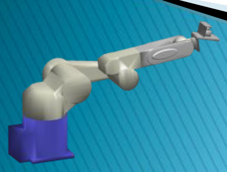
Week 4

Schedule

- Review & discussion of Lab 3 (5 mins)
- Discussion of Lab Assignment 1 (5 mins)
- Introduction to Lab 4 (5 mins)
- Collaboratively work on Lab 4 (≈2 hours)
- Overview of Quiz 2 (5 mins)
- Prework for Lab 5 (5 mins)
- Breakout rooms – quiz groups (10-? mins)
- Work on exercises, tutors available to call

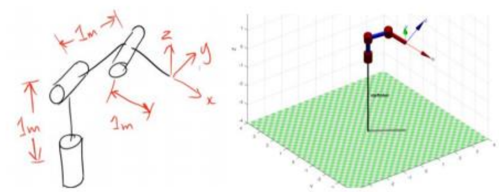


Lab 3 Review – Q1

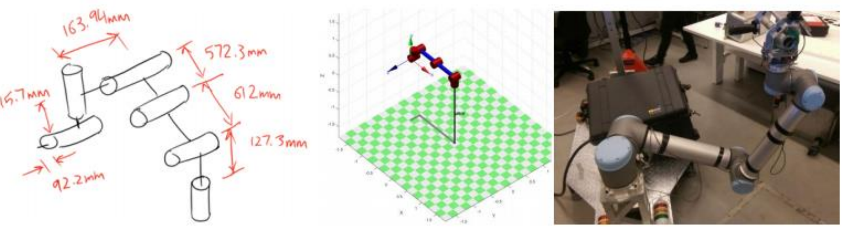
1. Derive DH parameters for each manipulator
   i. Be familiar with the variables: θ, d, a, α
   ii. Use the teach function, is it what you expect?
   iii. Obtain the current joint angles
   iv. Determine the forward kinematics transformation matrix
   v. Given some transformation T, determine the joint 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.

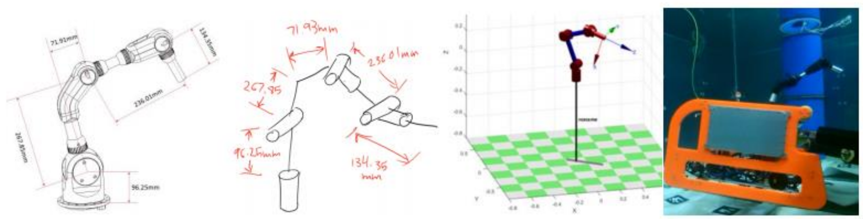


ROBOT 2: AAN-BOT (Assistance As Needed BOT) (6DOF)[1][2]



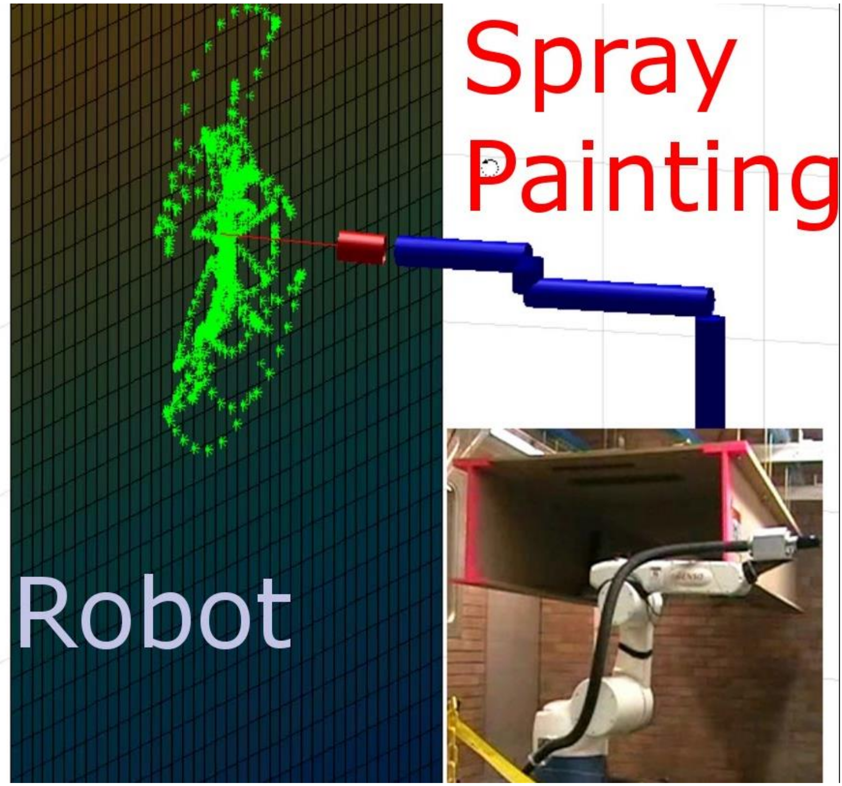
ROBOT 3: Submerged Pile Inspection Robot (SPIR)[3]

Note: this arm has 2DOF joints.



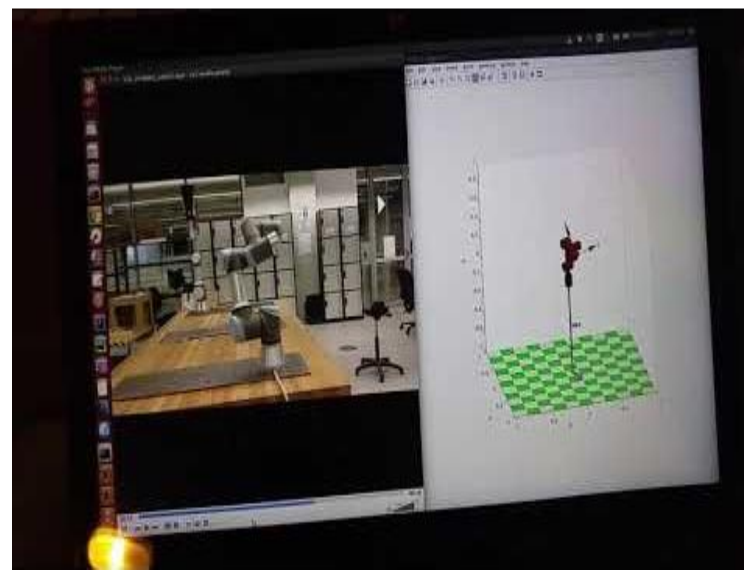
Lab 3 Review – Q2

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



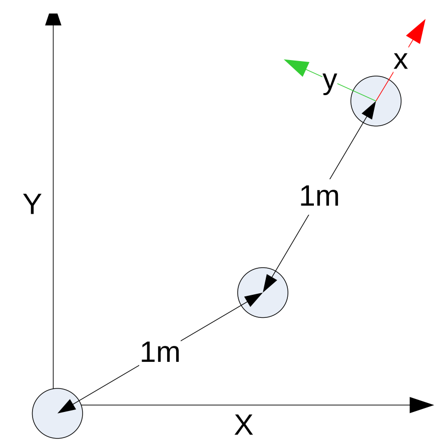
Lab Assignment 1

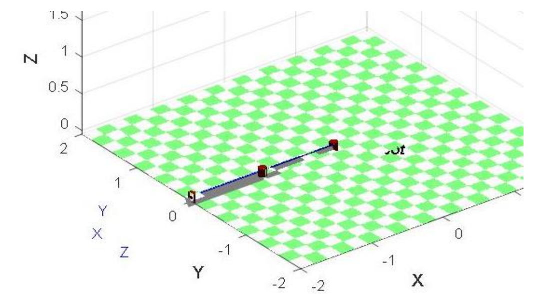
- Ensure you have done your Mechatronics Lab induction (via UTSOnline)
- DM via Teams to Sheila and me
  - Completed and signed Risk Assessment
  - Completed and signed Safe Work Method Statement (SWMS)
  - video evidence of a correct robot model to We'll send you the UR3 instructions



Introduction to Lab 4 – Q1

- Look at Lab 4 – Trajectory Interpolation Reference Guide.pdf
- Draw with 3DOF Planar Arm so the Z axis faces down ways
- Move robot with “teach”. Observe how there’s no way to affect the Z, roll or pitch values, no matter what joint value you choose.
- Use ikine and mask out the impossible-to-alter values
- Go through a loop to draw a “straight” line and observe its straightness
- Bonus: trace out a circle using ikine



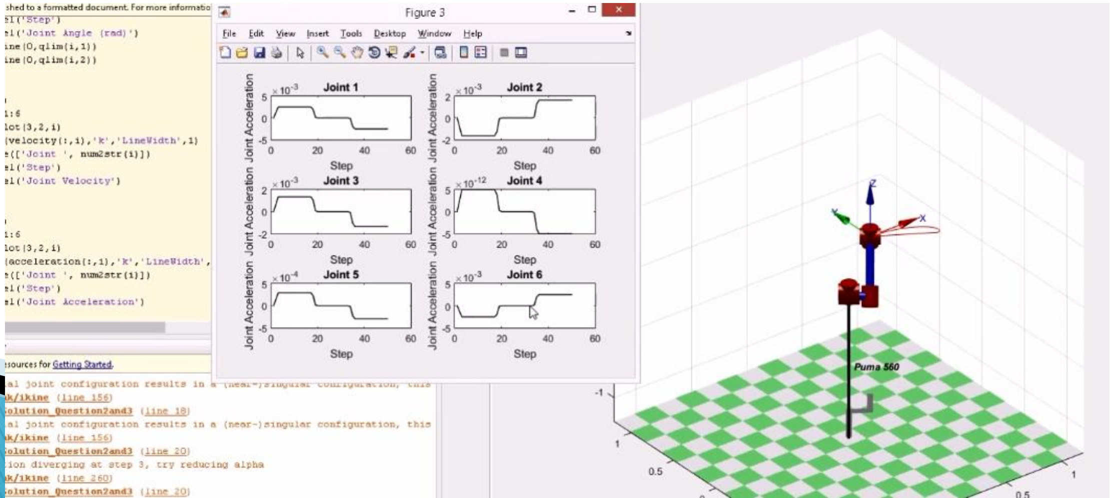


Introduction to Lab 4 – Q2

- Inverse Kinematics to Determine Joint Angles of Puma 560
- Solve inverse kinematics to get the required joint angles
- This is similar to the assignment question

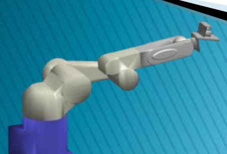
Introduction to Lab 4 – Q3

- Interpolation: Get from A to B using puma560
- Download Trajectory Interpolation Reference Guide
- Generate a matrix of interpolated joint angles with 50 steps between q1 and q2. Do steps 3.4 to 3.10 for each of the two methods:
  - Quintic Polynomial Profile
  - Trapezoidal Velocity Profile



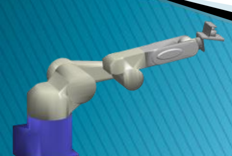
Overview of Week 5: Quiz 2

- In total worth 5%
- 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
  - 20 questions (2 question from each category)
  - Lots of talking



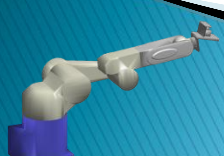
Question Categories (10)

- Baxter
  - Distance between arms in certain joint states
  - Distance between bases and/or arms in certain joint states
  - Straightforward Inverse kinematics using robotics toolbox
- Puma 560
  - Distance from end effector to surface in environment
  - Distance from end effector to surface in environment given new information about base location
  - Point in the end effector's coordinate frame
  - Straightforward Jacobian analysis
  - Straightforward Inverse kinematics using robotics toolbox
- Self collisions for a Hyper Redundant 2D arm
- D&H Transforms and forward kinematics in Matlab



…but how can I prepare?

- Finish Onramp & get 80% in Quiz 1 (thanks to those who have)
- Complete Expected Prework for Labs 1-3 & attempt all questions in Lab 4 Exercises
- Be familiar with 2D/3D transforms
- Use the Matlab Robotics Toolbox to Create models
  - Baxter (*mdl_baxter*), puma560 (*mdl_puma560*) and Hyper Rendunant 2d (*mdl_hyper2d*)
- Competently use the Matlab Robotics Toolbox for
  - fkine, ikine, qlim, plot, animate, teach, jacob0, etc
- Read the prescribed textbook sections in Chapters 1,2, and 7. Particularly focus on Chapters 7
- Watch the recommended videos



Pre-work before Week 5 lab

- Completed week 4 lab exercises
- Review "Week 4 - Trajectory Interpolation Reference Guide.pdf"
- Ensure you have achieved 80% in Quiz 1
- Read textbook
  - Section 3.2 (pages 51–56): "Time Varying Coordinate Frames"
  - Appendix E (pages 517–522): "Ellipses"
- Watch videos on UTSOnline
- Work on Lab Assignment #1 (due in week 6 lab)

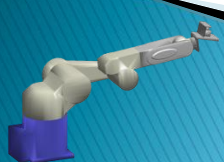
Note/slides from the textbook

- Textbook readings for week 4:
  - Section 7.4 (pages 152–158): “Trajectories
  - Section 7.3 (pages 146–152): “Inverse Kinematics”

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

Inverse Kinematics

- This is the inverse kinematics problem which is written in functional form as: $q = K^{-1}(\xi)$
- In general this function is not unique and for some classes of manipulator no closed form solution exists, necessitating a numerical solution.

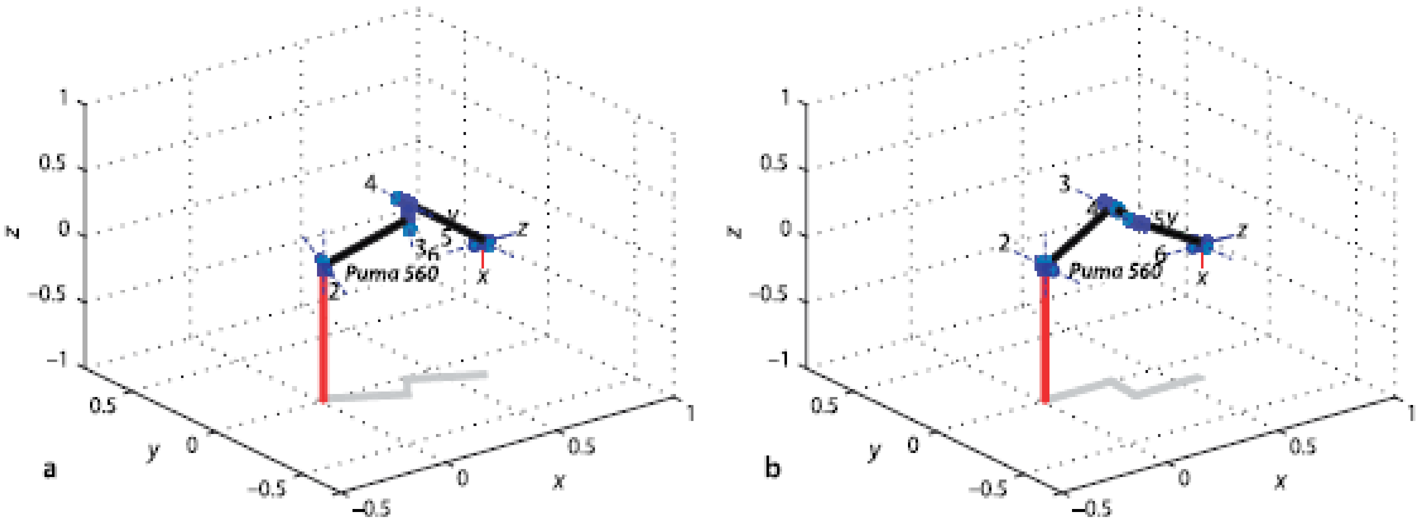


Closed-Form Solution

- A closed-form solution of a 6-axis robot is that the three wrist axes intersect at a single point.
- This means that motion of the wrist joints only changes the end-effector orientation, not its translation.

Closed-Form Solution (continued…)

- **Fig. 7.6.** Two solutions to the inverse kinematic problem, left-handed and right-handed solutions. The shadow shows clearly the two different configurations



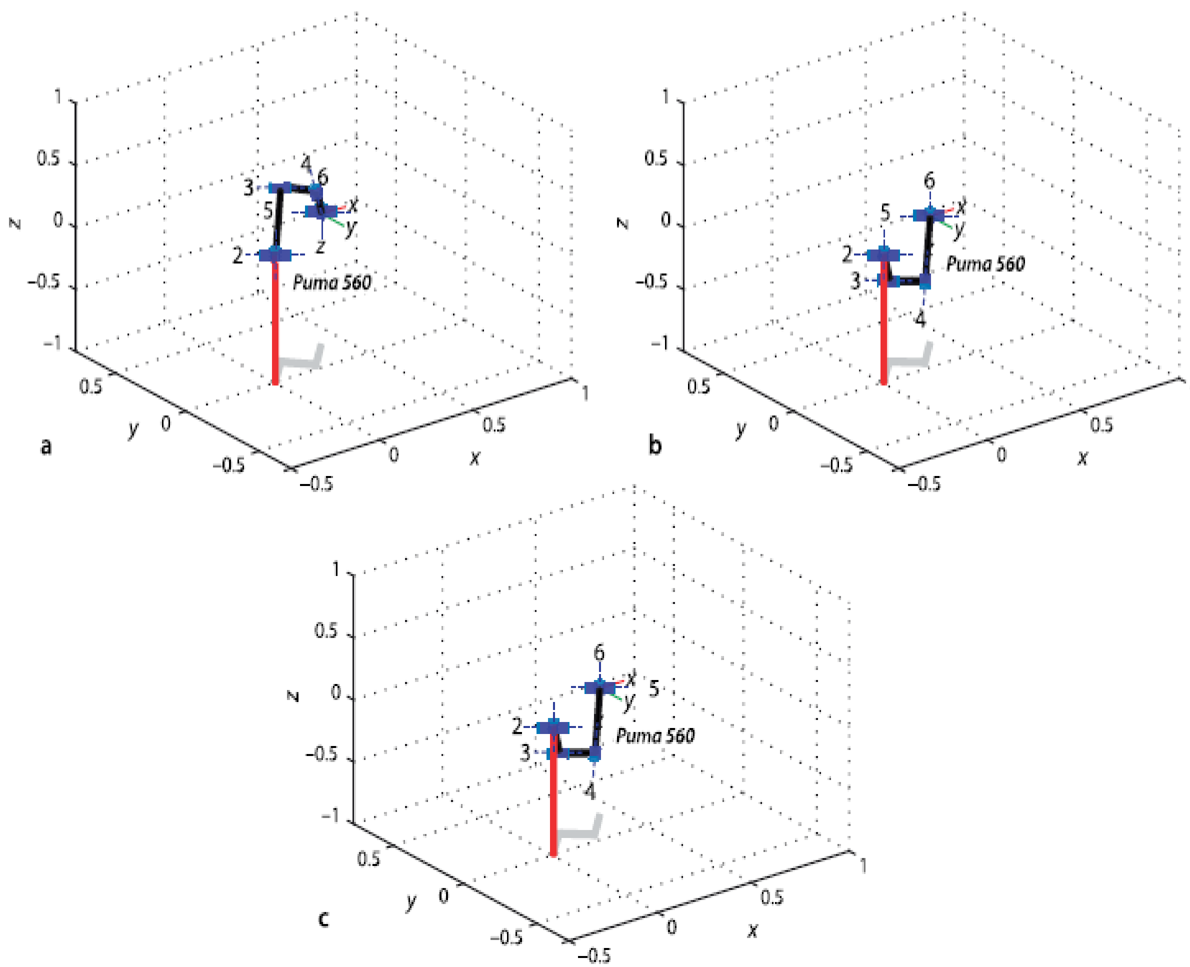
Closed-Form Solution (continued…)

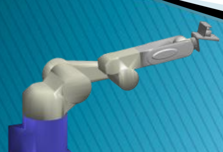
Fig. 7.7. Different configurations
of the Puma 560 robot

a) Right-up-noflip;

b) right-down-noflip;

c) right-down-flip





Under-Actuated Manipulator

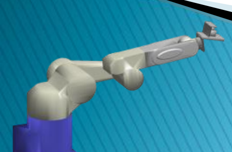
- An under-actuated manipulator is one that has fewer than six joints, and is therefore limited in the end-effector poses that it can attain.
- SCARA robots are a common example.
- They typically have an x–y–z–θ **task space,** $\mathcal{T} \subset \mathbb{R}^3 \times \mathbb{S}$ and a **configuration space** $\mathcal{C} \subset \mathbb{S}^3 \times \mathbb{R}$**.**

Redundant Manipulator

- A redundant manipulator is a robot with more than six joints.
- Six joints is theoretically sufficient to achieve any desired pose in a Cartesian taskspace $\mathcal{T} \subset SE(3)$.
- The Denavit-Hartenberg (DH) parameters for the base are

Link	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\propto_1}$	$\boldsymbol{\sigma_i}$
1	0	q_1	0	$-\frac{\pi}{2}$	1
2	$-\frac{\pi}{2}$	q_2	0	$\frac{\pi}{2}$	1

- The DH notation requires that prismatic joints cause translation in the local *z*-direction and the base-transform rotates that *z*-axis into the world *x*-axis direction.



Redundant Manipulator

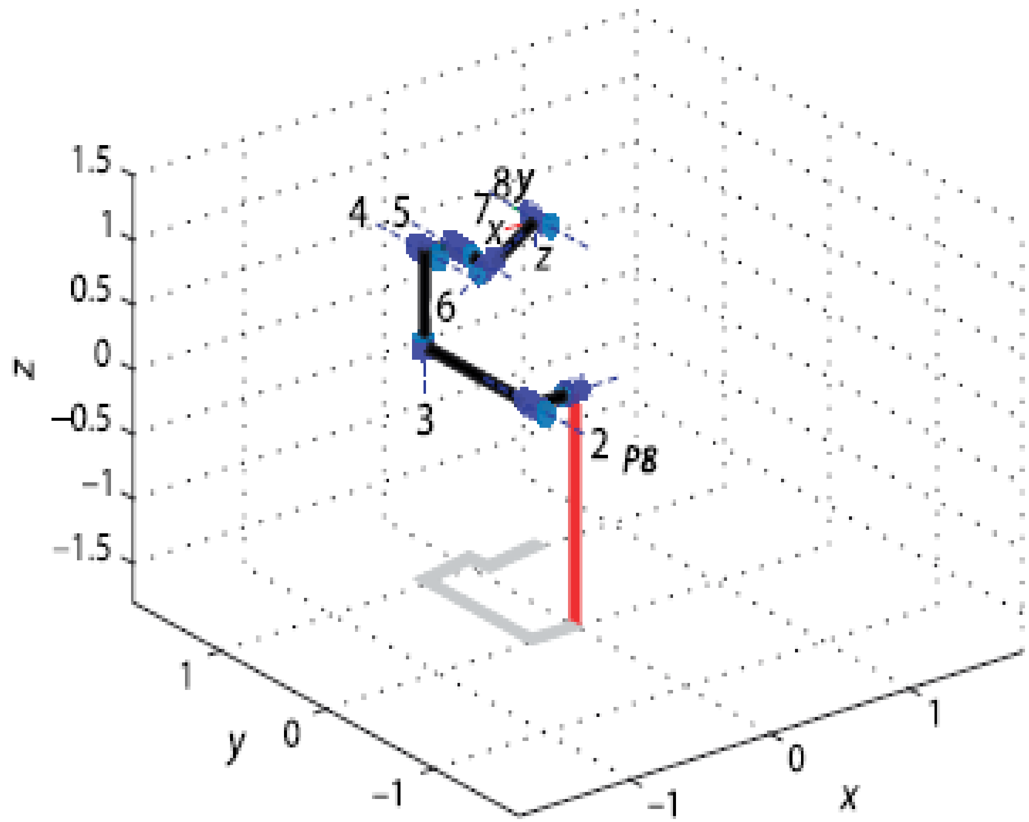


Fig. 7.8. Redundant manipulator

Trajectories

- One of the most common requirements in robotics is to move the end-effector smoothly from pose A to pose B.
- Trajectories has two approach:
  - straight lines in joint space
  - straight lines in Cartesian space.

Joint-Space Motion

- A joint-space trajectory is formed by smoothly interpolating between two configurations q1 and q2.
- The joint coordinate trajectory is smooth but we do not know how the robot’s end-effector will move in Cartesian space.

Joint–Space Motion (continued…)

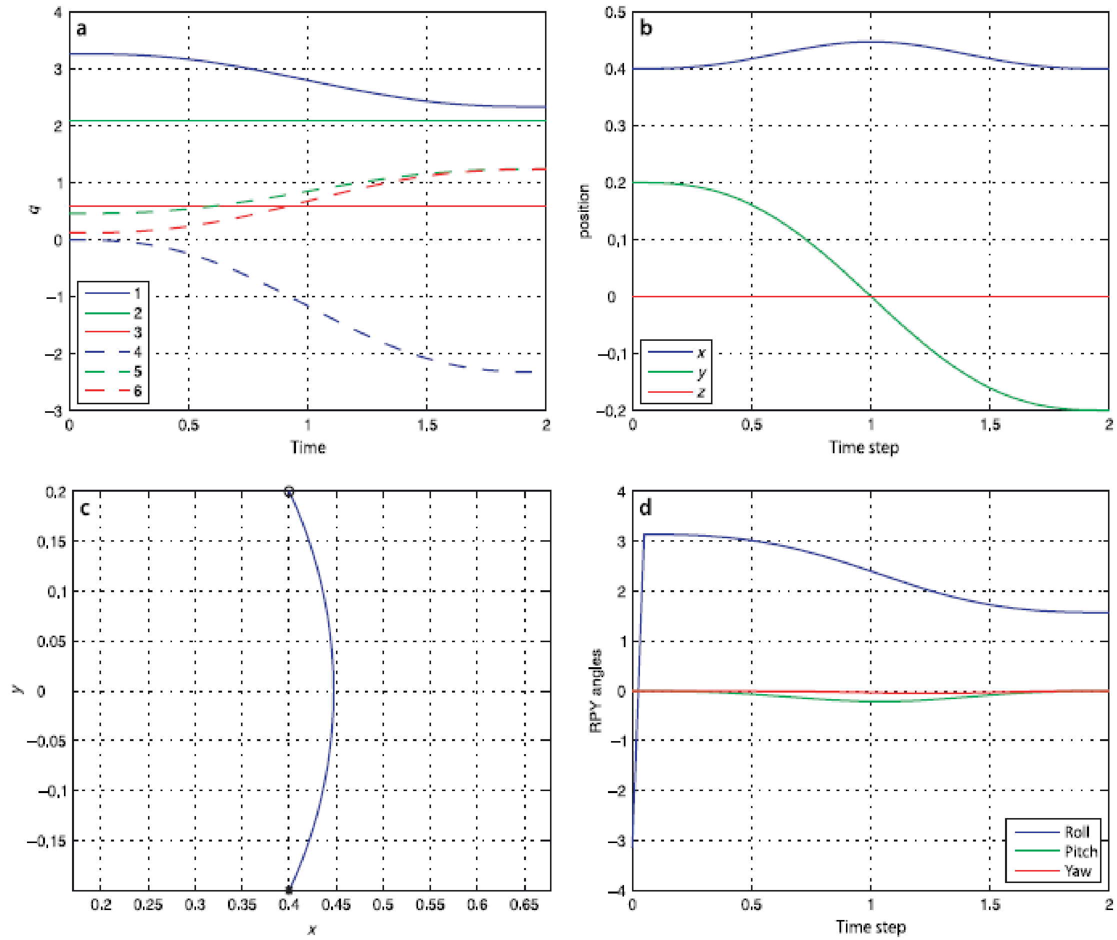
Fig. 7.9. Joint space motion.

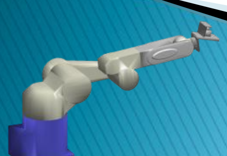
a Joint coordinates vs time;

b Cartesian position vs time;

c Cartesian position locus in the *xy*–plane

d roll–pitch–yaw angles vs time





Cartesian Motion

- The joint alignment means that the robot has lost one degree of freedom and is now effectively a 5-axis robot.
- Kinematically we can only solve for the sum $q_4 + q_6$ and there are an infinite number of solutions for q_4 and q_6 that would have the same sum.

Cartesian Motion (continued…)

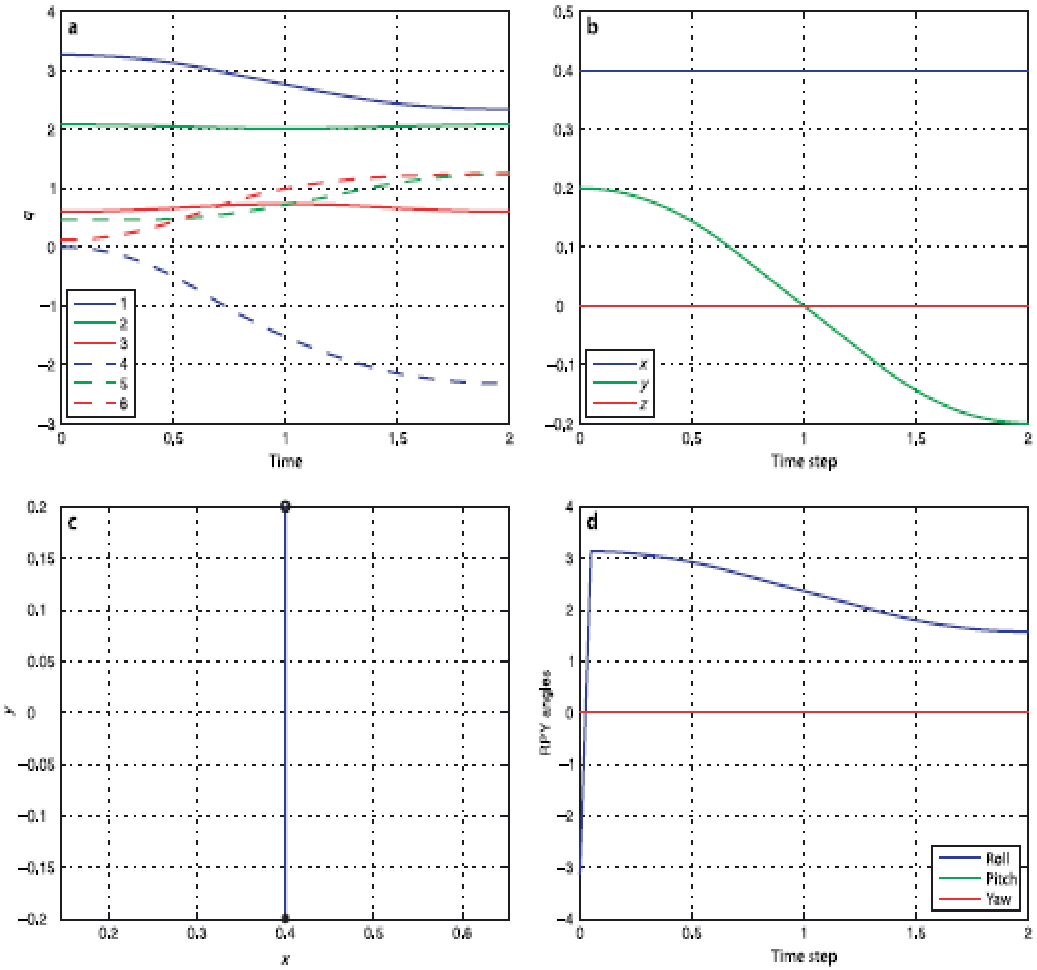
Fig. 7.11. Cartesian motion.

a) Joint coordinates vs time;

b) Cartesian position vs time;

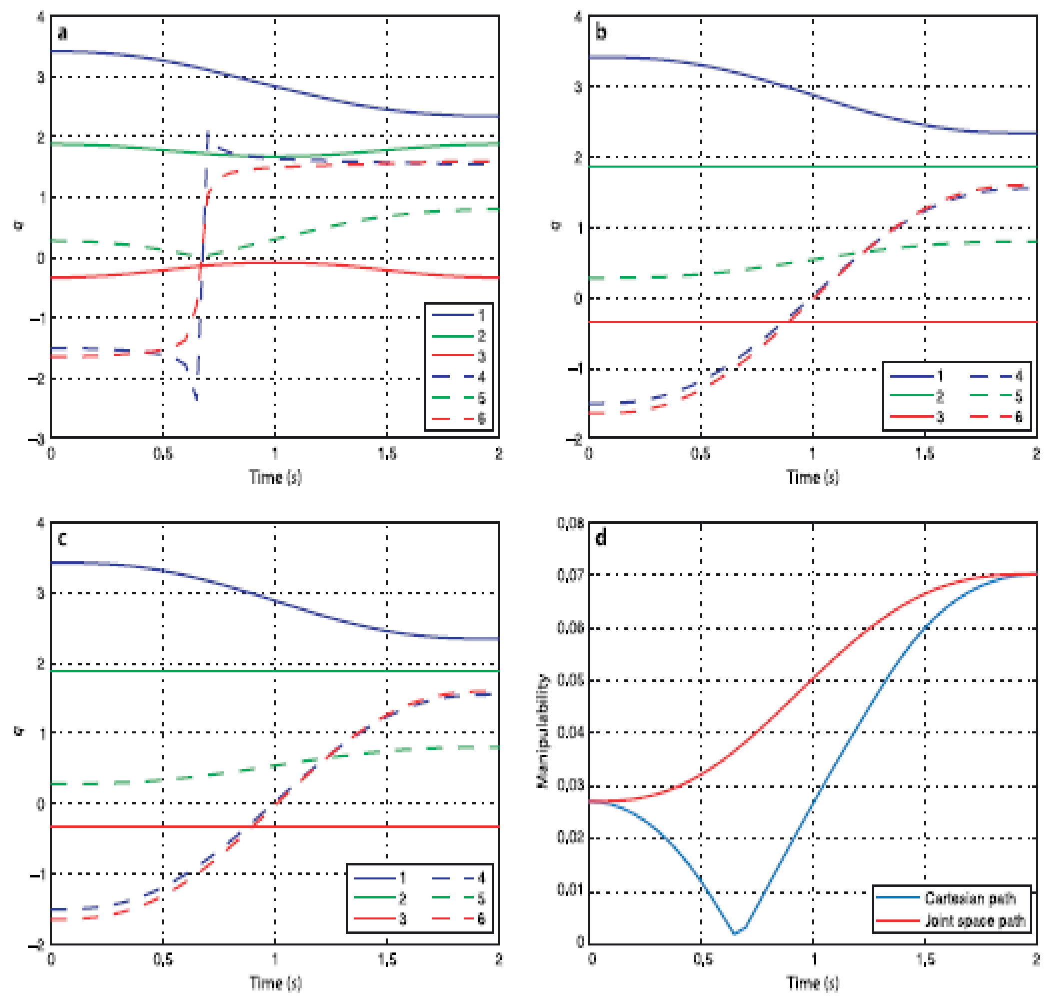
c) Cartesian position locus in the *xy*–plane;

d) roll–pitch–yaw angles vs time



Cartesian Motion (continued…)

- **Fig. 7.12.** Cartesian motion through a wrist singularity.
  - **a)** Joint coordinates computed by inverse kinematics (ikine6s);
  - **b)** joint coordinates computed by generalized inverse kinematics (ikine);
  - **c)** joint coordinates for joint-space motion;
  - **d)** manipulability

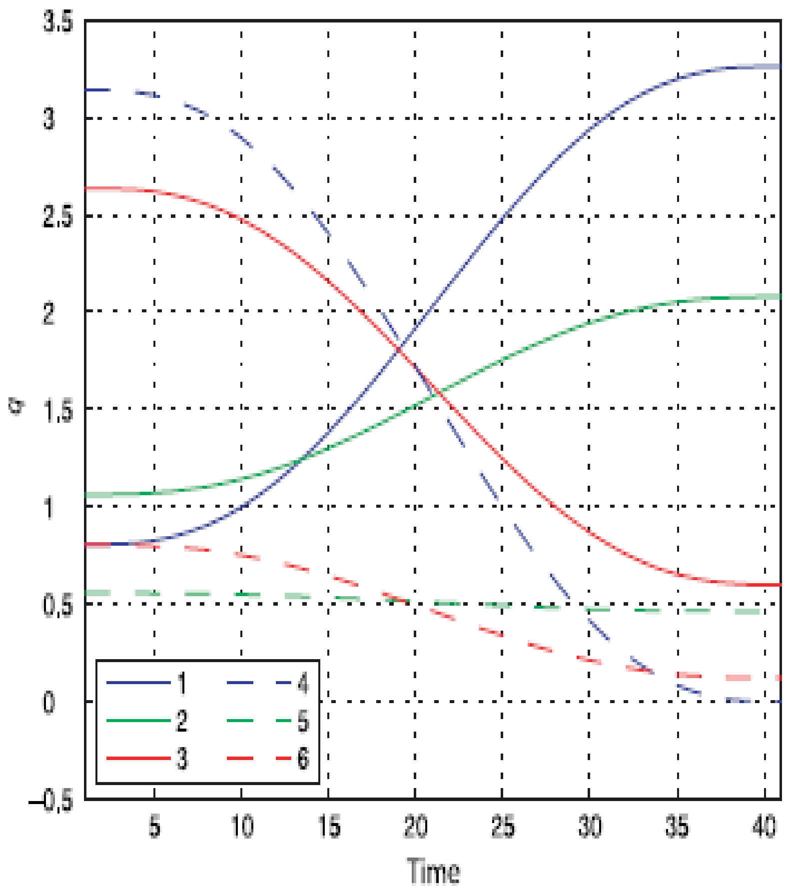


Configuration Change

- Movement from one configuration to another ultimately results in no change in the end-effector pose since both configuration have the same kinematic solution
- Therefore we *cannot* create a trajectory in Cartesian space. Instead we must use joint-space motion.

Configuration Change (continued…)

- **Fig. 7.13.** Joint space motions for configuration change from right-handed to left-handed



References

- [1] Robotics, Vision and Control. Peter Corke