Aalto University
School of Electrical Engineering

Introduction

ELEC-C1320 Robotics

Pekka Forsman

Topics

- Course structure
  - MyCourses platform
  - Classroom lectures
  - Textbook(s)
  - Matlab + robotics toolbox
  - Robot Offline Programming exercise
  - Grading of the course
- Overview of the course contents

Course structure

The course has **double lectures** on Thursdays, first the weekly subject is discussed in Finnish 14:15-16 o'clock followed by the English lecture event 16:15-18 o'clock (especially for the students of ELEC-**D**1320 course). Both lectures take place in lecture hall AS1. (exception: Finnish language lecture 7.9 is in TU2)

Every Tuesday afternoon, 16-18 o'clock, a **calculation exercise session** ”laskutupa” is arranged (starting 19.9) in lecture hall AS1. There you can get guidance/advice in robotics course related issues (especially solving homework problems).

On the evaluation week of the I-period, 16.-20.10, there are no lectures or exercises.

The lecturer of the course is Pekka Forsman, pekka.forsman@aalto.fi, +358 505669389. He will also be the advisor in the calculation excercise sessions. The other advisor in the calculation excercise sessions is Alvar de Wit, alvar.de.wit@aalto.fi.

Alvar de Wit is also responsible of the supervision of the Robot Offline Programming exercise.

MyCourses platform

The MyCourses platform will be used to keep the course material such as lecture slides, lecture videos, homework problem definition documents and model solution files as well as source material for the Robot Offline Programming exercise.

Also, solutions to the homework problems as well as deliverables for the Robot Offline Programming exercise will be returned through MyCourses platform by the students.

Classroom lectures

Lecture slides and **prerecorded lecture videos** are published in MyCourses Materials-section before the classroom lectures.

The actual classroom lecture events are NOT recorded so you are free to ask questions and participate to the discussions.

Textbook(s)

The main textbook of the course is Peter Corke, Robotics, Vision and Control, Fundamental Algorithms in MATLAB, **Second Edition**, Springer, 2017. The book can be accessed/downloaded from a computer connected to the Aalto network from https://link.springer.com/book/10.1007/978-3-319-54413-7.
Part of the chapters 2,3,7,8 and 9 are covered during the course. (*for page numbers of the book to be covered see the last page of the lecture slides each week*)

The second textbook is John J. Craig, Introduction to Robotics: Mechanics and Control, Third Edition, Prentice Hall, 2005. Chapters 3 and 4 are partly covered during the course.
(*Note that any content, relevant to this course, from the Craig's text book can also be found in the lecture slides or homework problem definition documents so that you don't necessarily need to acquire a copy of your own of Craig's text book*)

Matlab + robotics toolbox

Peter Corke’s textbook utilizes extensively Matlab-exercises to demonstrate methods and algorithms. The Matlab Robotics toolbox, Release 10, accompanying the book, can be downloaded from: https://petercorke.com/toolboxes/robotics-toolbox/

For the toolbox you will need a recent version of Matlab (e.g. version R2019a or later). Matlab is available for free for Aalto students and can be downloaded from https://download.aalto.fi/index-en.html

Robot Offline Programming exercise

The Robot Offline Programming exercise for the robotics course can be done individually or in groups of maximum of four students.

Each person doing the exercise alone or each group provides one set of deliverables. The exercise consists of a preliminary assignment (*consisting of 8 basic assignments*) and the main assignment.

Upon successful completion of the main assignment up to 20% of the total points for the grade of the course can be earned. To pass the course a minimum of 50% of the maximum points of the main assignment must be earned. 大作业的一半分

For more information, please take a look at the ”Robot Offline Programming Exercise” Section of the MyCourses pages of ELEC-C1320.

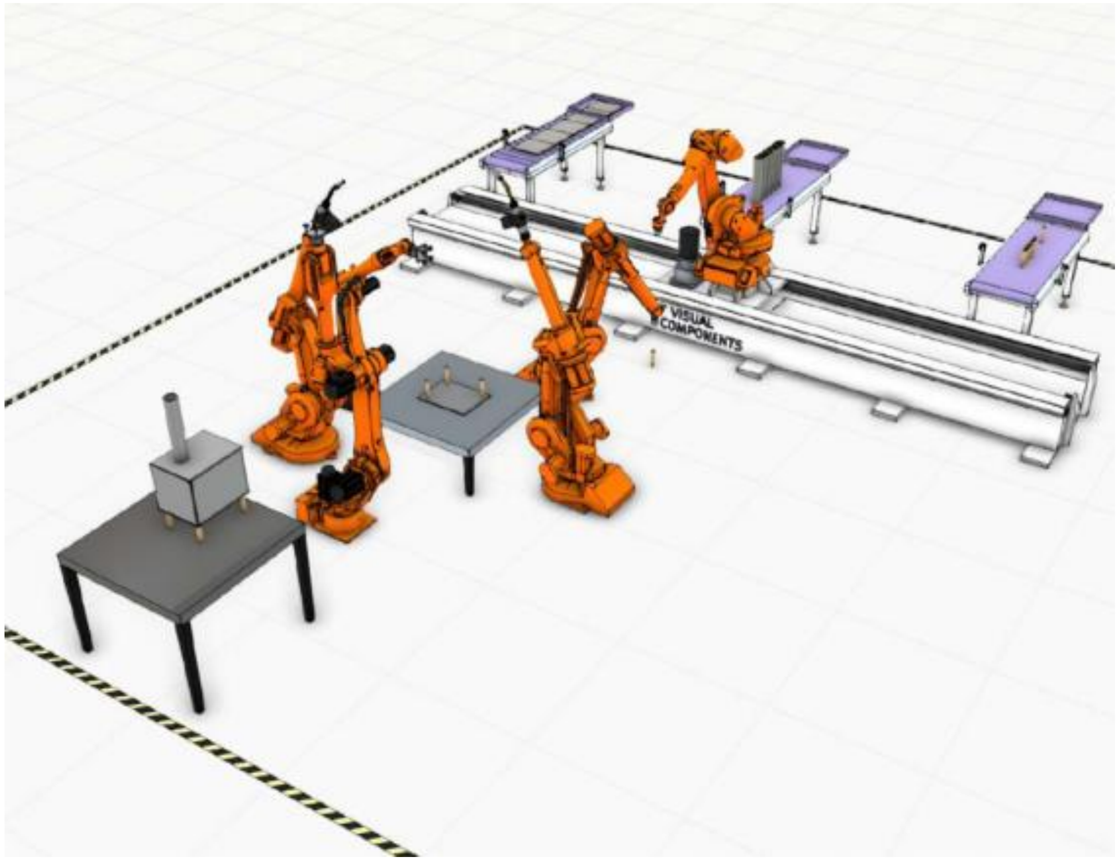
”Robot Offline Programming Exercise” Section will be published in MyCourses in the week of 18-22 of September.

The software platform for the Robot Offline Programming exercise is *Visual Components* 4.3 Premium together with *Delfoi* add-on module produced by the Finland based companies *Visual Components* and *Delfoi* respectively.

The exercise and the documentation have been designed such that the work can be carried out independently without instructor's help. If you face some major problems doing the exercise, you can contact the teaching assistant Alvar de Wit.

More detailed information about the exercise can be found in the "Robot Offline Programming Exercise" section of the ELEC-C1320 MyCourses pages.

The software can be found in the computers of selected classrooms (*see Offline Prog **Manual**, p.7 for more details*). The software can also be accessed via a remote 3D desktop connection.



A snapshot of the robot manufacturing unit simulation model for assembling Finnish sauna water heaters.

Grading of the course

The grading of the course will be based on the final exam (corresponding to maximum 60% of total points = 100 points), on the solutions of the homework problems (20% = 4x5% of total points) and on the main assignment of the robot off-line programming exercise (20% of total points).

To pass the course a minimum of about 45% of the maximum points in the final exam as well as good enough completion of the main assignment of the robot offline programming exercise are required. To get grade 1 for the course about 45 % of the total points is required. For grade 2 about 55 % of the total points is required and so on.

Homework schedule–ELEC-C1320/ELEC-D1320

In the exercise sessions (laskutupa) you can get support for solving the homework problems. The exercise sessions are on Tuesday evenings, 16-18 o'clock in lecture hall AS1, starting 19.9.

Homework 1, Release date 14.9 – Deadline for the solutions 5.10 noon

Homework 2, Release date 5.10 – Deadline for the solutions 26.10 noon

Homework 3, Release date 26.10 – Deadline for the solutions 9.11 noon

Homework 4, Release date 9.11 – Deadline for the solutions 30.11 noon

Lecture schedule – ELEC-C1320/ELEC-D1320

Lecture 1, Introduction, Thursday 7.9

Lecture 2, Representing position and orientation, Thursday 14.9

Lecture 3, Time and motion; part 1, Thursday 21.9

Lecture 4, Time and motion; part 2, Thursday 28.9

Lecture 5, Robot arm forward kinematics, Thursday 5.10

Lecture 6, Inverse kinematics (closed form solution), Thursday 12.10

16.10 – 20.10 evaluation week, no lectures or exercises

Lecture 7, Velocity relationships, part 1, Thursday 26.10

Lecture 8, Velocity relationships, part 2;

Dynamics and control of serial-link manipulator, Thursday 2.11

Lecture 9, Inverse kinematics (numerical solution); Trajectories, Thursday 9.11

Lecture 10, Kinematic motion models of mobile robots, Thursday 16.11

20.11 – 24.11 exercises normally, backup lecture time (no lectures preplanned)

Lecture 11, Wrap-up of the course contents, Thursday 30.11

Alternative course exam, Monday 4.12, 16:30-19:30 o'clock, lec. hall TU2

Course exam, Thursday 7.12, 13:00-16:00 o'clock, lecture hall AS2

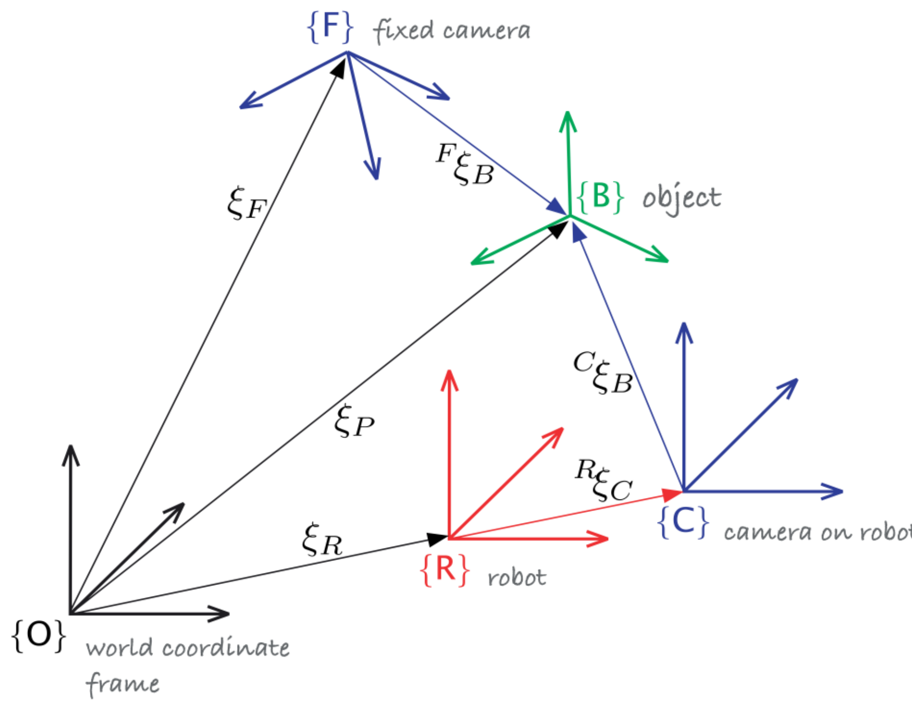
Lecture 2, Representing position and orientation

通过组合（链接）坐标系之间的变换来建立方程

We can make equations by combining (linking) transformations between coordinate frames:

$$\xi_F \oplus {}^F\xi_B = \xi_R \oplus {}^R\xi_C \oplus {}^C\xi_B$$

$$\xi_F \oplus {}^F\xi_R = {}^0\xi_R$$



Lecture 3, Time and motion; part 1

正交旋转
矩阵R如
何作为
角度函数
变化速度

We will study (amongst other things) how the ***orthonormal rotation matrix R changes as a function of angular velocity*** (Eqs. See Corke's text book)

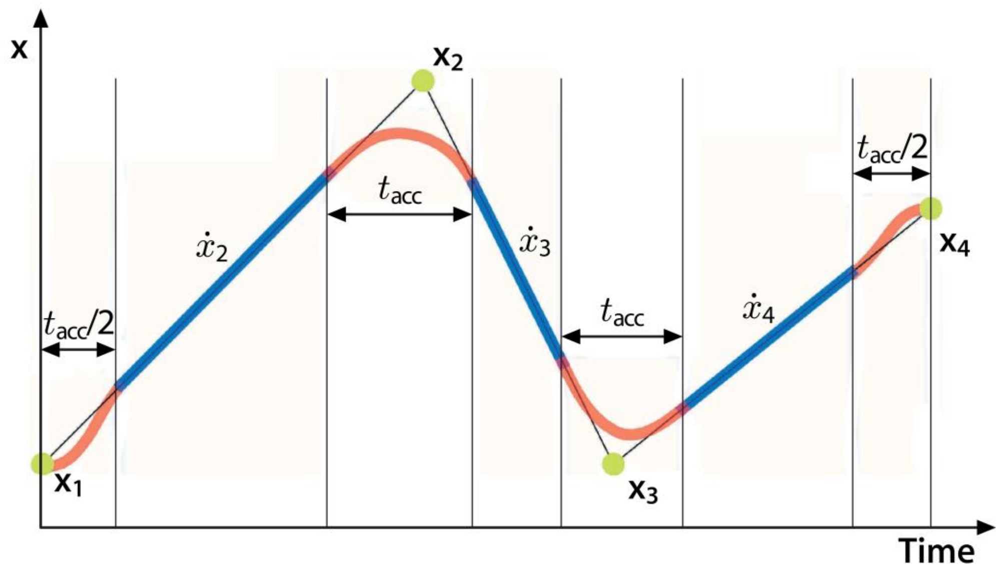
$$[\boldsymbol{\omega}]_\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \qquad (2.19)$$

$$^{A}\dot{\boldsymbol{R}}_B = {}^{A}\boldsymbol{R}_B \left[{}^{B}\boldsymbol{\omega}\right]_\times \in \mathbb{R}^{3\times 3} \qquad (3.2)$$

$$\boldsymbol{R}_B\langle t+\delta_t\rangle \approx \boldsymbol{R}_B\langle t\rangle + \delta_t \boldsymbol{R}_B\langle t\rangle[\boldsymbol{\omega}]_\times \qquad (3.7)$$

Lecture 4, Time and motion; part 2

A one-dimensional example of a multi-segment trajectory showing four positions and three motion segments



多段轨迹显示四个位置和三个运动段

Lecture 5, Robot arm forward kinematics

Forward kinematics aims at determining the equations that describe the position and orientation (pose) of the cartesian coordinate frame attached at the tip of the robot kinematic chain as a function of robot joint values

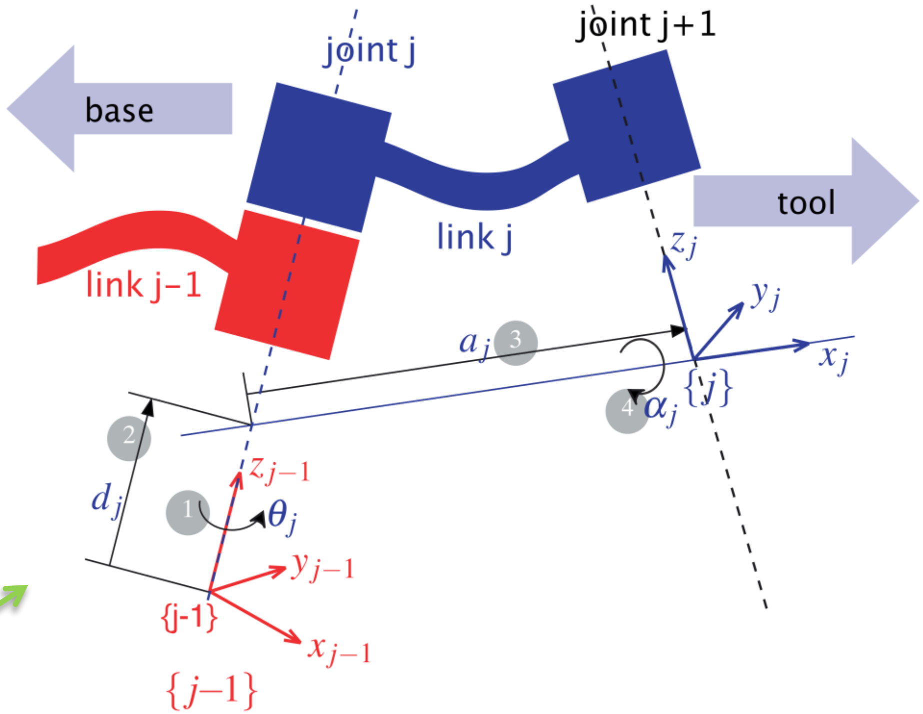
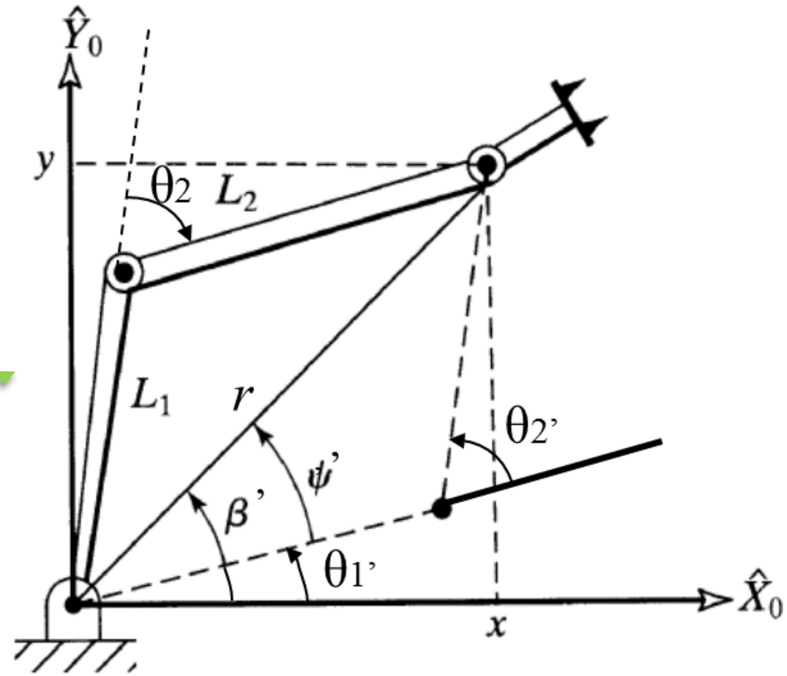


Illustration of **standard** Denavit and Hartenberg (DH) link parameters

Lecture 6, Inverse kinematics (closed form solution)

Inverse kinematics aims at determining the equations for calculating the robot joint values as a function of the desired robot tool frame pose.

Inverse kinematics calculation: geometric approach



Lecture 7, Velocity relationships, part 1

A Jacobian is the matrix equivalent of the derivative – the derivative of a vector-valued function of a vector with respect to a vector. If $\boldsymbol{y} = F(\boldsymbol{x})$ and $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ then the Jacobian is the $m \times n$ matrix

$$J = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}$$

The Jacobian is named after Carl Jacobi, and more details are given in Appendix G.

For example, if the forward kinematics model of the 2-degree of freedom (dof) planar manipulator is:

$x_r = l_1c_1 + l_2c_{12}$

$y_r = l_1s_1 + l_2s_{12}$

$\phi_r = \theta_1 + \theta_2$

then we get the Jacobian matrix:

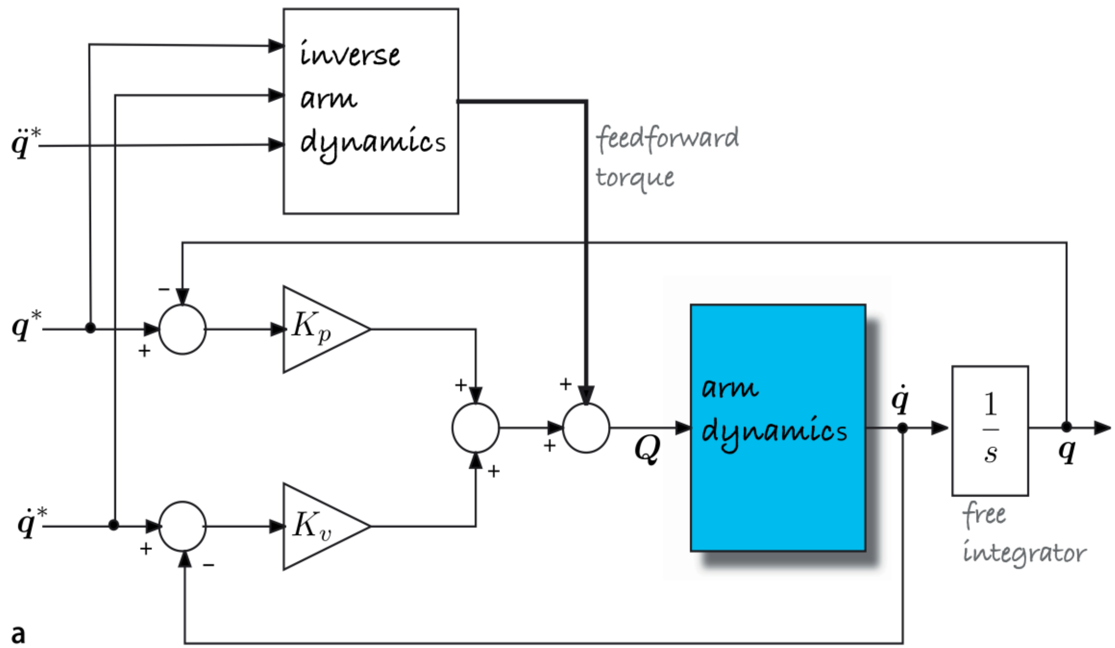
$$J_r(q) = \begin{bmatrix} -l_1s_1 - l_2s_{12} & -l_2s_{12} \\ l_1c_1 + l_2c_{12} & l_2c_{12} \\ 1 & 1 \end{bmatrix}$$

which obeys the equation:

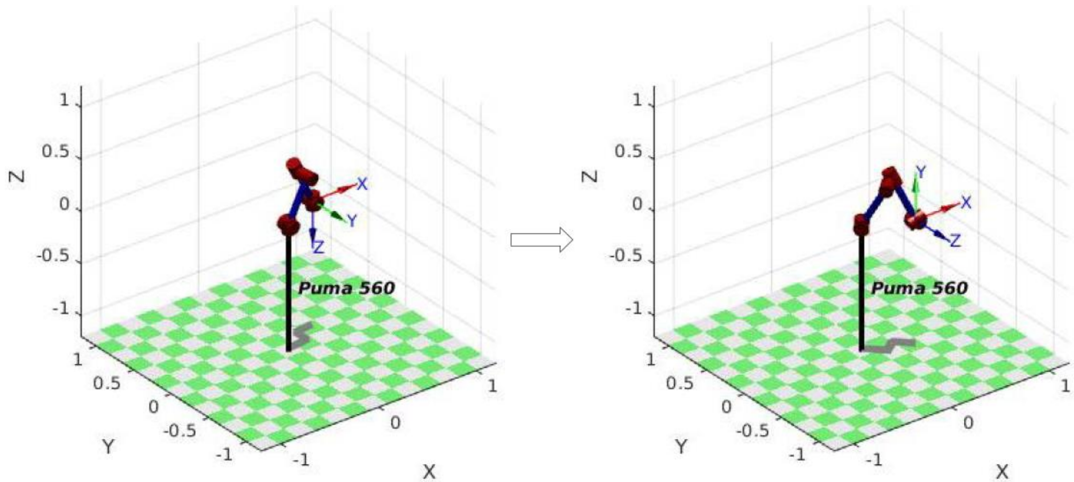
$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = J_r(q) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Lecture 8, Velocity relationships part 2; Dynamics and control of serial-link manipulator

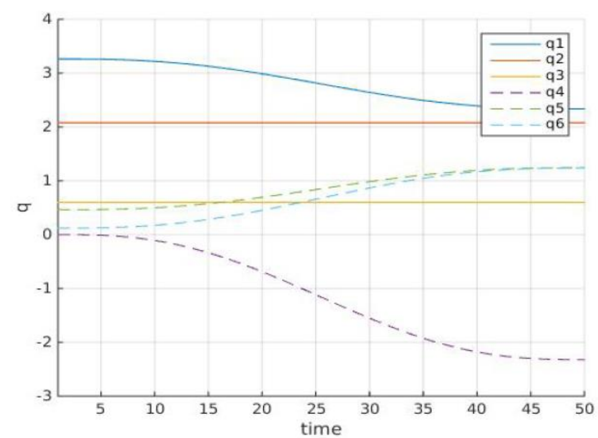
The inverse arm dynamics model is used to calculate torque components *(one for each dof)* for the robot controller to compensate for the non-linear robot forces and torques due to the motion of the robot arm.



Lecture 9, Inverse kinematics (numerical solution); Trajectories



On the image to the right, the paths of the robot joint values as a function of time (i.e. trajectories) to move the robot tool frame from the initial configuration to the final configuration are shown.



Lecture 10, Kinematic models of mobile robot platforms

The kinematic models of mobile robot platforms are required for relating vehicle control parameters (such as steering angle and wheel speeds) to the change of vehicle location on the plane