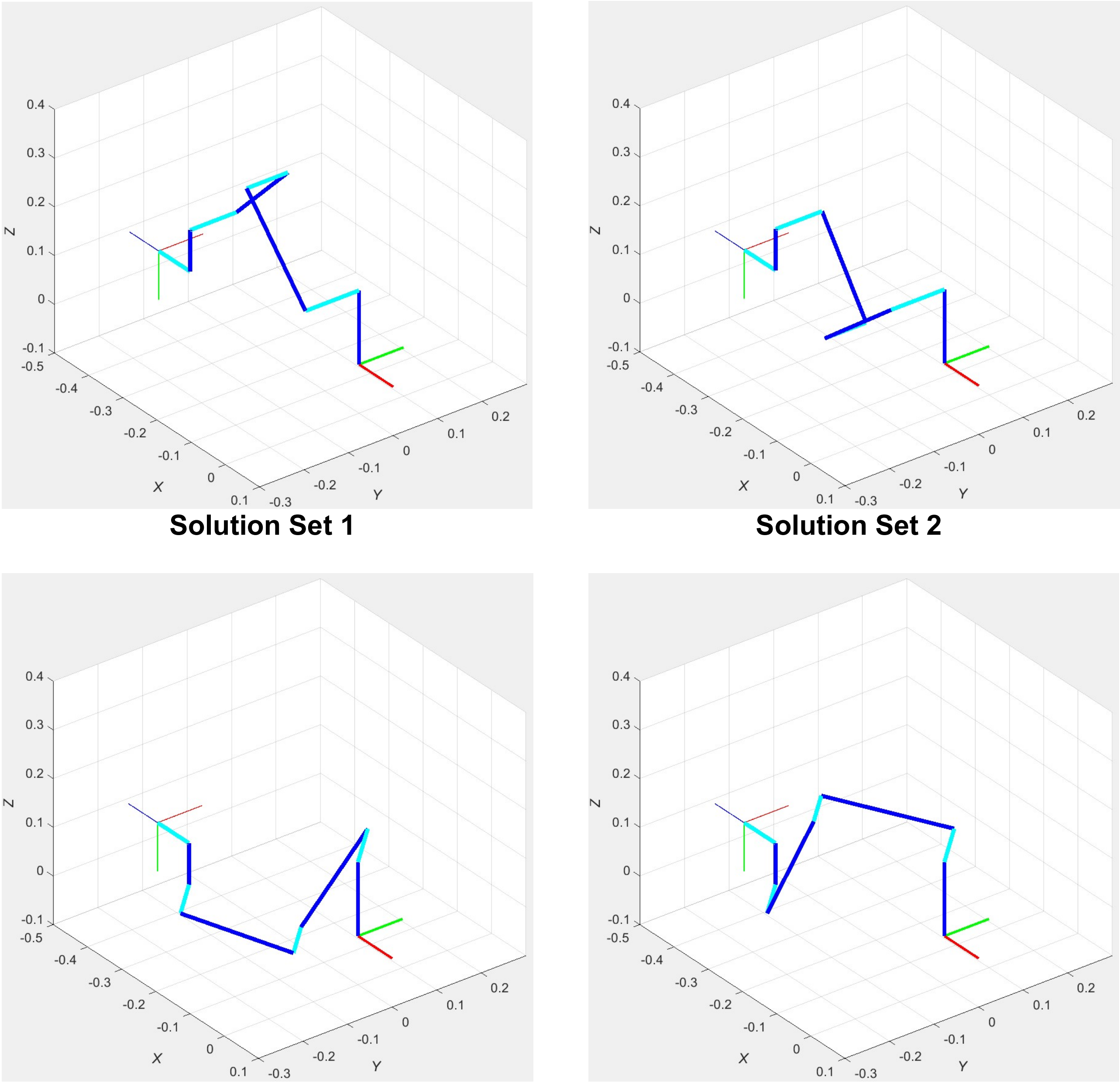
we first use a homogeneous transformation equation $\left[{}^{0}_{6}T\right]=\left[{}^{B}_{0}T\right]^{-1}\left[{}^{B}_{TP}T\right]\left[{}^{6}_{TP}T\right]^{-1}$ with constant matrices $\left[{}^{B}_{0}T\right],\left[{}^{6}_{TP}T\right]$ to find:

$$\left[{}^{0}_{6}T\right]=\begin{bmatrix}0 & 0 & -1 & -0.3231\\ 1 & 0 & 0 & -0.1310\\ 0 & -1 & 0 & -0.0631\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The IPK results are:

Solution Set	t_1 / t_2 sign	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	elbow
1	+ / –	0	45	90	45	90	0	up
2	+ / +	0	127.2	–90	142.8	90	0	down
3	– / –	–135.9	–150.5	78.1	72.4	45.9	180	down
4	– / +	–135.9	–78.7	–78.1	156.8	45.9	180	up

As seen in the figures on the following page, all four solution sets achieve the same correct commanded Cartesian pose. The four solutions are formed by permutations of the ± in the solutions for joint angles θ_1 and θ_2. The second column in the table above reports the signs used in the quadratic formula of t_1 and t_2, leading to θ_1 and θ_2. The last column in the table above reports the elbow condition of each pose, i.e. elbow-up or elbow-down. These results are also clearly visible in the figures on the following page.



Solution Set 1

Solution Set 2

Solution Set 3

Solution Set 4

IPK Example 3, Four Possible Solution Sets Poses

7. Ure-series Velocity Kinematics and Resolved-Rate Control

An attractive alternative to the Inverse Pose Kinematics (IPK) problem for control of serial robot arms is the resolved-rate control method (Whitney, 1969), based on the inverse velocity solution. The inverse velocity solution uses a linear set of equations which can easily be solved in a control loop at real-time rates. This section presents the URe-Series velocity kinematics, including the Jacobian matrix, followed by the resolved-rate control method.

7.1 Ure-series Cobot Jacobian Matrix

The Jacobian matrix [*J*] is a linear transformation mapping joint rates $\{\dot{\Theta}\}$ to Cartesian velocities $\{\dot{X}\}$:

$$\left\{{}^{k}\dot{X}\right\} = \left[{}^{k}J(\Theta)\right]\left\{\dot{\Theta}\right\}$$

$$m\times 1 \equiv (m\times n)(n\times 1)$$

Where *m* is the dimension of the Cartesian (task) space, *n* is the dimension of the joint space, and we can express the resulting Cartesian velocities in any frame $\{k\}$; $\{\dot{\Theta}\}$ are the relative joint angle rates and hence are expressed about the *n* different local *Z* axes. The Jacobian matrix is a function of the *n* joint angles Θ, in general; therefore, it must be calculated anew with each motion.

The Jacobian matrix is a multi-dimensional form of the derivative:

$$\left[J\right] = \left[\frac{\partial f_i}{\partial \theta_j}\right]$$

where f_i are the six pose functions, and θ_j are the seven joint angles. A thousand and one references state this about the Jacobian matrix, but it is only half-true. It works well for translational terms, where:

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} = \begin{Bmatrix} x(\Theta) \\ y(\Theta) \\ z(\Theta) \end{Bmatrix}$$

but there are no possible functions with respect to which we can take the partial derivatives to obtain the rotational terms of the Jacobian matrix. The rotational terms may be found using a relative angular velocity vector equation.

For the 6-joint URe-Series spatial Cobot arm, $m = n = 6$:

$$\left\{\dot{\Theta}\right\} = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{Bmatrix} \qquad \left\{{}^{k}\dot{X}\right\} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}$$

Physical interpretation of the Jacobian matrix

The i^{th} Jacobian matrix column is the end-effector translational and rotational velocity due to joint i, with the joint rate $\dot{\theta}_i$ factored out. Then by linear superposition, the overall end-effector Cartesian velocity is the sum of all n columns (each multiplied by the respective joint rate). Each Jacobian matrix column i is the absolute Cartesian velocity vector of the last active joint frame $\{N\}$ with respect to the base frame, due to joint i only, and with the variable joint rate $\dot{\theta}_i$ factored out (see Figure 11).

$$^{k}[J] = {}^{k}\begin{bmatrix} | & | & \cdots & | \\ \left\{{}^{0}\dot{X}_N\right\}_1 & \left\{{}^{0}\dot{X}_N\right\}_2 & \cdots & \left\{{}^{0}\dot{X}_N\right\}_N \\ | & | & \cdots & | \end{bmatrix}$$

$$^{k}\left\{{}^{0}\dot{X}_N\right\}_i = \begin{Bmatrix} {}^{k}\left\{{}^{0}V_N\right\}_i \\ {}^{k}\left\{{}^{0}\omega_N\right\}_i \end{Bmatrix}$$

Here is Jacobian matrix column i, for a revolute joint:

$$^{k}\{J\}_i = \begin{Bmatrix} \left\{{}^{k}\hat{Z}_i\right\} \times {}^{k}\left\{{}^{i}P_N\right\} \\ \left\{{}^{k}\hat{Z}_i\right\} \end{Bmatrix} = \begin{Bmatrix} \left[{}^{k}_{i}R\right]\left\{{}^{i}\hat{Z}_i \times {}^{i}P_N\right\} \\ \left[{}^{k}_{i}R\right]\left\{{}^{i}\hat{Z}_i\right\} \end{Bmatrix}$$

where:

$$\left\{{}^{k}\hat{Z}_i\right\} = \left[{}^{k}_{i}R\right]\left\{{}^{i}\hat{Z}_i\right\}$$

is the third column of orthonormal rotation matrix $\left[{}^{k}_{i}R\right]$ and:

$$^{k}\left\{{}^{i}P_N\right\} = \left[{}^{k}_{i}R\right]{}^{i}\left\{{}^{i}P_N\right\}$$

where ${}^{i}\left\{{}^{i}P_N\right\}$ is the translational part of homogeneous transformation matrix $\left[{}^{i}_{N}T\right]$ (the fourth column, rows 1 through 3). Here is the Jacobian matrix for an all-revolute-joint manipulator:

$$ {}^{k}[J] = {}^{k}\left[\begin{Bmatrix} \left\{ {}^{k}\hat{Z}_1 \right\} \times {}^{k}\left\{ {}^{1}P_N \right\} \\ \left\{ {}^{k}\hat{Z}_1 \right\} \end{Bmatrix} \quad \cdots \quad \begin{Bmatrix} \left\{ {}^{k}\hat{Z}_i \right\} \times {}^{k}\left\{ {}^{i}P_N \right\} \\ \left\{ {}^{k}\hat{Z}_i \right\} \end{Bmatrix} \quad \cdots \quad \begin{Bmatrix} \left\{ {}^{k}\hat{Z}_N \right\} \times {}^{k}\left\{ {}^{N}P_N \right\} \\ \left\{ {}^{k}\hat{Z}_N \right\} \end{Bmatrix} \right] $$

For the 6-dof URe-Series cobot, $N = 6$, $i = 1,2,3,4,5,6,$, and the Jacobian and Cartesian velocity frame of expression (basis) is any convenient coordinate frame k. Often k is chosen to be $\{0\}$.

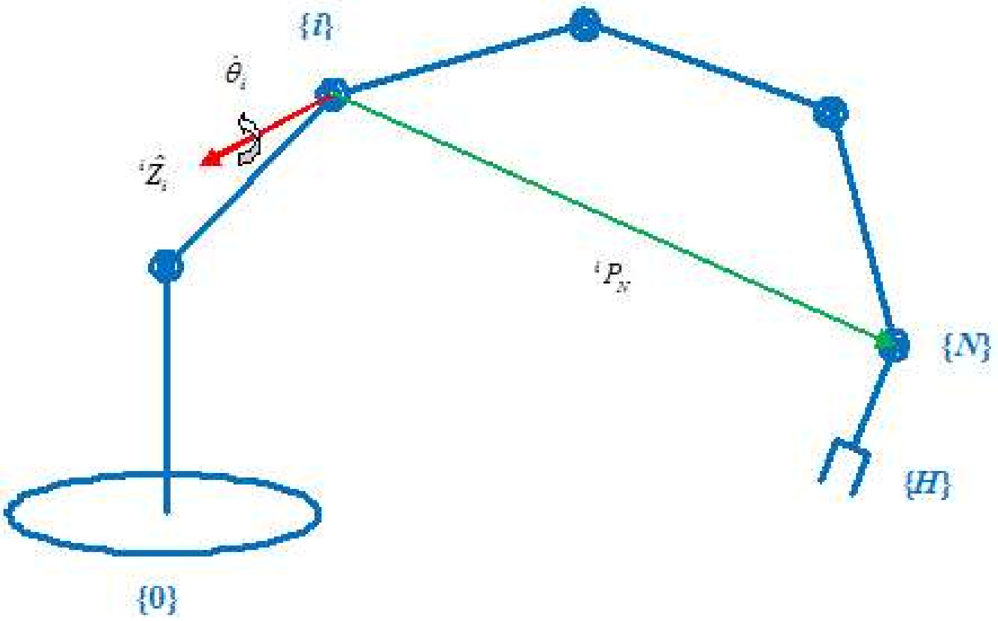


Figure 14. Jacobian Matrix Column *i* Derivation Image, Revolute Joints

The Universal Robot URe-Series 6-dof Cobot Jacobian matrix is given below. This Jacobian matrix expresses the velocity of $\{6\}$ with respect to $\{0\}$, expressed in $\{0\}$ coordinates.

$$ {}^{0}\left[{}^{0}J_6 \right] = \begin{bmatrix} d_4c_1 + s_1(a_2s_2 + a_3s_{23} + d_5s_{234}) & -c_1(a_2c_2 + a_3c_{23} + d_5c_{234}) & -c_1(a_3c_{23} + d_5c_{234}) & -d_5c_1c_{234} & 0 & 0 \\ d_4s_1 - c_1(a_2s_2 + a_3s_{23} + d_5s_{234}) & -s_1(a_2c_2 + a_3c_{23} + d_5c_{234}) & -s_1(a_3c_{23} + d_5c_{234}) & -d_5s_1c_{234} & 0 & 0 \\ 0 & -a_2s_2 - a_3s_{23} - d_5s_{234} & -a_3s_{23} - d_5s_{234} & -d_5s_{234} & 0 & 0 \\ 0 & s_1 & s_1 & s_1 & -c_1s_{234} & s_1c_5 + c_1c_{234}s_5 \\ 0 & -c_1 & -c_1 & -c_1 & -s_1s_{234} & -c_1c_5 + s_1c_{234}s_5 \\ 1 & 0 & 0 & 0 & c_{234} & s_{234}s_5 \end{bmatrix} $$

Cartesian Transformation of Translational and Rotational Velocities

The Jacobian matrix presented above relates the frame $\{6\}$ translational and rotational velocities with respect to the $\{0\}$ frame. For real-world applications, it is more useful to command translational and rotational velocities at the tool-plate (or end-effector frame) $\{TP\}$ instead. The same Jacobian may be used if the following velocity transformations are used first:

$$\{V_6\} = \{V_{TP}\} - \{\omega_6\} \times \{{}^6P_{TP}\}$$
$$\{\omega_6\} = \{\omega_{TP}\}$$

That is, since link 6 is a rigid link containing both {6} and {*TP*}, the rotational velocity vector is the same over the whole link. However, the angular velocity crossed into the position vector must be added to the translational velocity in {6} to yield that of {*TP*} (this statement must be reversed, i.e. subtracted, since we are given the velocities in {*TP*}). As always, a common basis frame, such as {0}, must be used above to ensure the coordinates of all vectors are expressed in a single frame (we could equally use {*B*} as the reference frame, this would make no different at all since L_B does not enter).

The reason we use the Jacobian matrix ${}^0\left[{}^0J_6\right]$ instead of ${}^0\left[{}^0J_{TP}\right]$ is simplicity. Length L_{TP} would appear all over ${}^0\left[{}^0J_{TP}\right]$, so ${}^0\left[{}^0J_6\right]$ is simpler.

Cartesian Wrench / Joint Torques Statics Transformation

It is well-known (Craig, 2005) that the relationship between static Cartesian wrenches (forces / moments) applied to the environment by the robot end-effector and the required robot joint torques to do this are calculated as follows:

$$\{\tau\} = [J]^T \{W\}$$

Where $\{\tau\} \equiv n \times 1$ is the vector of *n* joint torques, $\{W\} \equiv m \times 1$ is the Cartesian wrench (*m* forces and moments), *n* is the joint space dimension, and *m* is the Cartesian space dimension. The Jacobian matrix [*J*] and Cartesian wrench {*W*} must be expressed in the basis coordinates of the same frame {*k*}. Just like the joint rates, {τ} has no dependence on frame, since these are relative joint torques about the *n Z* axes.

The Jacobian matrix [*J*] for static torque calculations is the same as that for velocity analysis. This statics transformation is a mapping from Cartesian space to joint space which does not require an inverse. That is indeed a rare and beautiful property. It can never be singular and any number of joints is allowed. Very little computation is required compared to matrix inversion, since only matrix transposition and multiplication is required. Note that the associated wrench is applied at the frame for which the Jacobian matrix was derived for velocities, {6} in this work. Therefore, another transformation is required when the robot should apply the wrench at the tool-plate {*TP*} (again using a common basis):

$$\{F_6\} = \{F_{TP}\}$$
$$\{M_6\} = \{M_{TP}\} + \{{}^6P_{TP}\} \times \{F_{TP}\}$$

Crucial – Units

The translational rows of the Jacobian matrix have length units (m). The rotational rows of the Jacobian matrix are unitless. Therefore, since Cartesian velocity units of $\{\dot{X}\}$ are m/sec and rad/sec, for translational and rotational terms, respectively, in the overall velocity equation $\{\dot{X}\} = [J]\{\dot{\Theta}\}$, one MUST use units of rad/sec for $\{\dot{\Theta}\}$, not deg/sec!

Further, the units and size mismatch between translational and rotational rows of the Jacobian matrix can cause numerical troubles, which is well-known for serial robots. Using *m* rather than *mm* for length units will help this problem for the URe-Series Cobot arms.

Universal URe-Series Cobot Singularity Analysis

Serial robot singularities occur when the determinant of the velocity Jacobian matrix goes to zero. For serial robots, a physical singularity is associated with a loss of motion in the Cartesian space. (This is not an issue in the joint space, only Cartesian → joint space transformations.) Safety issues arise, as joint rates tend towards infinity in the neighborhood of singularities, and the resulting motion is not what the engineer expects it to be.

The determinant of the Universal URe Cobots Jacobian matrix $\left[{}^{0}J\right]$ is:

$$\left|{}^{0}J\right| = -a_2 a_3 s_3 s_5 (a_2 s_2 + a_3 s_{23} + d_5 s_{234})$$

When this Jacobian matrix determinant is zero, the 6-dof serial arm is at a robot singularity. Since cobot lengths $a_2 \neq 0$ and $a_3 \neq 0$, there are three singularity conditions:

$$\sin\theta_3 = 0$$

$$\sin\theta_5 = 0$$

$$a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) + d_5 \sin(\theta_2 + \theta_3 + \theta_4) = 0$$

<u>Singularity Conditions</u>

1. When $\sin\theta_3 = 0$, the singularity condition is $\theta_3 = -360^\circ, -180^\circ, 0^\circ, 180^\circ, 360^\circ$.

   This singularity is the classic elbow-out, workspace-boundary, singularity common to all elbow-based serial robot arms. It is not much of a problem, since it is on the edge of the useful workspace, where the cobot is not used effectively anyway.

The cobot kinematic diagram of Figure 6, with all angles zero, is in this elbow-out, workspace-boundary singularity. This singularity is the reason for the choice of $\theta_3 = 90^\circ$ initial angle shown in Table 9, i.e. as far away from this elbow-out singularity as possible.

2. When $\sin\theta_5 = 0$, the singularity condition is $\theta_5 = -360^\circ, -180^\circ, 0^\circ, 180^\circ, 360^\circ$.

   Though it appears similar to 1. above, this singularity is more troubling since it is a workspace-interior singularity, and thus reduces the effective workspace. At this singularity, the wrist axes Z_4 and Z_6 are parallel, meaning instantaneously there is no way to perform a wrist yaw motion. Geometrically, in any pose with this singularity condition, joints 2, 3, 4, and 5 are free to move as a four-bar mechanism while the end-effector frame remains stationary. This is very interesting but most troubling for safety.

   The cobot kinematic diagram of Figure 6, with all angles zero, is in this internal wrist singularity, a most clumsy wrist design where there is a damaging singularity in the middle of the useful wrist motion range (the classic 6-dof 6R PUMA serial robot has a similar singularity). This singularity is the reason for the choice of $\theta_5 = 90^\circ$ initial angle shown in Table 9, i.e. as far away from this internal wrist singularity as possible.

3. When $a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) + d_5 \sin(\theta_2 + \theta_3 + \theta_4) = 0$, the specific angle values causing this singularity are much harder to enumerate. Therefore, we will explain this singularity using geometry of the cobot serial chain.

   This third singularity condition results when the shared origin of the last two wrist Cartesian coordinate frames {5} and {6} lies on the plane formed by the first two rotational axes Z_1 and Z_2. Poses with this shoulder singularity condition are where the two IPK solution branches for θ_1 degenerate to one solution. Alarmingly, this singularity allows the cobot to rotate freely without control about the vertical axis, a big safety challenge!

   The initial cobot angles given in Table 9 were chosen so this shoulder singularity is not near singularity condition 3. It is not as far away as possible, since that would require the cobot end-effector being place on the workspace boundary. Not only would that be singularity condition 1, but it would not be a useful initial cobot pose for general tasks.

These three singularity conditions are displayed in the CAD poses of Figure 12.

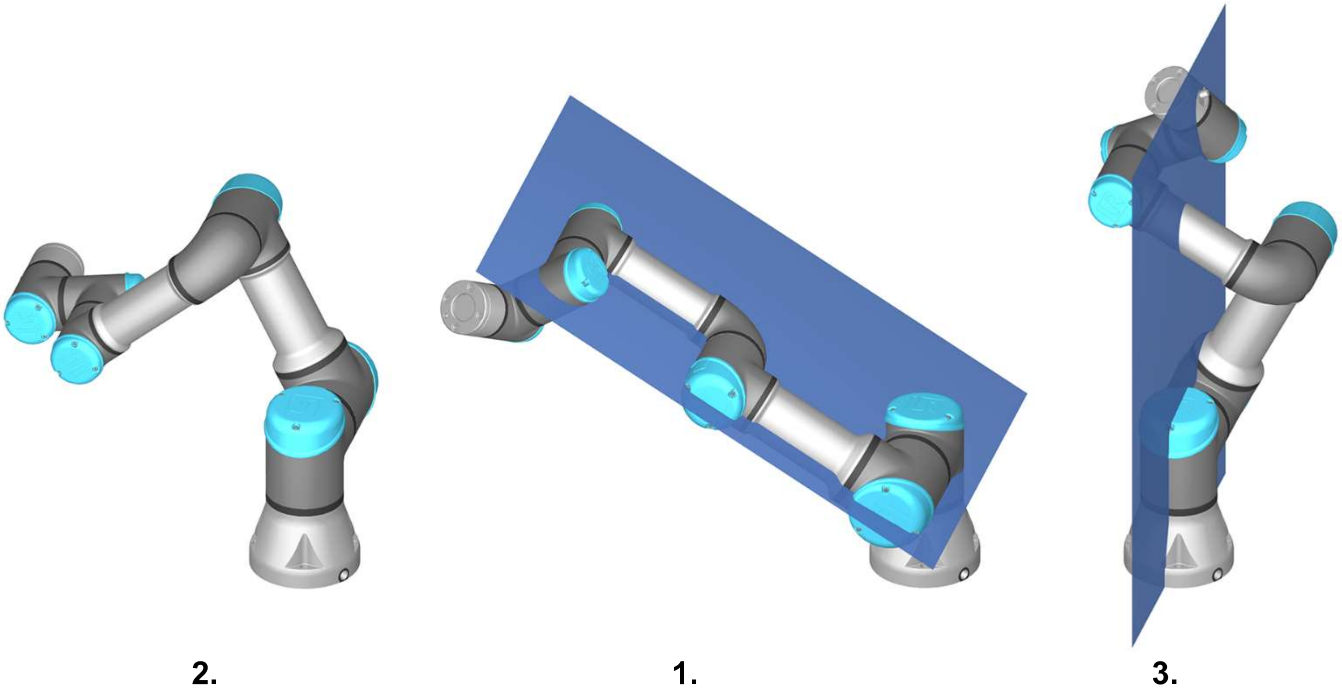


Figure 15. URe-Series Cobots Singularity Conditions

www.mecademic.com/resources/Singularities

Note that the third singularity condition formula is an important term in the FPK solution for both ${}^{0}x_{6}$ and ${}^{0}y_{6}$, as presented earlier.

One disadvantage of the generous $\pm 360^{\circ}$ joint limits on all joints of the URe Cobots is the possibility of much greater exposure to singularities. Exact singularities are not the only problem; when the cobot approaches the neighborhood of singularities, the joint rates can increase to unacceptable levels. One, two, or three of these singularity conditions can occur simultaneously. Again, there is not a singularity problem in joint-space. However, in lead-through programming control (where sets of joint angles are played back over time by the cobot controller after a human operator has taught the required cobot poses by physically moving the end-effector), the operator must avoid singularities, otherwise such undesired motions would interfere with the desired trajectories.

Each singularity condition must be understood and avoided in the case of Inverse Pose Control and/or Resolved-Rate Control. Often engineers arrange the task and cobot placement to totally avoid singularities during required Cartesian motions. Even though this can be done off-line with CAD and VR programs, it is painstaking and always leads to less performance from the cobot compared to what is possible. Another way to face and deal with singularities is to numerically monitor $\left|{}^{0}J\right|$ in real-time in the cobot controls programming. When joint rates are found to be increasing too high and too fast, these joint rates may be capped at a safe level. Then the Cartesian motion will not be correct, but the cobot can get back on track after the singularity is passed and joint rates are again safe.

7.2 Ure-series Resolved-Rate Control

Another useful application for the Jacobian matrix is the **Inverse Velocity Problem**, whose solution is the basis for the **Resolved-Rate Control Algorithm** (Whitney, 1969). The Inverse Velocity Problem is stated, in general:

Given:

The robot (including all constant DH Parameters), values for all of the joint variables $\{\Theta\}$ (angles θ_i for **R** joints and lengths d_i for **P** joints), and the end-effector translational and rotational Cartesian velocity $\left\{\dot{X}\right\}$;

Find:

All of the relative joint rates $\left\{\dot{\Theta}\right\}$ (joint rates $\dot{\theta}_i$ for **R** joints and $\dot{d}_i$ for **P** joints).

For the *m* = *n* case, the Inverse Velocity Solution is (for the UR3-Series Cobots, *m* = *n* = 6):

$$\left\{\dot{\Theta}\right\} = \left[{}^{k}J(\Theta)^{-1}\right]\left\{{}^{k}\dot{X}\right\}$$

where we calculate the required relative joint rates $\left\{\dot{\Theta}\right\}$ to achieve the given desired Cartesian velocities $\left\{{}^{k}\dot{X}\right\}$, using the inverse of the configuration-dependent Jacobian matrix $\left[{}^{k}J(\Theta)\right]$. This only works for an *m* = *n* square matrix, assuming full rank, i.e. the Jacobian matrix determinant is not zero. Here are two options to solve the Inverse Velocity problem in MATLAB. In practical real-time implementations, use of numerical Gaussian elimination is more computationally efficient and more robust in the neighborhood of singularities.

Figure 16 shows the Resolved-Rate Control simulation block diagram; note that in simulation, we will assume perfect joint control, i.e. the actual joint values are the same as the commanded joint values, $\left\{\Theta_A\right\} \rightarrow \left\{\Theta\right\}$. Note that the Jacobian matrix must always be updated during resolved-rate motion simulation since it is a function of $\left\{\Theta\right\}$.

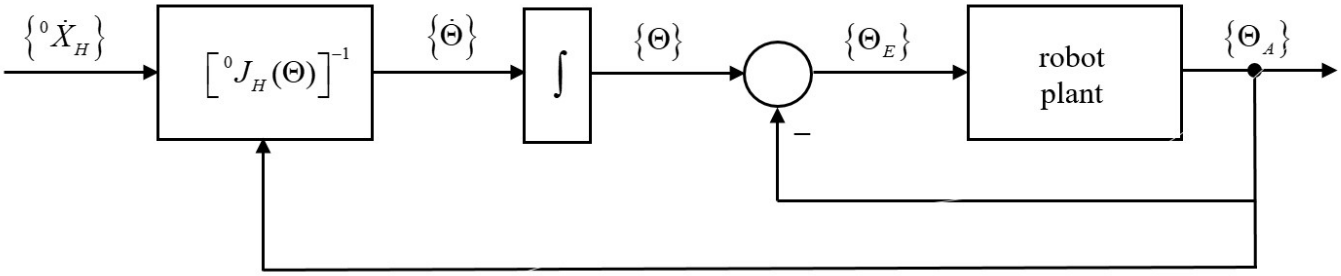

$\left\{{}^{0}\dot{X}_H\right\}$
$\left[{}^{0}J_H(\Theta)\right]^{-1}$
$\left\{\dot{\Theta}\right\}$
$\int$
$\left\{\Theta\right\}$
$\left\{\Theta_E\right\}$
robot plant
$\left\{\Theta_A\right\}$
–

Figure 16. Resolved-Rate Control Block Diagram

Figure 17 shows the Resolved-Rate Control simulation algorithm flowchart for programming.

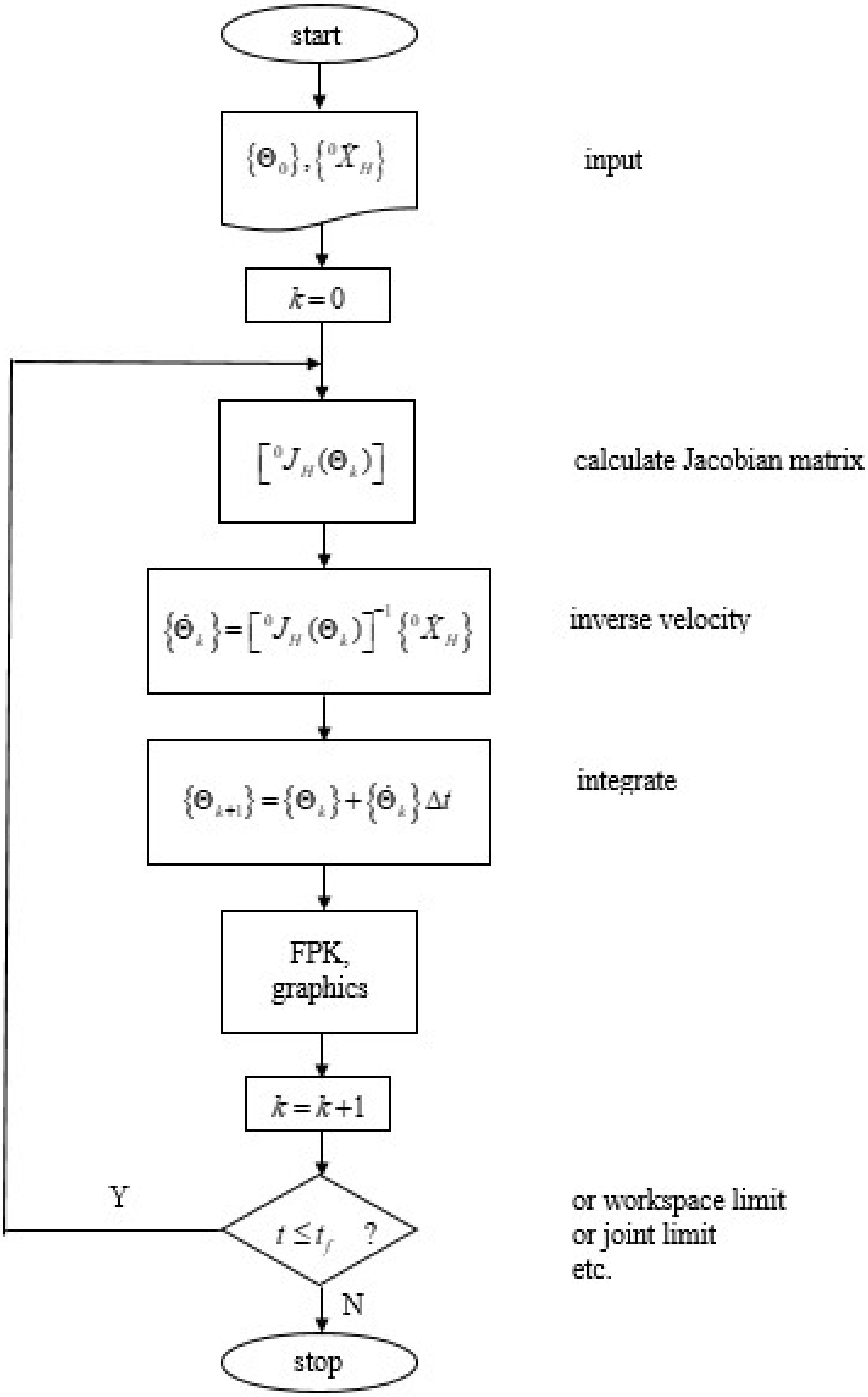

start
$\{\Theta_0\},\{{}^0\dot{X}_H\}$
input
$k=0$
$\left[{}^0J_H(\Theta_k)\right]$
calculate Jacobian matrix
$\{\dot{\Theta}_k\}=\left[{}^0J_H(\Theta_k)\right]^{-1}\{{}^0\dot{X}_H\}$
inverse velocity
$\{\Theta_{k+1}\}=\{\Theta_k\}+\{\dot{\Theta}_k\}\Delta t$
integrate
FPK, graphics
$k=k+1$
Y
$t\le t_f$?
or workspace limit
or joint limit
etc.
N
stop

Figure 17. Resolved-Rate Algorithm Flowchart

6-dof Universal UR3e Resolved-Rate Control Simulation Example

Starting from the Initial Joint Angles for the Universal UR3e Cobot (from FPK Example 3 and the IPK Example):

$$\{\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6\} = \{0 \quad 45^\circ \quad 90^\circ \quad 45^\circ \quad 90^\circ \quad 0\}$$

the following Cartesian velocities and wrench are applied at the tool-plate frame $\{TP\}$, motion with respect to the $\{0\}$ frame, and in the basis frame of $\{0\}$:

$$^0\left\{{}^0\dot{X}_{TP}\right\} = \begin{Bmatrix} ^0\left\{{}^0V_{TP}\right\} \\ ^0\left\{{}^0\omega_{TP}\right\} \end{Bmatrix} = \begin{Bmatrix} 0.08 \\ 0.10 \\ 0.12 \\ 0.1 \\ 0.2 \\ 0.3 \end{Bmatrix} \qquad {}^0\left\{{}^0W_{TP}\right\} = \begin{Bmatrix} \{F_{TP}\} \\ \{M_{TP}\} \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 0.4 \\ 0.5 \\ 0.6 \end{Bmatrix}$$

(m/sec and rad/sec) (N and Nm)

Assuming the input velocity and wrench are constant, the resolved-rate simulation in MATLAB yielded the following plots, running from 0 to 4 sec, using time steps of 0.04 sec.

The first step is to apply the velocity and wrench transformations from $\{TP\}$ to $\{6\}$, given earlier. The reason for this is that the Jacobian matrix is simpler for motions of $\{6\}$ rather than $\{TP\}$.

$$^0\left\{{}^0\dot{X}_6\right\} = \begin{Bmatrix} ^0\left\{{}^0V_6\right\} \\ ^0\left\{{}^0\omega_6\right\} \end{Bmatrix} = \begin{Bmatrix} 0.0616 \\ 0.1092 \\ 0.1200 \\ 0.1 \\ 0.2 \\ 0.3 \end{Bmatrix} \qquad {}^0\left\{{}^0W_6\right\} = \begin{Bmatrix} \{F_6\} \\ \{M_6\} \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 0.2160 \\ 0.5920 \\ 0.6000 \end{Bmatrix}$$

(m/sec and rad/sec) (N and Nm)

The final UR3e Cobot pose, after the specific resolved-rate motion simulation, is given in Figure 18a. The initial UR3e Cobot pose for the resolved-rate motion simulation was already given in Figure 9 from FPK Example 3 (and Solution Set 1 from the IPK Example).

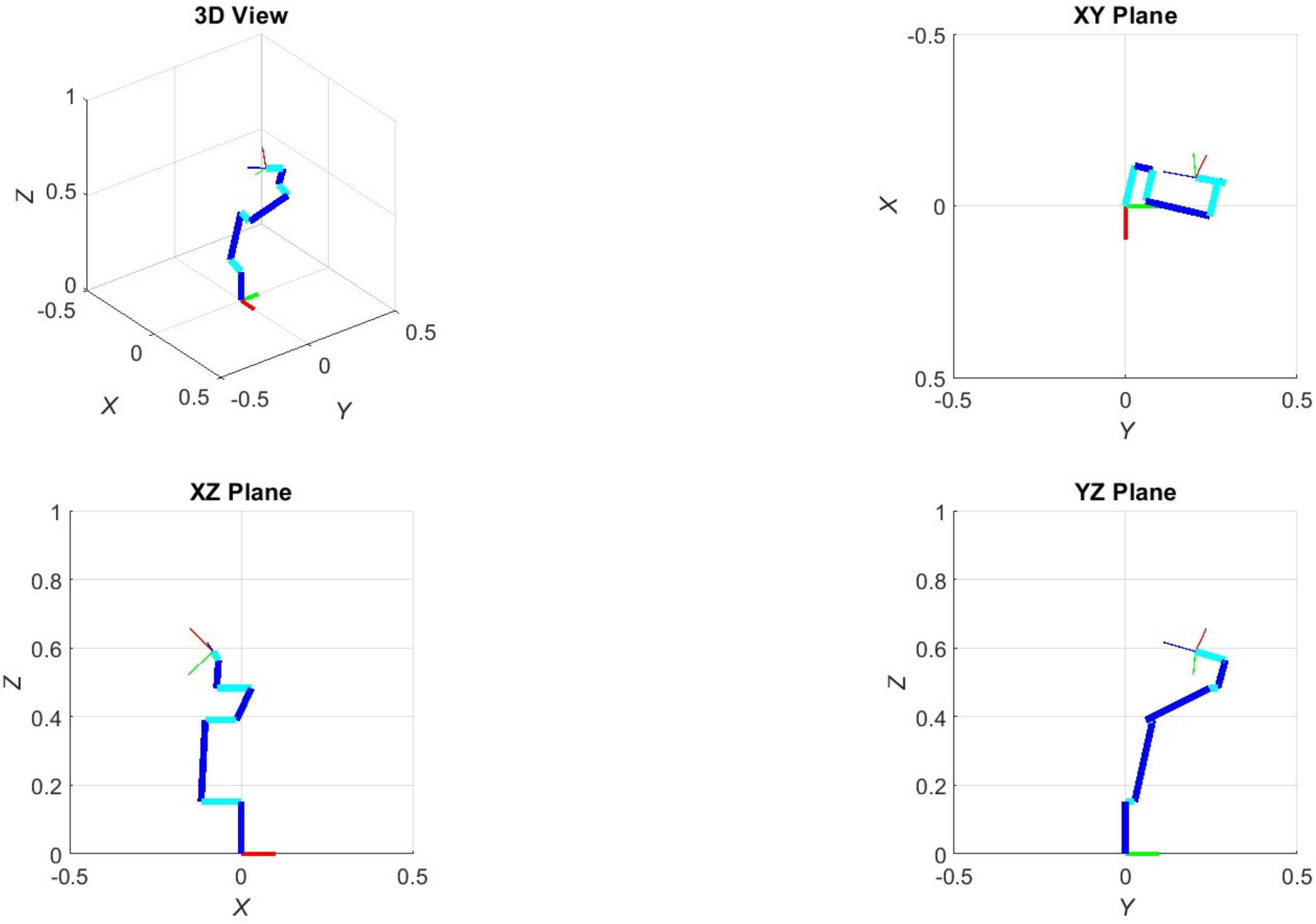


Figure 18a. Resolved-Rate Simulation, Final Cobot Pose

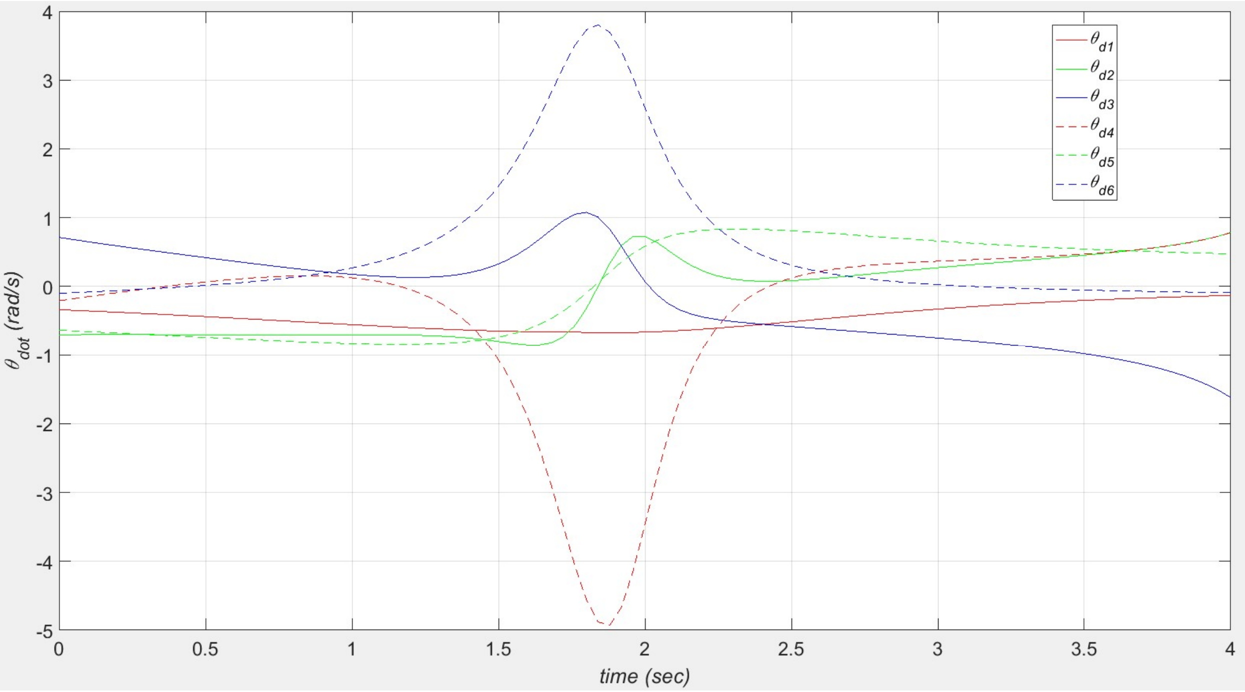


Figure 18b. Resolved-Rate Simulation, Joint Rates

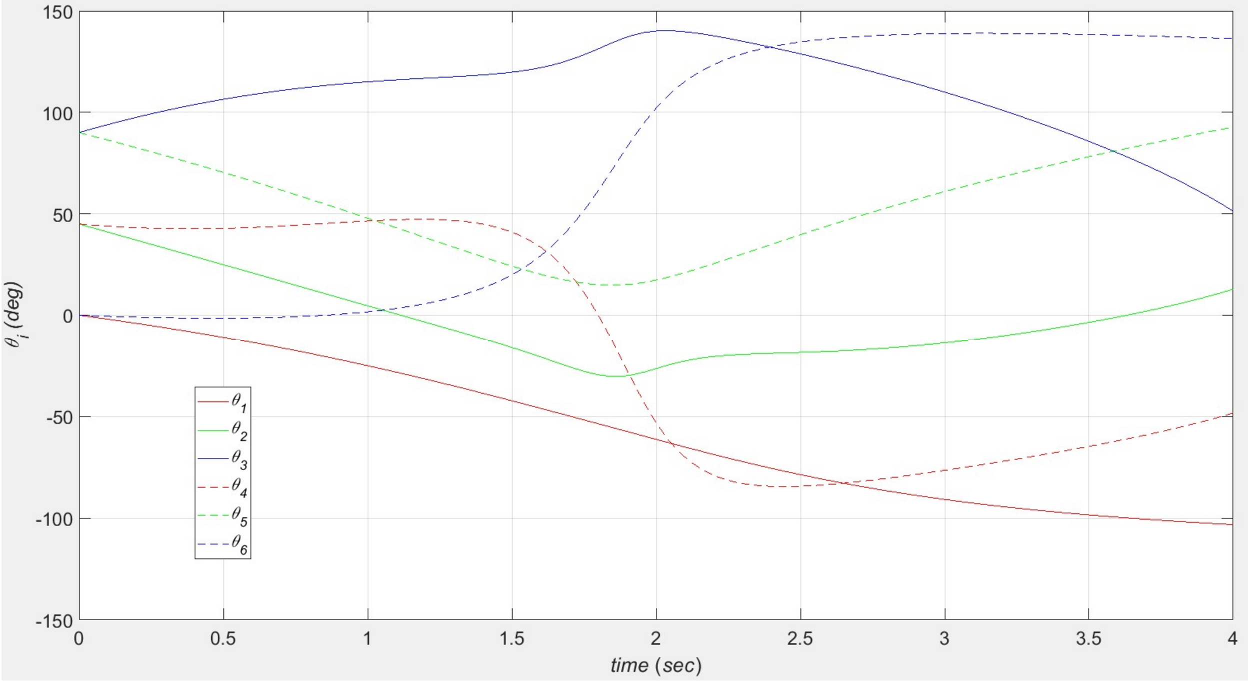


Figure 18c. Resolved-Rate Simulation, Joint Angles

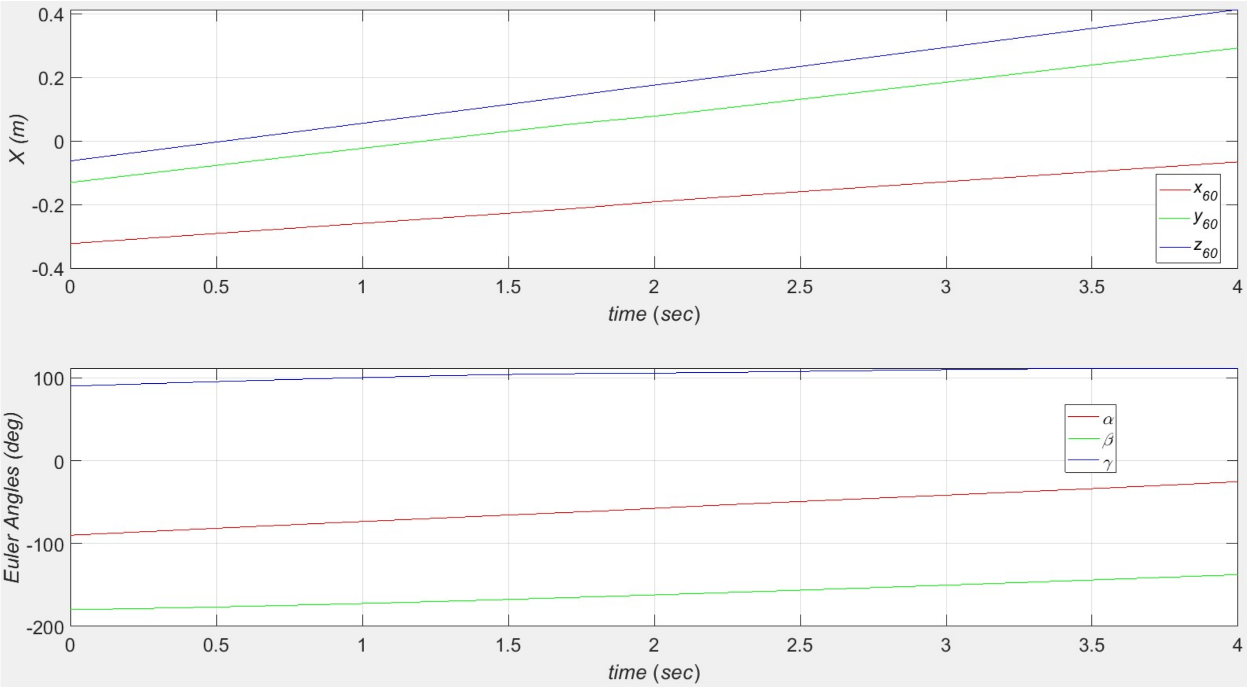


Figure 18d. Resolved-Rate Simulation, Cartesian Displacements

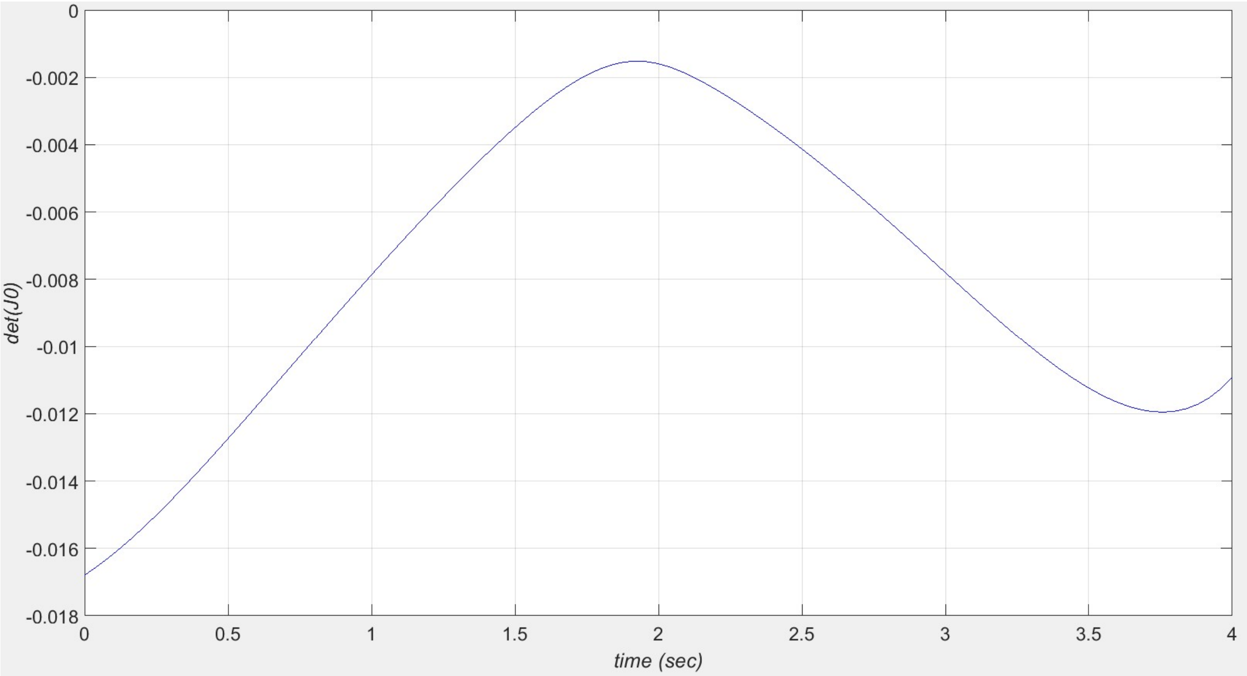


Figure 18e. Resolved-Rate Simulation, $\left| {}^{0}\left[{}^{0}J_{6} \right] \right|$ **Determinant**

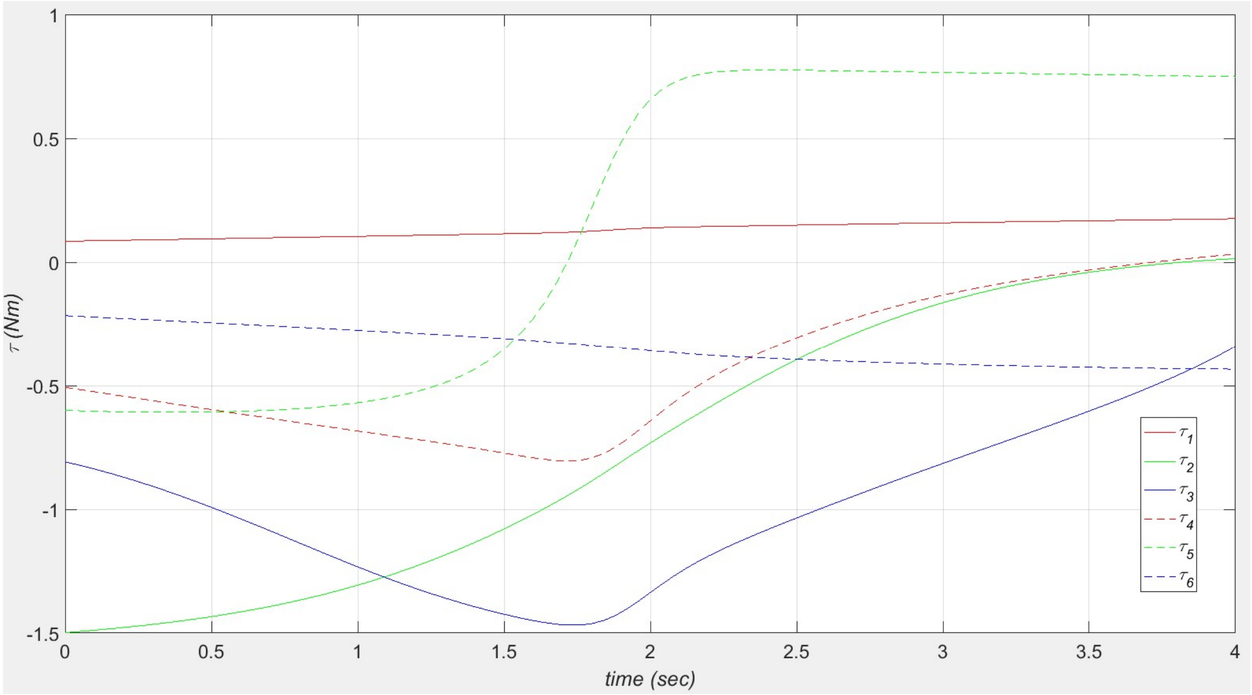


Figure 18f. Resolved-Rate Simulation, Pseudostatic Joint Torques

Figures 18b – 18f show the plot results from this UR3e Cobot resolved-rate motion MATLAB simulation.

Figure 18b shows the required joint rates to achieve the commanded Cartesian rates over the 4 second simulation time period, followed by the associated joint angles (Figure 18c), integrated from the joint rates. Figure 18d shows the 6-dof Cartesian displacements, calculated via the FPK solution; these demonstrate correct motion in the resolved-rate simulation since the constant slope of each Cartesian variable plot is the given constant Cartesian velocity. Figure 18e shows this simulation never reaches one of the three types of singularities identified, since $\left|{}^{0}\left[{}^{0}J_{6}\right]\right|$ never crosses through zero. However, near the middle of the motion time, the Cobot approaches the $\theta_5 = 0$ wrist singularity. This is verified in the θ_5 plot of Figure 18c. Lastly, Figure 18f shows the required six joint torques to exert the given constant Cartesian wrench, ignoring dynamics, at frame $\{TP\}$ onto the environment during this simulated motion (pseudostatics).

8. URe-Series Cobot Dynamics

Kinematics is the study of motion ***without*** regard to forces.

Dynamics is the study of motion ***with*** regard to forces. It is the study of the relationship between forces/torques and motion. Dynamics is composed of kinematics and kinetics.

a) Forward Dynamics (simulation) – given the actuator forces and torques, compute the resulting motion (this requires the solution of highly coupled, nonlinear ODEs): Given $\{\tau\}$, calculate $\{\theta\},\{\dot{\theta}\},\{\ddot{\theta}\}$ (all are N x 1 vectors).

b) Inverse Dynamics (control) – given the desired motion, calculate the actuator forces and torques (this linear algebraic solution is much more straight-forward than Forward Dynamics): Given $\{\theta\},\{\dot{\theta}\},\{\ddot{\theta}\}$, calculate $\{\tau\}$ (all N x 1 vectors).

Both problems require the *N* ***dynamic equations of motion***, one for each link, which are highly coupled and nonlinear. There are two basic methods for deriving the ***dynamic equations of motion***.

- Newton-Euler recursion (force balance, including inertial forces with D'Alembert's principle).
- Lagrange-Euler formulation (energy method).

Kinetics

Translational **Newton's Second Law**

Inertial force at center of mass

Rotational **Euler's Equation**

Inertial moment anywhere on body

The **kinematics** terms $\{a_{Ci}\},\{\omega_i\},\{\alpha_i\}$ must be moving with respect to an inertially-fixed frame. The frame of expression $\{k\}$ needn't be an inertially-fixed frame.

Assumptions

- serial robot
- rigid links
- ignore actuator dynamics
- no friction
- no joint or link flexibility

Tables 4 and 5 present the joint torque capacities of the 5 sizes of motors use throughout the URe-Series Cobots. Table 10 presents the remaining data required for dynamics analysis of each link throughout the URe-Series Cobots (mass and center-of-mass (COM) for each link).

Table 10: Universal Robot Ure-series Dynamics Data

link	UR3e		UR5e		UR10e	
	mass (kg)	COM (m)	mass (kg)	COM (m)	mass (kg)	COM (m)
Link 1	1.98	[0,–0.02,0]	3.76	[0,–0.02,0]	7.37	[0.021,0,0.027]
Link 2	3.44	[0.13,0,0.112]	8.06	[0.212,0,0.113]	13.05	[0.38,0,0.158]
Link 3	1.44	[0.05,0,0.024]	2.85	[0.15,0,0.027]	3.99	[0.24,0,0.068]
Link 4	0.87	[0,0,0.01]	1.37	[0,0,0.016]	2.10	[0,0.007,0.018]
Link 5	0.81	[0,0,0.01]	1.30	[0,0.002,0.016]	1.98	[0,0.007,0.018]
Link 6	0.26	[0,0,–0.02]	0.37	[0,0,–0.001]	0.62	[0,0,–0.026]

The mass-moments-of-inertia for each link (units: kg-m^2) are required for the following Newton-Euler numerical inverse dynamics simulation. These could not be found published on the Universal Robots website. The principal mass-moments-of-inertia for each link about its own center-of-mass (COM) may easily be estimated by assuming the material and using the formulae for a cylindrical distribution of the mass.

Newton-Euler Recursive Algorithm Summary

This method can be used to find the robot dynamics equations of motion. It can also be used to directly solve the inverse dynamics problem numerically. The summary of equations below, from Craig (2005), assume an all revolute-joint manipulator (prismatic joint dynamics have different equations).

Outward iteration for kinematics $i:0 \rightarrow N-1$

(without regard for frames of expression, for clarity)

Velocities and accelerations (kinematics)

$$\{\omega_{i+1}\} = \{\omega_i\} + \{\hat{Z}_{i+1}\}\dot{\theta}_{i+1}$$
$$\{\alpha_{i+1}\} = \{\alpha_i\} + \{\omega_i\} \times \{\hat{Z}_{i+1}\}\dot{\theta}_{i+1} + \{\hat{Z}_{i+1}\}\ddot{\theta}_{i+1}$$
$$\{a_{i+1}\} = \{a_i\} + \{\alpha_i\} \times \{{}^iP_{i+1}\} + \{\omega_i\} \times \left(\{\omega_i\} \times \{{}^iP_{i+1}\}\right)$$
$$\{a_{Ci+1}\} = \{a_{i+1}\} + \{\alpha_{i+1}\} \times \{{}^{i+1}P_{Ci+1}\} + \{\omega_{i+1}\} \times \left(\{\omega_{i+1}\} \times \{{}^{i+1}P_{Ci+1}\}\right)$$

Inertial loading (kinetics)

$$\{F_{i+1}\} = m_{i+1}\{a_{Ci+1}\}$$
$$\{N_{i+1}\} = \left[{}^CI\right]\{\alpha_{i+1}\} + \{\omega_{i+1}\} \times \left[{}^CI\right]\{\omega_{i+1}\}$$

Inward iteration for kinetics $i:N \rightarrow 1$

(without regard for frames of expression, for clarity)

Internal forces and moments

$$\{f_i\} = \{f_{i+1}\} + \{F_i\}$$
$$\{n_i\} = \{n_{i+1}\} + \{{}^iP_{Ci}\} \times \{F_i\} + \{{}^iP_{i+1}\} \times \{f_{i+1}\} + \{N_i\}$$

Externally applied joint torques

$$\tau_i = \{n_i\} \cdot \{Z_i\}$$

Inclusion of gravity forces

$$\{{}^0a_0\} = \{g\}$$

This is equivalent to a fictitious upward acceleration of $1g$ of the robot base, which accounts for the downward acceleration due to gravity (i.e. this conveniently includes the weight of all links).

If the dynamics equations-of-motion (EOM) were derived analytically in closed-form, the structure of the resulting equations is given below. These are the configuration space matrix/vector dynamics equations.

$$\{\tau\} = \left[M(\Theta)\right]\left\{\ddot{\Theta}\right\} + \left[B(\Theta)\right]\left\{\dot{\Theta}\dot{\Theta}\right\} + \left[C(\Theta)\right]\left\{\dot{\Theta}^2\right\} + \left\{G(\Theta)\right\}$$

$\left[M(\Theta)\right]$	$N \times N$ mass matrix; symmetric and positive definite	
$\left[B(\Theta)\right]$	$N \times \frac{N(N-1)}{2}$ Coriolis matrix	
$\left\{\dot{\Theta}\dot{\Theta}\right\}$	$\frac{N(N-1)}{2} \times 1$	$\left\{\dot{\theta}_1\dot{\theta}_2 \quad \dot{\theta}_1\dot{\theta}_3 \quad \cdots \quad \dot{\theta}_{N-1}\dot{\theta}_N\right\}^T$
$\left[C(\Theta)\right]$	$N \times N$ centripetal matrix	
$\left\{\dot{\Theta}^2\right\}$	$N \times 1$	$\left\{\dot{\theta}_1^2 \quad \dot{\theta}_2^2 \quad \cdots \quad \dot{\theta}_N^2\right\}^T$
$\left\{G(\Theta)\right\}$	$N \times 1$ vector of gravity terms	

9. Conclusion

This paper has presented detailed kinematic and dynamic analysis for the Universal Robots 6-dof URe-Series Cobot serial arms. The Craig (modified) Denavit-Hartenberg Parameters for each serial chain, specific length parameters, and joint angle limits were given. The general 6-dof forward pose kinematics (FPK) solutions were developed analytically. The analytical inverse pose kinematics (IPK) solution was given. The Jacobian matrix was presented, along with singularity analysis, and resolved-rate control simulations. MATLAB Examples were given for all kinematics developments.

References

J.J. Craig, 2005, Introduction to Robotics: Mechanics and Control, Third Edition, Pearson Prentice Hall, Upper Saddle River, NJ.

J. Denavit and R.S. Hartenberg, 1955, A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, Journal of Applied Mechanics: 215-221.

N. Kovincic, A. Muller, H. Gattringer, M. Weyrer, A. Schlotzhauer, and M. Brandstotter, 2019, Dynamic parameter identification of the Universal Robots UR5, Proceedings of the ARW & OAGM Workshop, DOI: 10.3217/978-3-85125-663-5-07.

D.L. Pieper, 1968, “The Kinematics of Manipulators Under Computer Control”, PhD thesis, Stanford University, Department of Mechanical Engineering.

D.E. Whitney, 1969, Resolved Motion Rate Control of Manipulators and Human Prostheses, IEEE Trans on Man-Machine Systems.