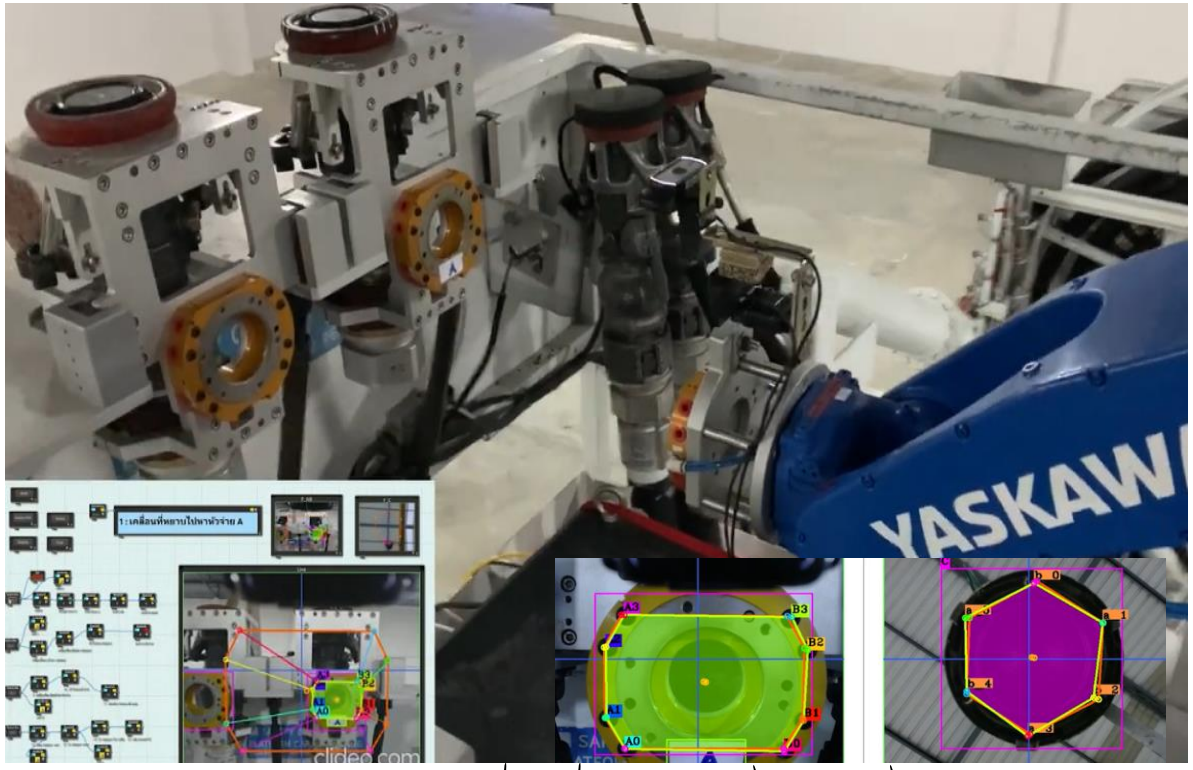
พื้นฐานหุ่นยนต์อุตสาหกรรม

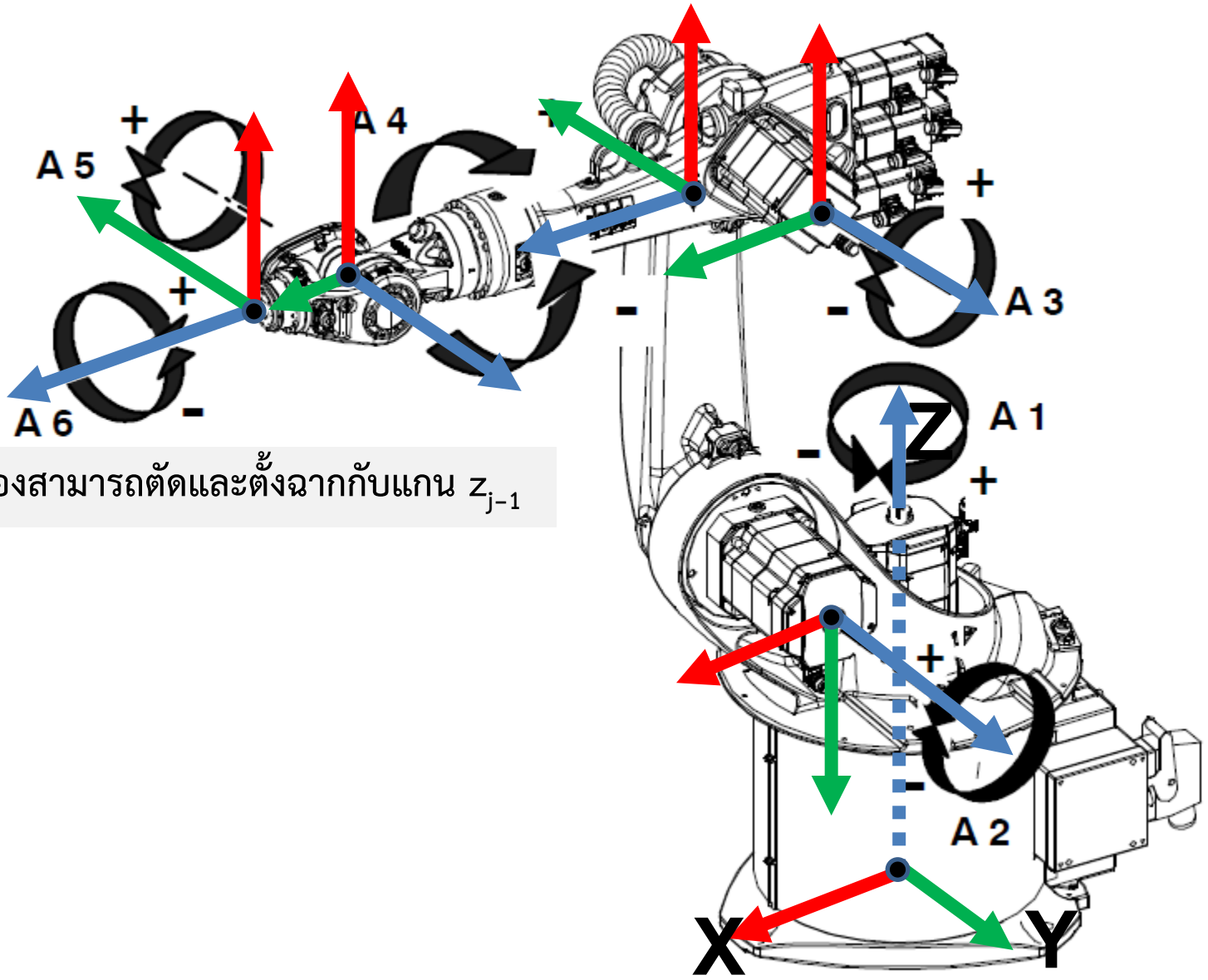
Fundamentals of Industrial Robotics



รศ.ดร.สันทัด ชูวงค์อินทร์
คณะเทคโนโลยีนวัตกรรมบูรณาการ

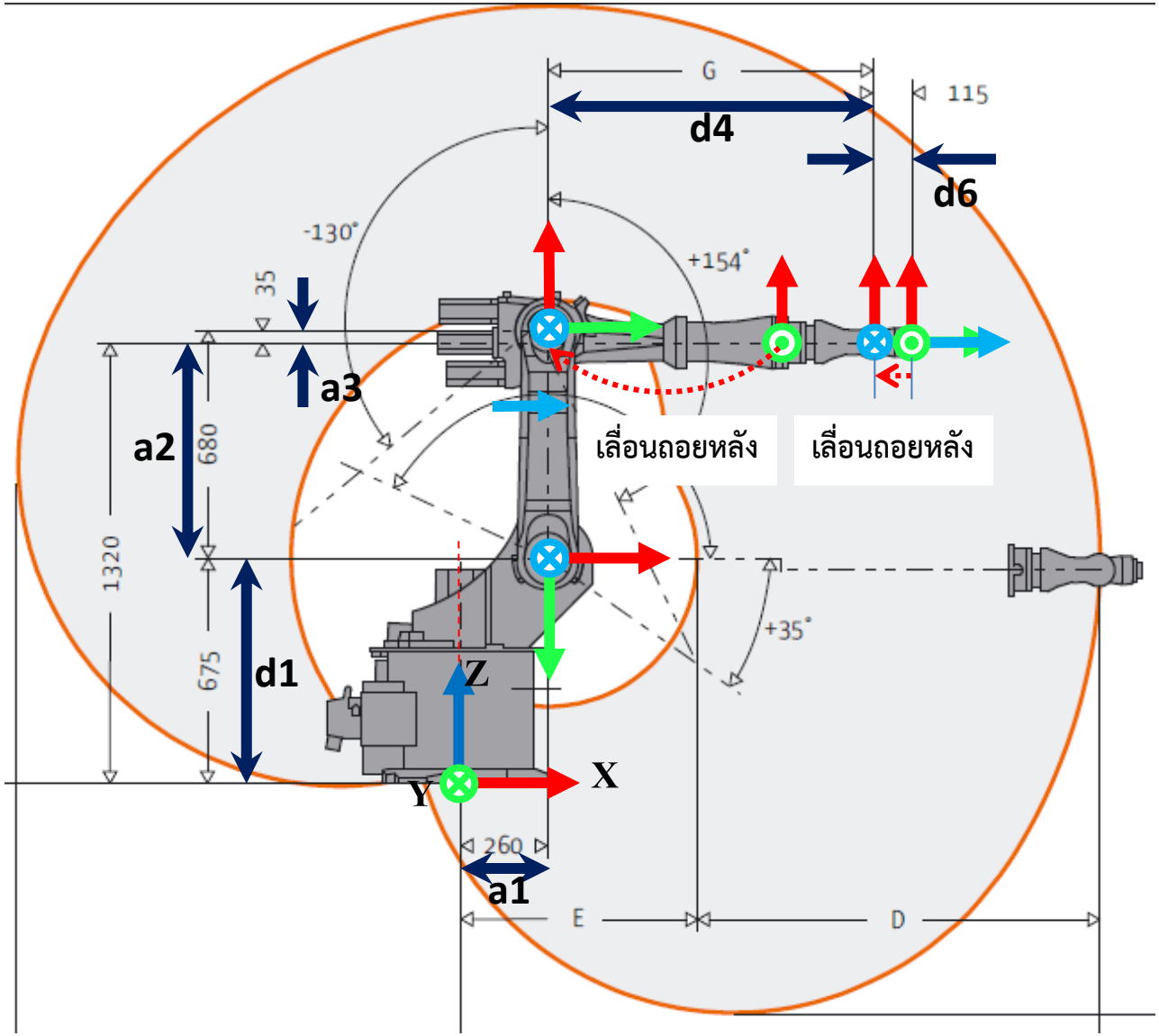
KUKA KR16L6

Denavit-Hartenberg Parameters



<u>ข้อบังคับ</u> : แนวแกน x_j ต้องสามารถตัดและตั้งฉากกับแกน z_{j-1}

Denavit-Hartenberg Parameters



Pose in 3D:Homogeneous Transformation Matrix of KUKA KR16

A1=0; A2=-90; A3=0; A4=0; A5=0; A6=0; #as figure

T1 = trotz(A1,'deg');

T2 = transl(0.26,0,0.675)@trotx(-90,'deg')@trotz(A2+90,'deg');

T3 = transl (0,-0.68,0)@trotz(-90,'deg')@trotz(A3,'deg');

T4 = transl(-0.035,0,0)@trotx(-90,'deg')@trotz(A4,'deg');

T5 = transl(0,0,0.97)@trotx(90,'deg')@trotz(A5,'deg');

T6 = transl(0,0.115,0)@trotx(-90,'deg')@trotz(A6,'deg');

T = T1@T2@T3@T4@T5@T6

```
array([[ 0,  0,  1,  1.345],
       [ 0, -1,  0,  0],
       [ 1,  0,  0,  1.32],
       [ 0,  0,  0,  1]])
```

KR16L6 = models.DH.KR16L6();

T = KR16L6.fkine([0,-pi/2, 0, 0, 0, 0]). printline("rpy/xyz") #as figure

#sol = KR16L6.ikine_a(T)

```
t = 1.34, 0, 1.32; rpy/xyz = 0°, 90°, -180°
 0   0   1   1.345
 0  -1   0   0
 1   0   0   1.32
 0   0   0   1
```

R = rpy2r(0, pi/2, -pi, order="xyz")

```
>>> R = rpy2r(0, pi/2, -pi, order="xyz")
...:
array([[ 0,  0,  1],
       [ 0, -1,  0],
       [ 1,  0,  0]])
```

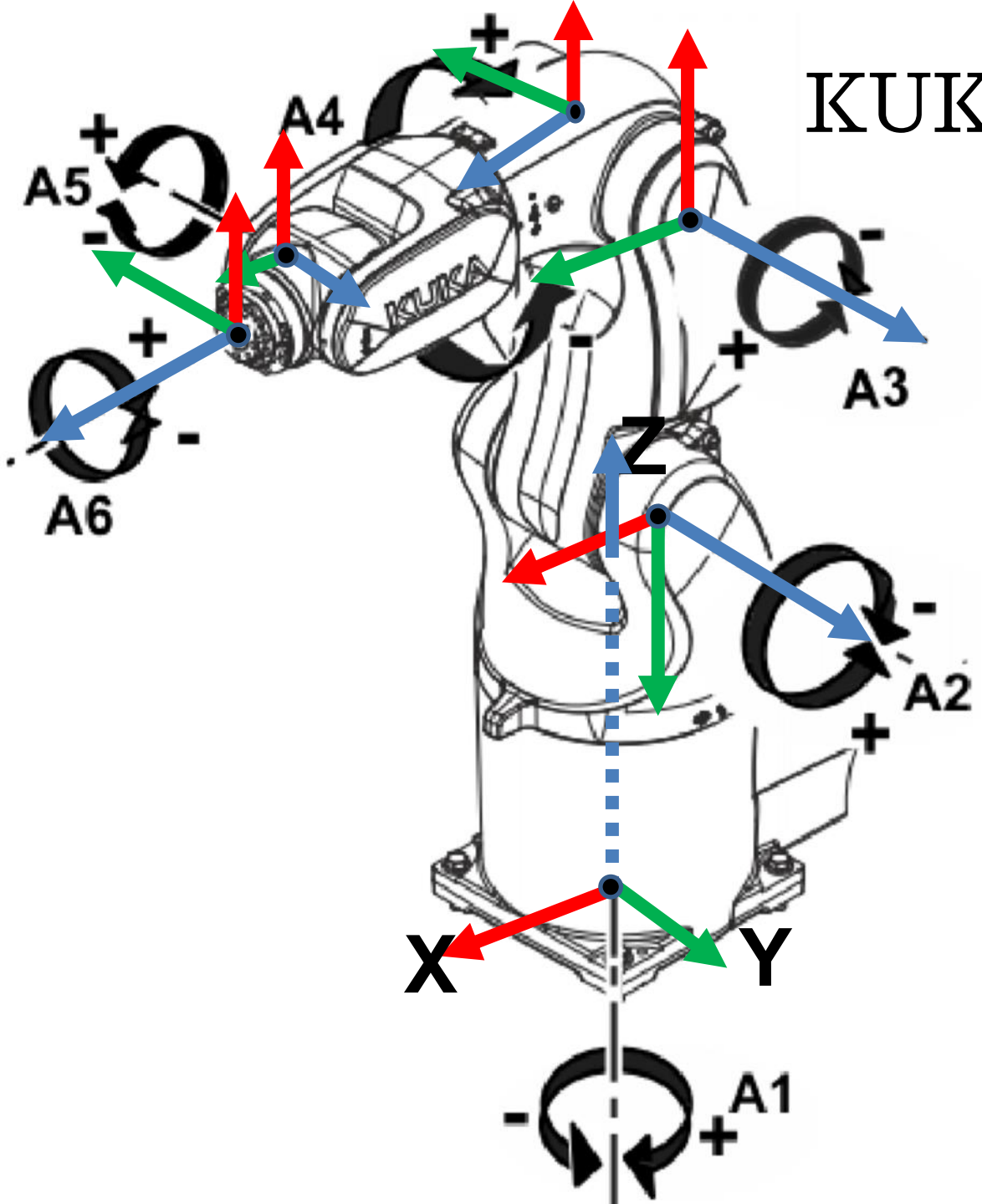
DH Parameters for KUKA KR16_L6

```
>>> KR16L6
DHRobot: KR16L6 (by KUKA), 6 joints (RRRRRR), dynamics,
geometry, standard DH parameters
```

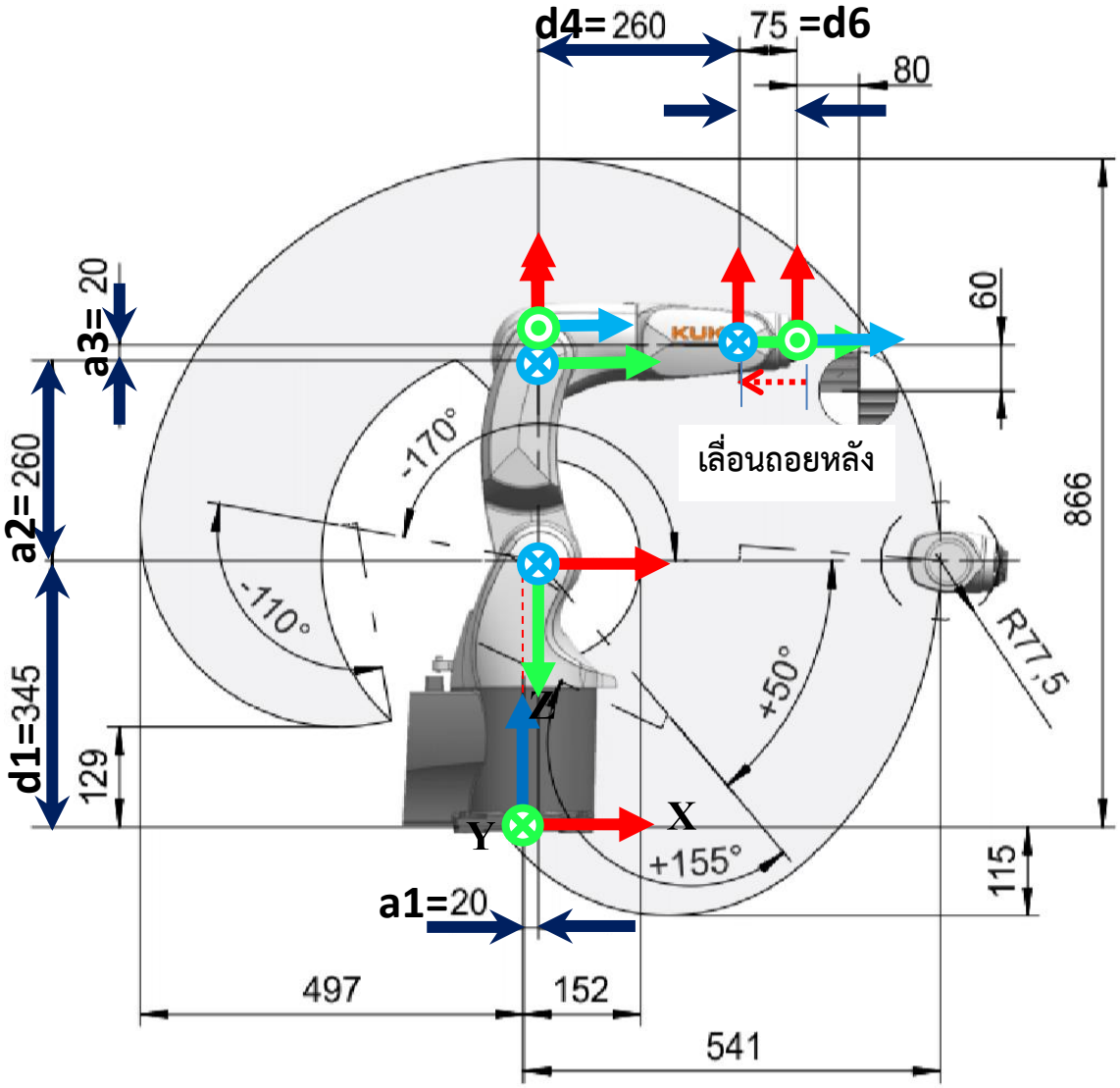
θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.675	0.26	-90.0°	-185.0°	185.0°
q2	0	0.68	0.0°	-155.0°	35.0°
q3	0	-0.035	-90.0°	-130.0°	154.0°
q4	0.97	0	90.0°	-350.0°	350.0°
q5	0	0	-90.0°	-130.0°	130.0°
q6	0.115	0	0.0°	-350.0°	350.0°

KUKA KR3

KUKA KR3



Denavit-Hartenberg Parameters



Pose in 3D:Homogeneous Transformation Matrix of KUKA KR3 (Python Version)

```
### Python code ###
A1=0; A2=-90; A3=0; A4=0; A5=0; A6=0; #as figure
#A1=0;A2=-90;A3=0;A4=0;A5=0;A6=0; #totally vertical
T1 = trotz(A1,'deg');
T2 = transl(0.020,0,0.345)@trotx(-90,'deg')@trotz(A2+90,'deg');
T3 = transl(0,-0.260,0)@trotz(-90,'deg')@trotz(A3,'deg');
T4 = transl(0.020,0,0)@trotx(-90,'deg')@trotz(A4,'deg');
T5 = transl(0,0,0.260)@trotx(90,'deg')@trotz(A5,'deg');
T6 = transl(0,0.075,0)@trotx(-90,'deg')@trotz(A6,'deg');
T = T1@T2@T3@T4@T5@T6
```

```
### Python code ###
KR3 = models.DH.KR3();
KR3.fkine([0,-pi/2,0,0,0,0]);
KR3.fkine([0,-pi/2,0,0,0,0]).printline("rpy/xyz") #as figure
#KR3.fkine([0,-pi/2,-pi/2,0,0,0]).printline("rpy/xyz") #totally vertical
```

```
t = 0.355, 0, 0.625; rpy/xyz = 0°, 90°, -180°
```

```
0     0     1     0.355
0    -1     0     0
1     0     0     0.625
0     0     0     1
```

```
array([[     0,      0,      1,  0.355],
       [     0,     -1,      0,      0],
       [     1,      0,      0,  0.625],
       [     0,      0,      0,      1]])
```

R = rpy2r(0, pi/2, -pi, order="xyz")

```
>>> R = rpy2r(0, pi/2, -pi, order="xyz")
...:
array([[     0,      0,      1],
       [     0,     -1,      0],
       [     1,      0,      0]])
```

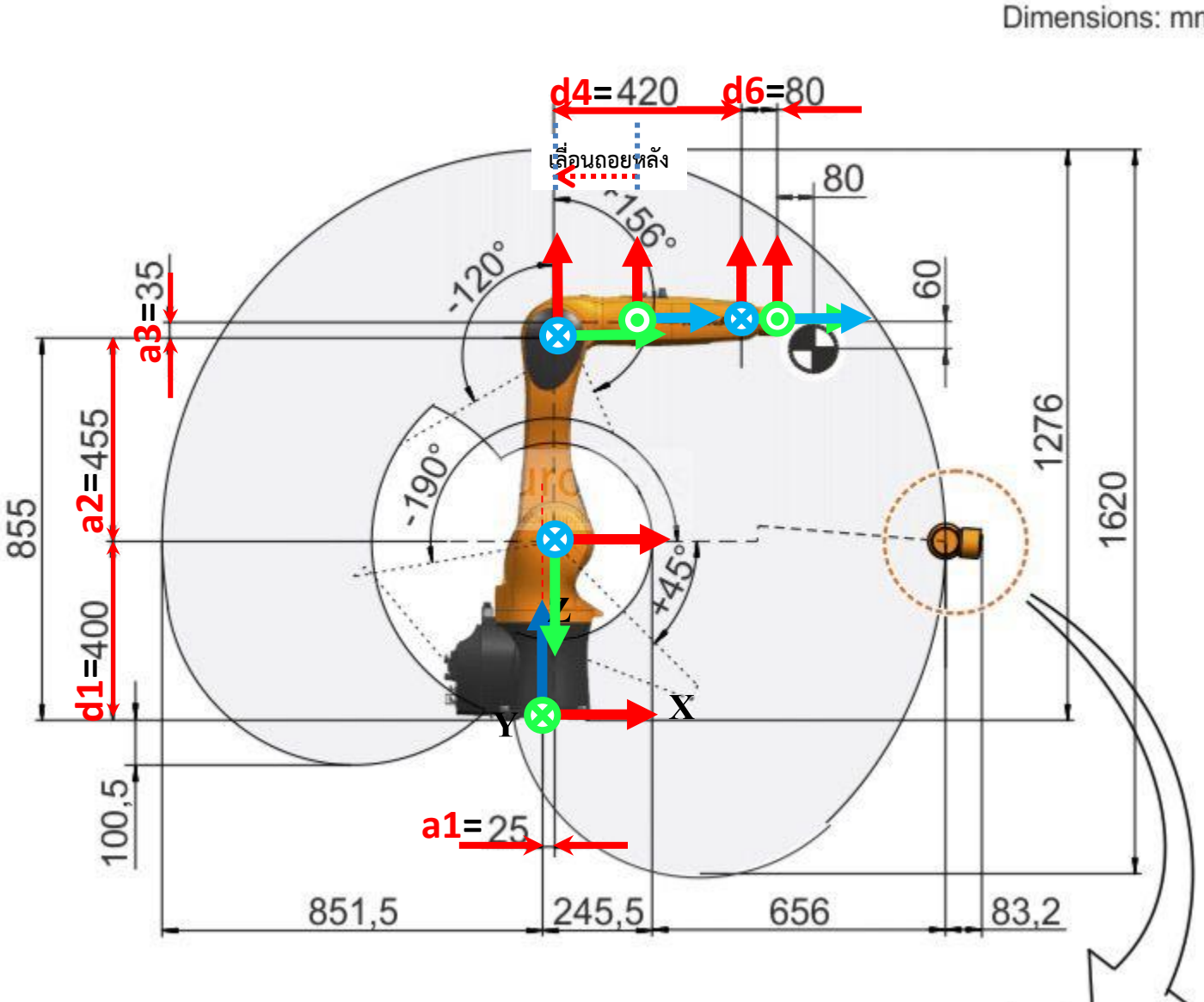
DH Parameters for KUKA KR3

```
>>> KR3
DHRobot: KR3 (by KUKA), 6 joints (RRRRRR), dynamics,
geometry, standard DH parameters
```

θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.345	0.02	-90.0°	-170.0°	170.0°
q2	0	0.26	0.0°	-170.0°	50.0°
q3	0	0.02	-90.0°	-110.0°	155.0°
q4	0.26	0	90.0°	-175.0°	175.0°
q5	0	0	-90.0°	-120.0°	120.0°
q6	0.075	0	0.0°	-350.0°	350.0°

KUKA KR6

KUKA KR6


Dimensions: mm
d4=420
d6=80
เลื่อนถอยหลัง
80
-156°
-120°
60
a3=35
a2=455
855
-190°
+45°
1276
1620
d1=400
Z
Y
X
100,5
a1=25
851,5
245,5
656
83,2

Pose in 3D:Homogeneous Transformation Matrix of KUKA KR6

```
### Python code ###

A1=0; A2=-90; A3=0; A4=0; A5=0; A6=0;

T1 = trotz(A1,'deg');

T2 = transl(0.025,0,0.4)@trotx(-90,'deg')@trotz(A2+90,'deg');

T3 = transl (0,-0.455,0)@trotz(-90,'deg')@trotz(A3,'deg');

T4 = transl(0.035,0,0)@trotx(-90,'deg')@trotz(A4,'deg');

T5 = transl(0,0,0.420)@trotx(90,'deg')@trotz(A5,'deg');

T6 = transl(0,0.080,0)@trotx(-90,'deg')@trotz(A6,'deg');

T = T1@T2@T3@T4@T5@T6
```

```
### Python code ###

KR6 = models.DH.KR6();

KR6.fkine([0,-pi/2,0,0,0,0]).printline("rpy/xyz") #as figure

#KR6.fkine([0,-pi/2,-pi/2,0,0,0]).printline("rpy/xyz") #totally vertical
```

```
t = 0.525, 0, 0.89; rpy/xyz = 0°, 90°, -180°

    0     0     1     0.525
    0    -1     0     0
    1     0     0     0.89
    0     0     0     1
```

```
array([[     0,      0,      1,  0.525],
       [     0,     -1,      0,      0],
       [     1,      0,      0,   0.89],
       [     0,      0,      0,      1]])
```

R = rpy2r(0, pi/2, -pi, order="xyz")

```
>>> R = rpy2r(0, pi/2, -pi, order="xyz")
...:
array([[     0,      0,      1],
       [     0,     -1,      0],
       [     1,      0,      0]])
```

DH Parameters for KUKA KR6

```
>>> KR6
DHRobot: KR6 (by KUKA), 6 joints (RRRRRR), dynamics,
 geometry, standard DH parameters
```

θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.4	0.025	-90.0°	-170.0°	170.0°
q2	0	0.455	0.0°	-190.0°	45.0°
q3	0	0.035	-90.0°	-120.0°	156.0°
q4	0.42	0	90.0°	-185.0°	185.0°
q5	0	0	-90.0°	-120.0°	120.0°
q6	0.08	0	0.0°	-350.0°	350.0°

Pose in 3D:Homogeneous Transformation Matrix of KUKA KR6

%ZYX=$R_z(\theta_y)\ R_y(\theta_p)\ R_x(\theta_r)$

A=-116.09; % yaw

B=90; % pitch

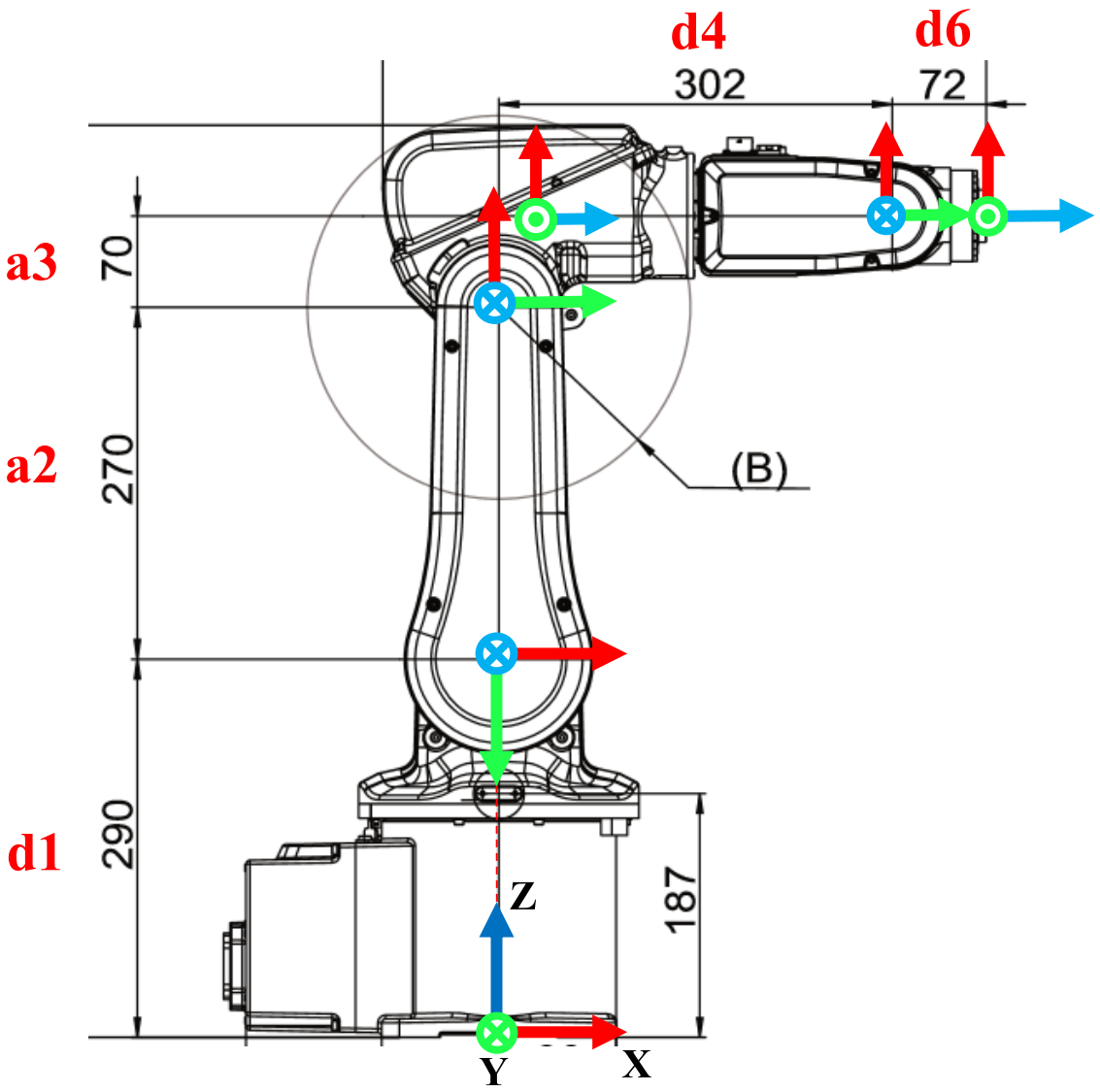
C=-116.09; % roll

RPY1 = rpy2tr(C, B, A ,'deg') %

RPY2 = rotz(A ,'deg') * roty(B ,'deg') * rotx(C ,'deg')

ABB IRB120

ABB IRB120



Pose in 3D:Homogeneous Transformation Matrix of ABB IRB120

Python code

A1=0; A2=-90; A3=0; A4=0; A5=0; A6=0;

```
T1 = trotz(A1,'deg');
T2 = transl(0,0,0.29)@trotx(-90,'deg')@trotz(A2+90,'deg');
T3 = transl(0,-0.27,0)@trotz(-90,'deg')@trotz(A3,'deg');
T4 = transl(0.07,0,0)@trotx(-90,'deg')@trotz(A4,'deg');
T5 = transl(0,0,0.302)@trotx(90,'deg')@trotz(A5,'deg');
T6 = transl(0,0.072,0)@trotx(-90,'deg')@trotz(A6,'deg');
T = T1@T2@T3@T4@T5@T6
```

```
array([[ 0,  0,  1,  0.374],
       [ 0, -1,  0,      0],
       [ 1,  0,  0,   0.63],
       [ 0,  0,  0,      1]])
```

Python code

```
IRB120 = models.DH.IRB120();
IRB120.fkine([0,-pi/2,0,0,0,0]).printline("rpy/xyz") #as figure
#IRB120.fkine([0,-pi/2,-pi/2,0,0,0]).printline("rpy/xyz") #totally vertical
```

```
t = 0.374, 0, 0.63; rpy/xyz = 0°, 90°, -180°
```

0	0	1	0.374
0	−1	0	0
1	0	0	0.63
0	0	0	1

R = rpy2r(0, pi/2, -pi, order="xyz")

```
>>> R = rpy2r(0, pi/2, -pi, order="xyz")
...:
array([[ 0,  0,  1],
       [ 0, -1,  0],
       [ 1,  0,  0]])
```

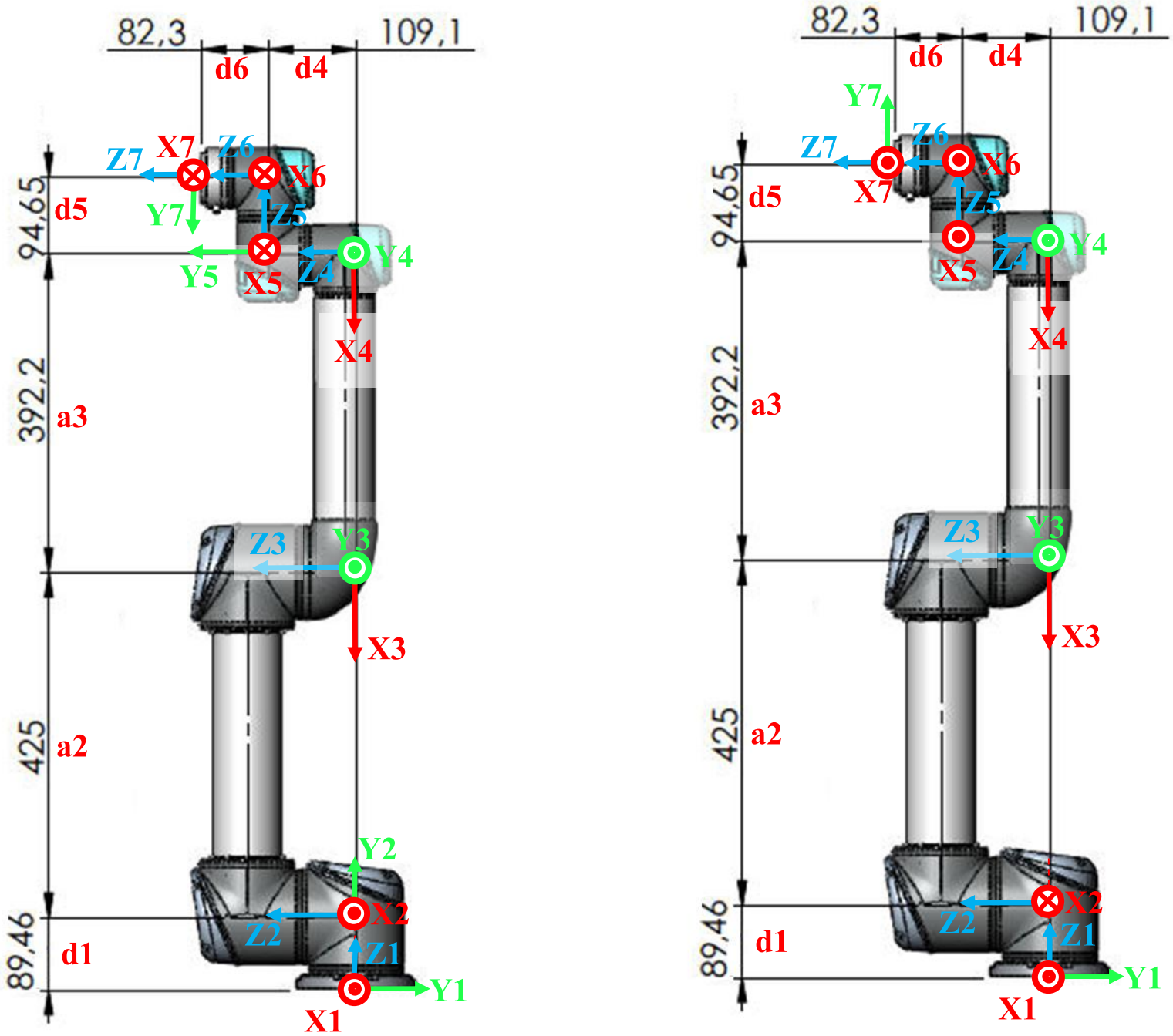
Ref: D:\Study\Study\KMITL_Fundamentals_of_Industrial_Robot\RVC3-python-main\RVC3-python-main\robotics-toolbox-python\roboticstoolbox\models\DH\IRB120

DH Parameters for ABB IRB120

```
>>> IRB120
DHRobot: IRB 120 (by ABB), 6 joints (RRRRRR), dynamics,
 geometry, standard DH parameters
```

θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.29	0	-90.0°	-180.0°	180.0°
q2	0	0.27	0.0°	-100.0°	100.0°
q3	0	0.07	-90.0°	-220.0°	60.0°
q4	0.302	0	90.0°	-200.0°	200.0°
q5	0	0	-90.0°	-120.0°	120.0°
q6	0.072	0	0.0°	-400.0°	400.0°

Universal Robots UR5


82,3
109,1
d6
d4
94,65
d5
392,2
a3
425
a2
89,46
d1
X7
Z7
Z6
X6
Y7
Z5
Y5
X5
Z4
Y4
X4
Z3
Y3
X3
Y2
X2
Z2
Z1
Y1
X1

Pose in 3D:Homogeneous Transformation Matrix of UR5

```
### Python code ###

A1=0; A2=-90; A3=0; A4=-90; A5=0; A6=0;

T1 = trotz(A1,'deg');

T2 = transl(0,0,0.089459)@trotx(90,'deg')@trotz(A2+90,'deg');

T3 = transl(0,0.425,0)@trotz(-90,'deg')@trotz(A3,'deg');

T4 = transl(-0.39225,0,0)@trotz(A4+90,'deg');

#T5 = transl(0,0,0.10915)@troty(-90,'deg')@trotz(-90,'deg')@ trotz(A5+180,'deg');

T5 = transl(0,0,0.10915)@trotx(90,'deg')@troty(-90,'deg')@ trotz(A5+180,'deg');

T6 = transl(0,0,0.09465)@trotx(90,'deg')@trotz(A6,'deg')@transl(0,0,0.0823);

T = T1@T2@T3@T4@T5@T6
```

```
array([[ 1,  0,  0,  0],
       [ 0,  0, -1, -0.1914],
       [ 0,  1,  0,  1.001],
       [ 0,  0,  0,  1]])
```

```
### Python code ###

UR5 = models.DH.UR5();

UR5.fkine([0,-pi/2,0,-pi/2,0,0]).printline("rpy/xyz") #as figure
```

```
t = 0, -0.191, 1; rpy/xyz = -180°, 0°, 90°

-1   0   0   0
 0   0  -1  -0.1914
 0  -1   0   1.001
 0   0   0   1
```

R = rpy2r(-pi, 0, pi/2, order="xyz")

```
>>> R = rpy2r(-pi, 0, pi/2, order="xyz")
...:
array([[-1,  0,  0],
       [ 0,  0, -1],
       [ 0, -1,  0]])
```

Ref: D:\Study\Study\KMITL_Fundamentals_of_Industrial_Robot\RVC3-python-main\RVC3-python-main\robotics-toolbox-python\roboticstoolbox\models\DH\UR5

Pose in 3D:Homogeneous Transformation Matrix of UR5

Python code

A1=0; A2=-90; A3=0; A4=-90; A5=0; A6=0;

```
T1 = trotz(A1,'deg');
T2 = transl(0,0,0.089459) @trotx(90,'deg')@trotz(180,'deg')@trotz(A2+90,'deg');
T3 = transl(0,-0.425,0)@trotz(90,'deg')@trotz(A3,'deg');
T4 = transl(-0.39225,0,0)@trotz(A4+90,'deg');
T5 = transl(0,0,0.10915)@trotx(-90,'deg')@troty(-90,'deg')@ trotz(A5,'deg');
T6 = transl(0,0,0.09465)@trotx(90,'deg')@trotz(A6,'deg')@transl(0,0,0.0823);
T = T1@T2@T3@T4@T5@T6
```

```
array([[ 1,  0,  0,  0],
       [ 0,  0, -1, -0.1914],
       [ 0,  1,  0,  1.001],
       [ 0,  0,  0,  1]])
```

Python code

```
UR5 = models.DH.UR5();
UR5.fkine([0,-pi/2,0,-pi/2,0,0]).printline("rpy/xyz") #as figure
```

```
t = 0, -0.191, 1; rpy/xyz = -180°, 0°, 90°
```

```
-1   0   0   0
 0   0  -1  -0.1914
 0  -1   0   1.001
 0   0   0   1
```

R = rpy2r(-pi, 0, pi/2, order="xyz")

```
>>> R = rpy2r(-pi, 0, pi/2, order="xyz")
...:
array([[-1,  0,  0],
       [ 0,  0, -1],
       [ 0, -1,  0]])
```

Ref: D:\Study\Study\KMITL_Fundamentals_of_Industrial_Robot\RVC3-python-main\RVC3-python-main\robotics-toolbox-python\roboticstoolbox\models\DH\UR5

DH Parameters for UR5

```
>>> UR5
DHRobot: UR5 (by Universal Robotics),
6 joints (RRRRRR), dynamics,
standard DH parameters
```

θ_j	d_j	a_j	α_j
q1	0.08946	0	90.0°
q2	0	-0.425	0.0°
q3	0	-0.3922	0.0°
q4	0.1091	0	90.0°
q5	0.09465	0	-90.0°
q6	0.0823	0	0.0°

PUMA560

Denavit-Hartenberg Parameters

```
puma = models.DH.Puma560();

T = puma.fkine(puma.qn);

T.printline()

sol = puma.ikine_a(T)
```

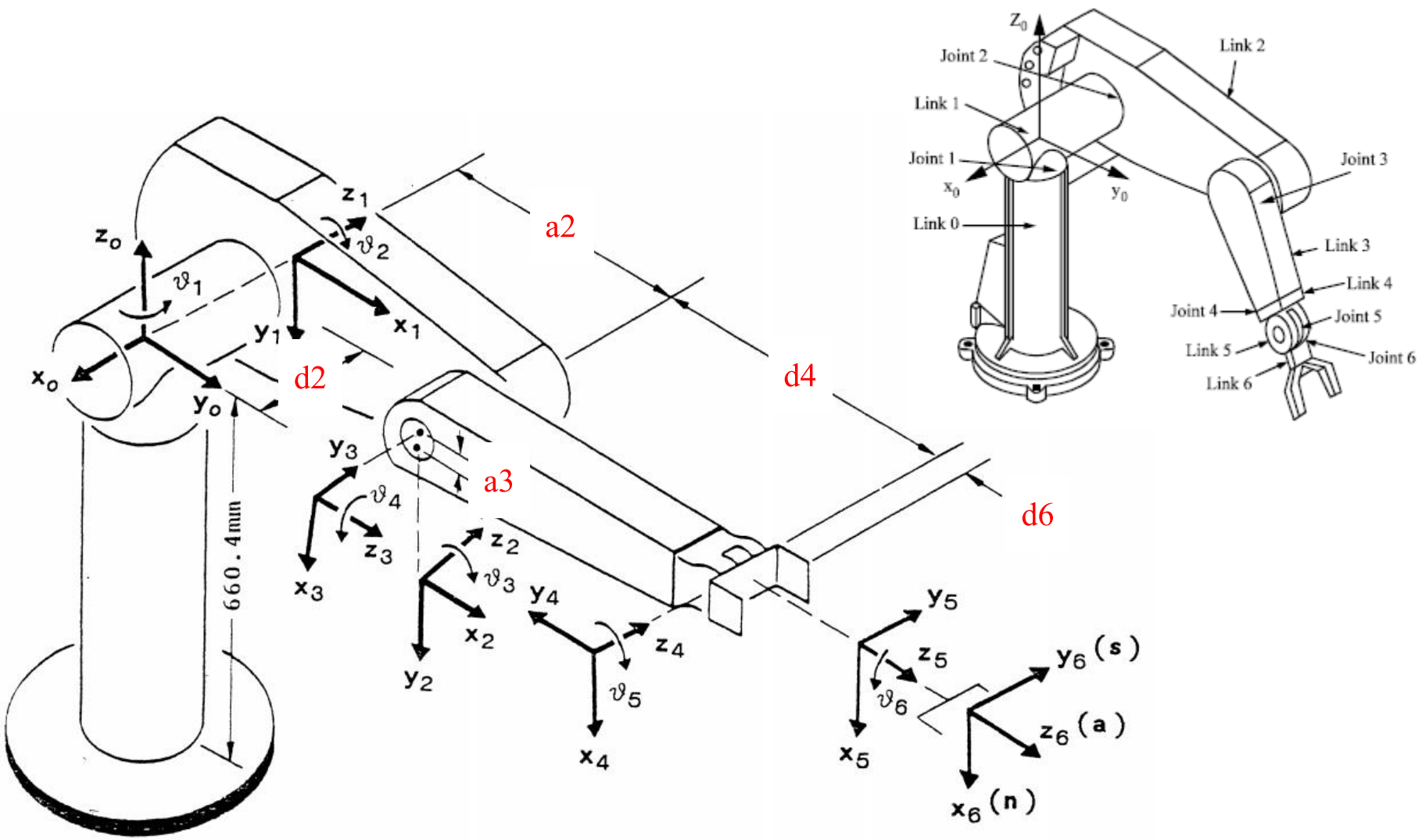
Denavit-Hartenberg parameters for Puma

Unimate Puma 500
Oussama Khatib | Used with permission

θ_j	d_j	a_j	α_j
q1	0.0000	0.0000	$\pi/2$
q2	0.0000	0.4318	0
q3	0.1500	0.0203	$-\pi/2$
q4	0.4318	0.0000	$\pi/2$
q5	0.0000	0.0000	$-\pi/2$
q6	0.0000	0.0000	0

https://robotacademy.net.au/

Denavit-Hartenberg Parameters



DH Parameters for Puma560

```
>>> puma = models.DH.Puma560();
...:
DHRobot: Puma 560 (by Unimation), 6 joints (RRRRRR),
dynamics, geometry, standard DH parameters
```

θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.6718	0	90.0°	-160.0°	160.0°
q2	0	0.4318	0.0°	-110.0°	110.0°
q3	0.15	0.0203	-90.0°	-135.0°	135.0°
q4	0.4318	0	90.0°	-266.0°	266.0°
q5	0	0	-90.0°	-100.0°	100.0°
q6	0	0	0.0°	-266.0°	266.0°