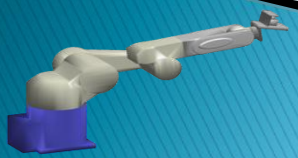
Week 3

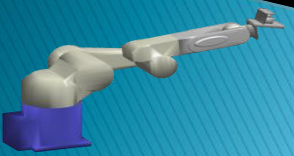
<u>Schedule</u>

- Quiz 1:
  - Individual (30 mins)
  - Team (20 mins)
- Review & discussion of Lab 2 (5 mins)
- Introduction to Lab 3 (5 mins)
- Collaboratively work on Lab 3 (≈2 hours)
- Prework for Lab 4 (5 mins)



Overview of Week 3: Quiz 1

- Two quizzes worth 5% in total
- Individual quiz
  - worth 4%
  - will go from lab start time for 30 minutes
  - 10 questions (1 question from each category)
  - No talking
- Group quiz
  - worth 1%
  - will start immediately after the individual quiz
  - will go for 20 minutes
  - Groups of 3 people or less
  - 20 questions (2 question from each category)
  - Lots of talking within group

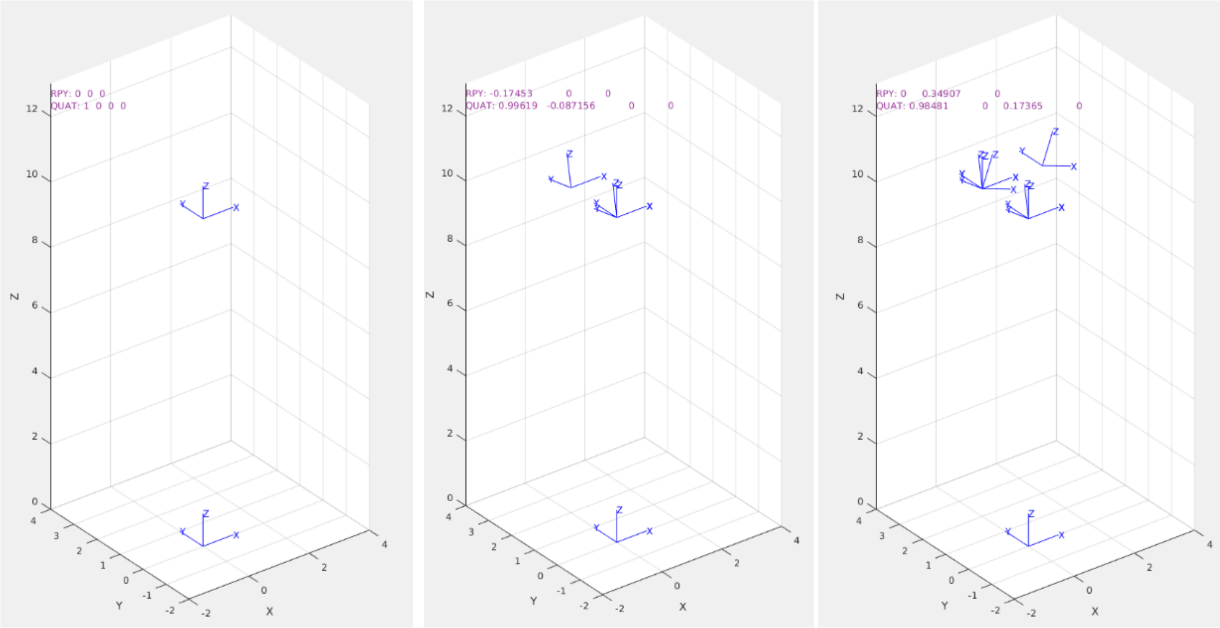


Lab 2 Review – Q1

1. Consider an UAV flying to monitor cattle, and plot a 3D transform
2. Plot the frame moving by specifying translations or a combination of translations and rotations
3. Encode the steps in a for-loop

```
uavTR{2} = transl([0,0,10]);
uavTR{3} = transl([0,0,10]) * trotx(-10 * pi/180);
uavTR{4} = transl([0,2,10]) * trotx(-10 * pi/180);
uavTR{5} = transl([0,2,10]);
uavTR{6} = transl([0,2,10]) * troty(20 * pi/180);
uavTR{7} = transl([2,2,10]) * troty(20 * pi/180);
uavTR{8} = transl([2,2,10]);
uavTR{9} = transl([2,2,0]);
```

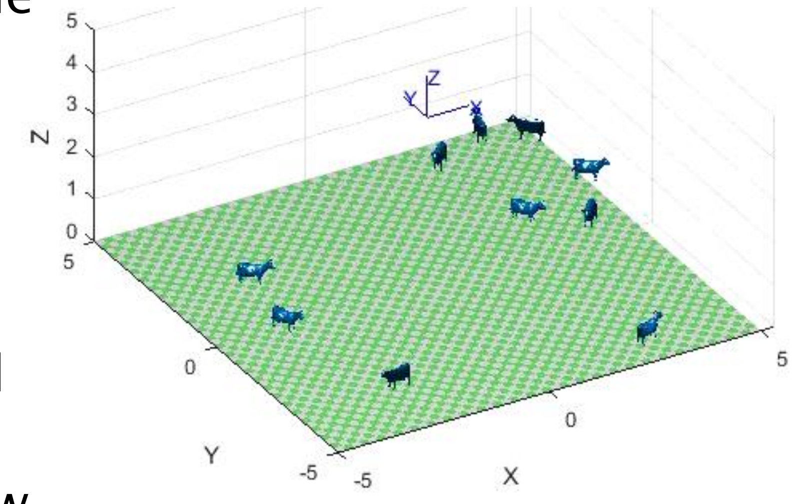
```
waypoints(:,:,1) = transl(0,0,0);
waypoints(:,:,2) = transl(0,0,10);
waypoints(:,:,3) = waypoints(:,:,2)*trotx(deg2rad(-10));
waypoints(:,:,4) = transl(0,2,10)*trotx(deg2rad(-10));
waypoints(:,:,5) = transl(0,2,10);
waypoints(:,:,6) = waypoints(:,:,5)*troty(deg2rad(20));
waypoints(:,:,7) = transl(2,2,10)*troty(deg2rad(20));
waypoints(:,:,8) = transl(2,2,10);
waypoints(:,:,9) = transl(2,2,0);
```



Lab 2 Review – Q2 & Q3

1. Plot a cow herd and make the cows move
   i. Spawn varying number of cows
   ii. Make the cows randomly move one step
   iii. Obtain the location of each cow relative to the world frame

   ```
   cowHerd.cow{2}.base
   ```



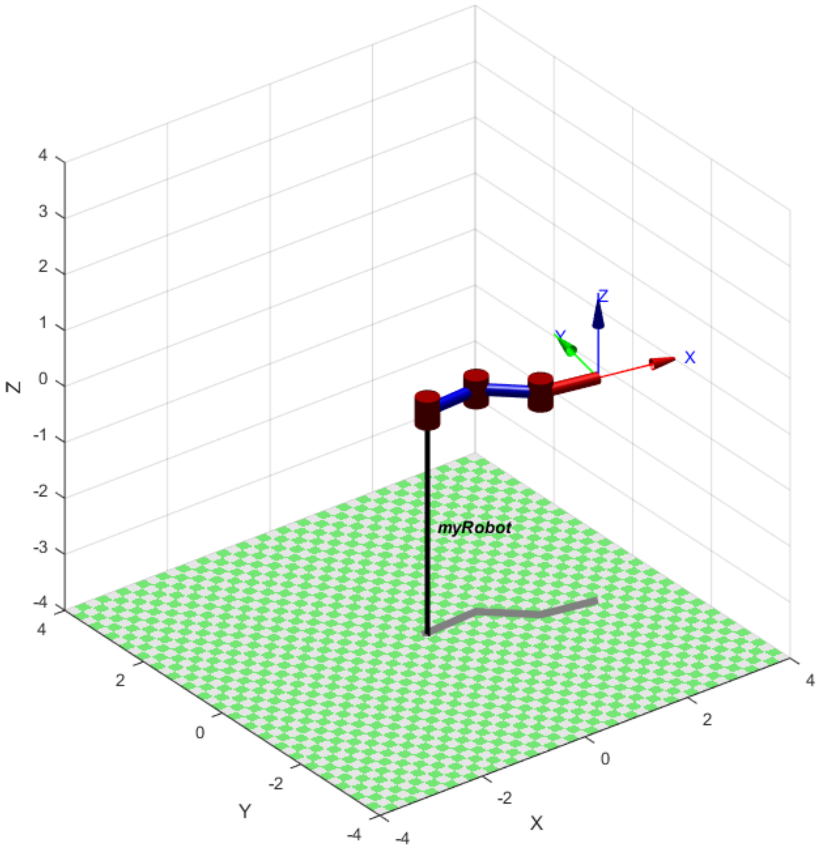
2. Plot the cows and the UAV (a coordinate frame moving in 3D), obtain transform between the UAV and each cow
   i. How do you calculate the transform between the UAV and cow?
   ii. Use “display” to print information to the command window
3. Plot and track a cow
   i. If you know the transform of the cow, how do you calculate the transform of the UAV monitoring the cow from 5 metres above?

Lab 2 Review – Q4

1. Plot and move a robot after determining the DH parameters
2. Learn to use the teach mode
3. Obtain the current joint angles displayed in the teach mode
4. Show the joint limits of the robot in the command window

```
% 4.1 and 4.2: Define the DH Parameters to create the Kinematic model
L1 = Link('d',0,'a',1,'alpha',0,'qlim',[-pi pi])
L2 = Link('d',0,'a',1,'alpha',0,'qlim',[-pi pi])
L3 = Link('d',0,'a',1,'alpha',0,'qlim',[-pi pi])

robot = SerialLink([L1 L2 L3],'name','myRobot')                    % Generate the model
```

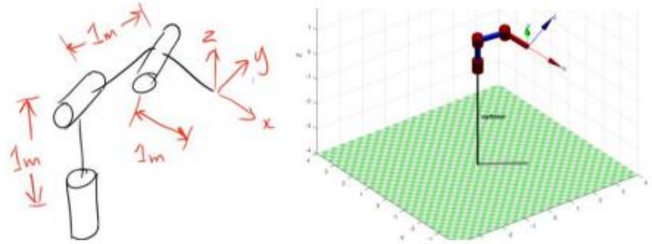


Introducing Lab 3

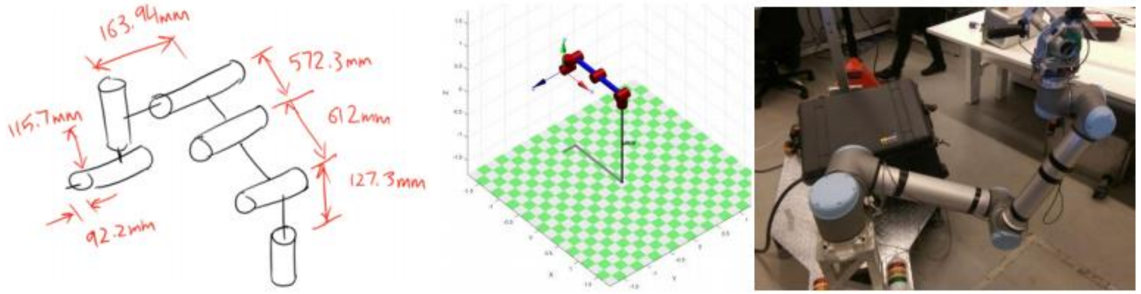
1. Derive DH parameters for each manipulator
   i. Be familiar with the variables: θ, d, a, α
   ii. Use the teach function to observe how each of the joints move, is it what you expect?
   iii. Obtain the current joint angles
   iv. Determine the forward transformation matrix
   v. Given some transformation T, determine the joint angles required to satisfy T
   vi. What does the Jacobian tell you about the end-effector velocity given some joint configuration?

ROBOT 1: 3-Link 3D Robot

Note the orientation of the end-effector frame.

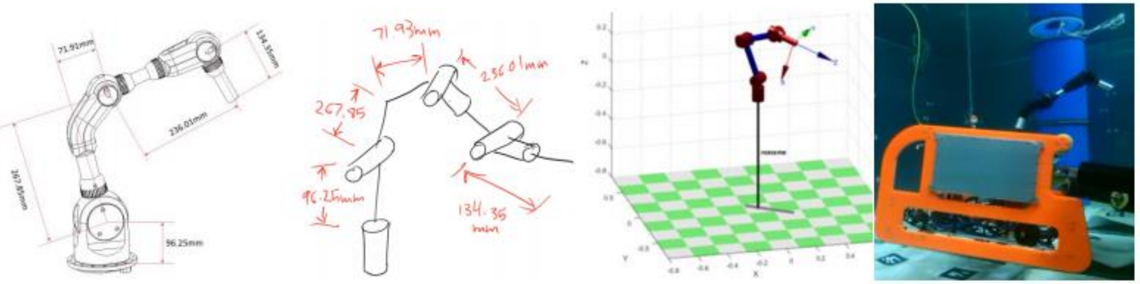


ROBOT 2: AAN-BOT (Assistance As Needed BOT) (6DOF)[1][2]



ROBOT 3: Submerged Pile Inspection Robot (SPIR)[3]

Note: this arm has 2DOF joints.



Introducing Lab 3

1. Derive DH parameters for the Denso VM-6083D-W
2. Sample the joint angles of the manipulator then determine the forward transformation matrix at each joint configuration
3. Using the list of XYZ locations obtained from the forward transformation matrices plot the workspace of the manipulator

2 Consider a Denso VM-6083D-W industrial robot shown in the PDF available on UTSOnline[4]

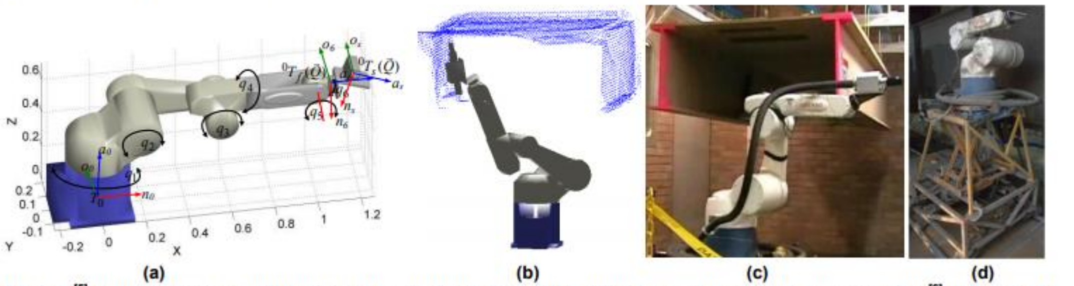


Fig 1. (a)[5] Simulated robot model: joints and coordinate frames (i.e. robot base, end-effector and sensor), (b)[6] Blasting of a simulated environment constructed from exploration data, (c)[5] (d[7]) Grit-blasting system: a 6DOF manipulator on 2DOF base.

2.1 Download the PDF of the robot from UTSOnline

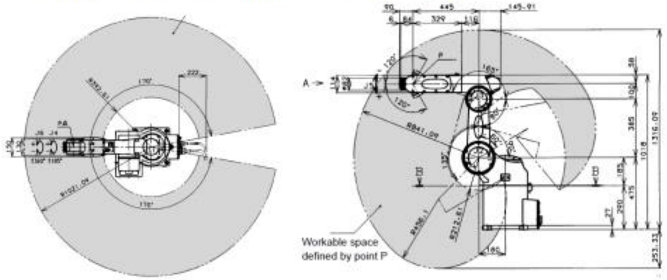
2.2 Determine the D&H parameters based upon the link measurements on the PDF

2.3 Determine and include the joint limits in your model

2.4 Sample the joint angles within the joint limits at 30 degree increments between each of the joint limits (note: from *jacob0* or *teach* you see that joint 6 is irrelevant to the position so don't iterate through it).

2.5 Use fkine to determine the point in space for each of these poses, so that you end up with a big list of points

2.6 Create a 3D model showing where the end effector can be over all these samples. (hint: it should look similar to the top and side workspace view on the PDF)



2.7 (Bonus) Get the start of the blast using fkine (to get end effector transform)

2.8 (Bonus) Get the end of the stream with TR * transl (blast stream length (i.e. 1m along z)

2.9 (Bonus) Project a line out of the end effector (i.e. a mock grit-blasting stream)

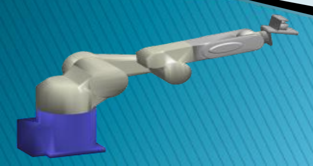
2.10 (Bonus) Create a surface plane that goes through [1.5,0,0] with a normal [-1,0,0]

```
[Y,Z] = meshgrid(-2:0.1:2,-2:0.1:2);
X = repmat(1.5,size(Y,1),size(Y,2));
surf(X,Y,Z);
```

2.11 (Bonus) Determine if and where the blast stream intersects with the surface plane

Pre-work before Week 4 lab

- Completed Lab 3 exercises (solution available after class)
- Review “Week 3 – D&H Reference Guide.pdf” on UTSOnline
- Ensure you have achieved 80% in Quiz 1
- Read textbook
  - Section 7.3 (pages 146-152): “Inverse Kinematics”
  - Section 7.4 (pages 152-158): “Trajectories”
- Watch video lectures on
  - Inverse Kinematics https://www.youtube.com/watch?v=3ZcYSKVDlOc
  - Velocity, Force and the Jacobian https://www.youtube.com/watch?v=g_Nl4iPd1jI
- Attempt Lab 4 exercises
- Work on Lab Assignment #1 (due week 6)



Note/slides from the textbook

- Textbook readings for week 3:
  - Section 7.2 (pages 141-146): “Forward Kinematics”

- Although it is better to read the textbook, some notes (in slide format) have been summarised below

Forward Kinematics

- The forward kinematics is often expressed in functional form $\xi_E = K(q)$
- Using homogeneous transformations this is simply the product of the individual link transformation matrices which for an N-axis manipulator is:

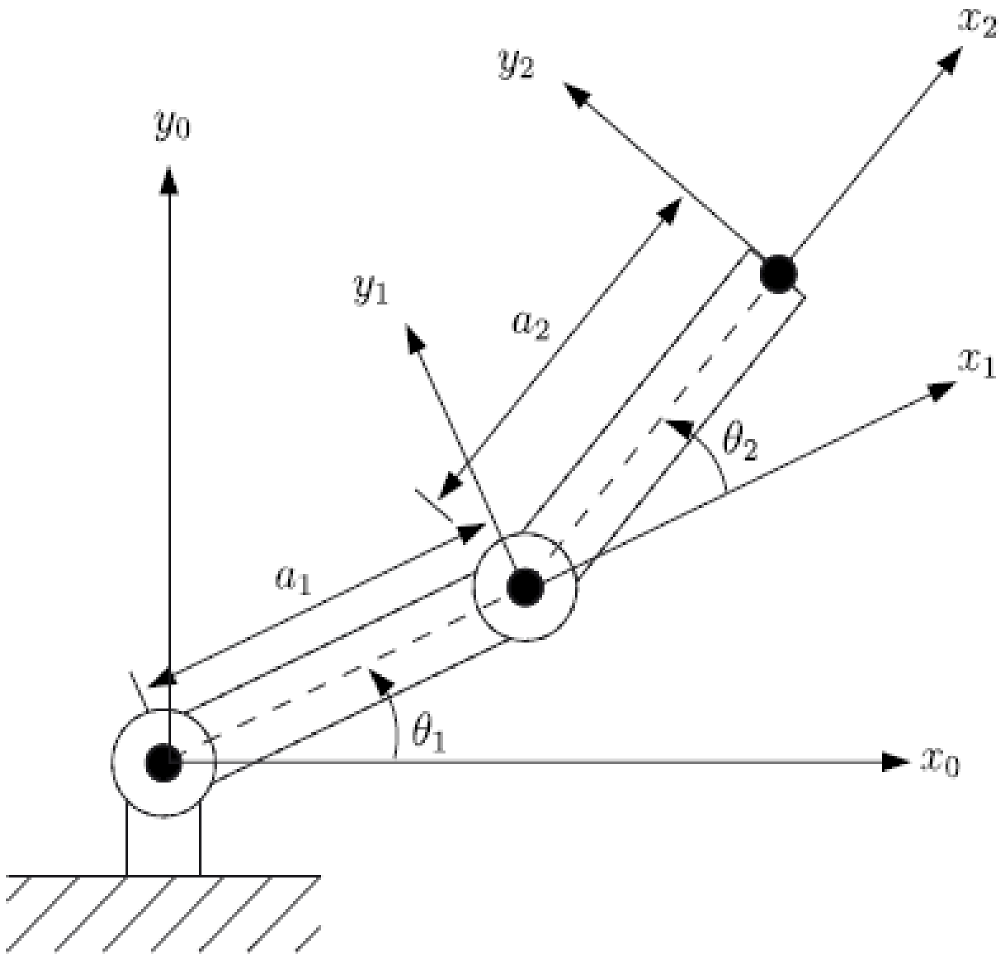
$$\xi_E \sim 0_{T_E} = 0_{A_1} 0_{A_2} \ldots N-1_{T_E}$$

A 2-Link Robot

Link	θ_i	d_i	a_i	α_i	σ_i
1	q_1	0	1	0	0
2	q_2	0	1	0	0

A 2-Link Robot (continued...)

Fig. 7.3. Two-link robot



A 2-Link Robot (continued…)

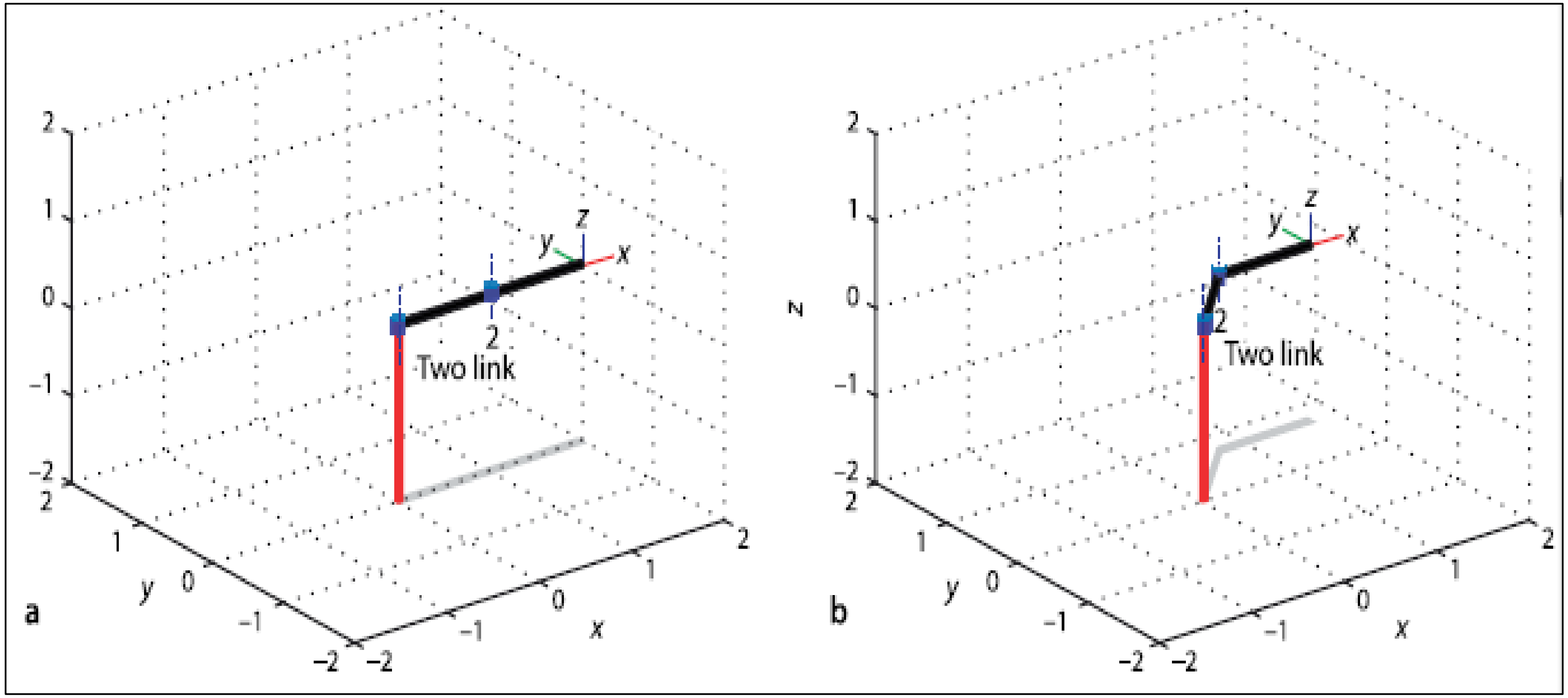
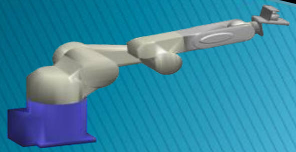


Fig. 7.4. The two-link robot in two different poses, a the pose (0, 0); b the pose $(\frac{\pi}{4}, -\frac{\pi}{4})$ Note the graphical details. Revolute joints are indicated by cylinders and the joint axes are shown as line segments. The final-link coordinate frame and a shadow on the ground are also shown.

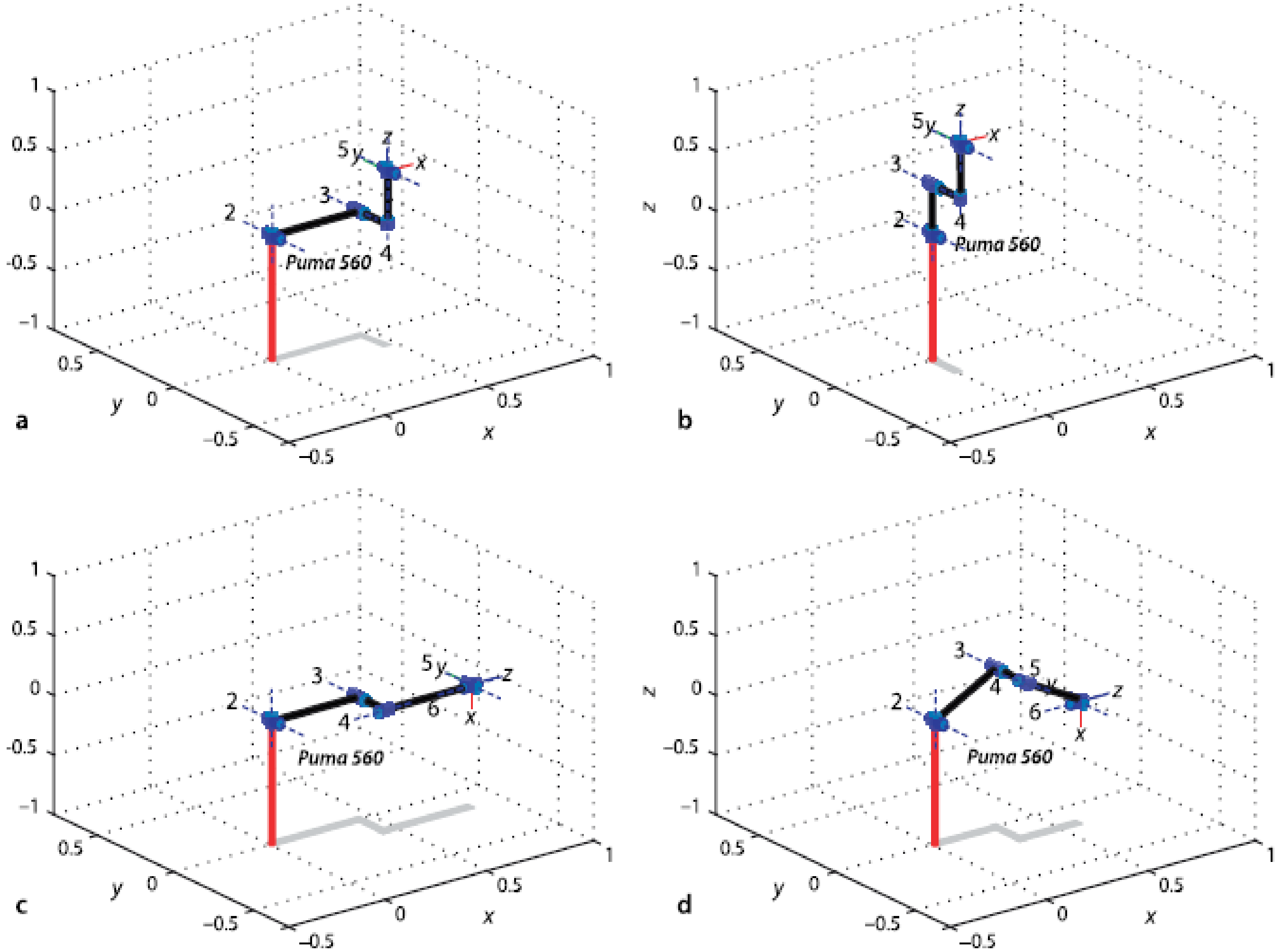
A 6-Axis Robot

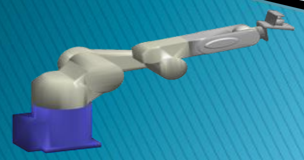
- Truly useful robots have a task space $\mathcal{T} \subset SE(3)$ enabling arbitrary position and attitude of the end-effector.
- The task space has six spatial degrees of freedom: three translational and three rotational.



A 6-Axis Robot

Fig. 7.5. The Puma robot in 4 different poses.
a *Zero angle*;
b *ready* pose;
c *stretch*;
d *nominal*





Reference

- Spong MW, Hutchinson S, Vidyasagar M (2006) Robot modeling and control.