Project Assignment 2020 Robotics

Robotics 31383
Department of Electrical Engineering • Technical University of Denmark

29 September 2020

Please read pages 2-3 before you start of the robot project assignment

PURPOSE:

The purpose of this project assignment is to practice the topics learned in the Robotics course 31383 on a realistic robotic application.

INTRODUCTION:

In this project assignment consider an *Alto Robot* from the Ramsta Robotics Company. This robot is applied in automatic water jet cleaning of livestock buildings (e.g. pig pens) and containers, etc. Please refer to the front page of this document for a picture of the robot itself, as well as a picture of the robot in action with the water jet cleaning of a pig pen.
The project assignment treats tasks with questions/problems in simulation, dynamics, direct kinematics, inverse kinematics, singularities, trajectory planning as well as control, etc. of the Alto robot. The Alto Robot is an electrically driven spherical robot with four degrees of freedom (rotating tower and shoulder, prismatic upper arm, and rotating elbow joint – please refer to Appendix A for further details). The robot's tool is a folding high-pressure unit with a rotating head (refer to the picture on the front page of this note). In this project assignment the high-pressure unit is unfolded and the functionality (DOF) of the rotating head is neglected. The physical robot is mounted on a moving base, but in this assignment the base is considered stationary and without dynamics i.e. fixed to the floor.

Anecdote: DTU has participated in a research and development project concerning the development of a camera based sensor system for detection of dirt in pig pens. In short or long term, robots such as the Alto robot, will be integrated with this sensor system into a fully automated cleaning system that is much needed due to the fact that manual cleaning of livestock buildings, using high-pressure water technology, is one of the most tedious and health threatening tasks which is conducted by human labour. The cleaning process itself contributes to deterioration of the working environment due to stirring up dirt, microorganisms and water, which is inhaled by the operator. Consequently, the working conditions for personnel who are performing today's cleaning of livestock buildings are essential to solve.

ABOUT THIS MATERIAL:

This paper describes a project assignment tasks that consists of **21 -problems** denoted 'Problem 1' to 'Problem 21'. Four Annexes (denoted A to D) have been attached to support the execution of the project tasks. Annex A is a description of the Alto Robot used in the project assignment, Annex B consists of dynamic calculations related to the Alto Robot, Annex C contains a description of the 'independent joint control principle', and finally Annex D is a description of simulation blocks for robot simulation in Matlab/Simulink (for solving problem 17-21).

CONTENTS:

The direct- and inverse kinematic transformations of the Alto Robot are addressed in *Problem 1 and 2*. The inverse kinematic transformation is applied in *Problem 3* to calculated four joint angles corresponding to four given Cartesian knot-points located on a tool path for the robot to follow.

In *Problem 4 and 5* Cartesian and Joint-space trajectories are calculated along the tool path for the robot to do a simple cleaning task (simulated in *Problems 17, 18, 19, 20* and *21*).

The four given Cartesian knot-points are checked for singularities in *Problem 6*.

In *Problem 7* the inertia tensor of the robot is determined. This inertia tensor can be used to derive a non-linear, coupled dynamic model of the robot. This derivation work has been done for you and reported in Appendix B. A DC motor model is considered in *Problem 8* for joint control. The dynamic model from Appendix B is used in *Problem 9* to calculate a linearized, decoupled dynamic model of the robot. This approximated dynamic model is derived from the calculated, worst-case moments of inertia as seen from the motor axes.

In *Problem 11*, a PD controller is investigated for the robot and subsequently implemented in Matlab/Simulink in *Problem 12*. From *Problem 13* the gravitational effects of the manipulator are included into the model as worst case torque load. A PID controller is implemented into the model in *Problem 14* and *15*. In *Problem 16* a pre-filter is added for further enhancement of the control model.

In *Problems 17, 18, 19* and *20* and the robot tool movement is simulated in different setups. In *Problem 21* it is to be shown that the robot model including the motors can be written in a compact matrix-vector form well suited for simulation, etc.

SOLVING AND REPORTING THE PROJECT ASSIGNMENT:

Solving the numbered problems in this note carries out the project assignment. The answers shall be supported by a sufficient number of intermediate calculations and explanatory text, so that the principles and method used are clear. Plots should be numbered consecutive $A0, A1, ...A12$.

The report is assessed as a whole based on the quality of the explanatory text and the correctness of the answers. The last page of the report should be signed by the participant(s). Remember your 'study registration number'. The project assignment needs to be carried out by team of two or three persons, the individual contribution to the project work must be clearly indicated contributions to the answer for each of the 21 Problems presented in the project assignment report. Specify the work contribution a problem by the percentage scale, e.g. example Elisabeth 30% / Peter 30% / Jack 40%.

DEADLINES:

The project report must be submitted via DTU Learn, NO LATER THAN 5pm, 8.Dec.2020.

THE PROJECT ASSIGNMENT

PART 1: KINEMATICS

Consider the Alto robot shown in Annex A. The robot manipulator has four degrees of freedom (4DOF) used in this project assignment for positioning the robot manipulator in its workspace. *Do not consider the orientation of the tool in this project.*

•**Problem 1:** Find, by the use of Annex A, the direct kinematic transformation, T_4^0, for the robot manipulator. (T_4^0 is the tool centre frame, L_4's, position and orientation relative to base frame L_0.

•**Problem 2:** Determine the inverse kinematic transformation

$$q = [q_1, q_2, q_3, q_4]^T = f\big([x_4]^0, [p_4]^0\big)$$

where ($[x_4]^0$, $[p_4]^0$) are first and last column of T_4^0, respectively.

The robot manipulator is now supposed to clean the floor of the pig pen. In this assignment we only consider a simple cleaning path for the robot tool as shown in Figure 1. The path is defined by five knot-points (frames) T_{p1}^0, T_{p2}^0, T_{p3}^0, T_{p4}^0 and T_{p5}^0 through which the manipulator is supposed to move the robot tool.

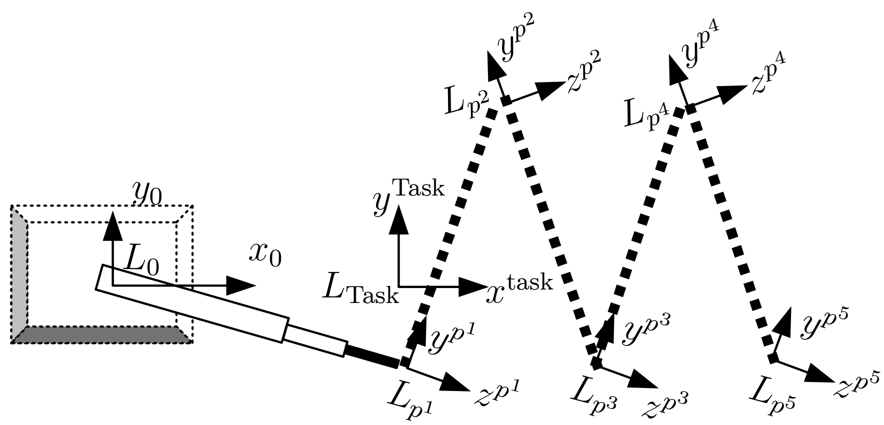


Figure 1: Work scene for the point to point movement.

The movement is done in the following order: $T_{p1}^0 \to T_{p2}^0$ (segment 1), $T_{p2}^0 \to T_{p3}^0$ (segment 2), $T_{p3}^0 \to T_{p4}^0$ (segment 3) and $T_{p4}^0 \to T_{p5}^0$ (segment 4). The *task frame* is given relative to the base frame L_0 of the robot manipulator by the following transformation:

$$T_{\text{Task}}^0 = \begin{bmatrix} 1 & 0 & 0 & 1.4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since all knot points are given in the task frame it is necessary to determine all knot in the base frame system instead of the *task frame*.

The duration of the movement is requested to be 8 seconds in total for all four segments. The via-points are given relative to the *task frame* L_{Task} and defined by the following homogenous transformation matrices (all measures are given in meters):

$$T_{p1}^{\text{Task}} = \begin{bmatrix} 0 & & & 0 \\ 0 & ? & ? & -0.5 \\ -1 & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p2}^{\text{Task}} = \begin{bmatrix} 0 & & & 0.35 \\ 0 & ? & ? & 0.5 \\ -1 & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p3}^{\text{Task}} = \begin{bmatrix} 0 & & & 0.7 \\ 0 & ? & ? & -0.5 \\ -1 & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{p4}^{\text{Task}} = \begin{bmatrix} 0 & & & 1.05 \\ 0 & ? & ? & 0.5 \\ -1 & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p5}^{\text{Task}} = \begin{bmatrix} 0 & & & 1.4 \\ 0 & ? & ? & -0.5 \\ -1 & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Some of the columns in the matrices above are marked with "?". The reason for this is, that knowing the value of these elements would make it possible to solve the problem, without using the inverse kinematics.

•**Problem 3:** Find the corresponding joint coordinates for each knot-point

$$q = [q_1, q_2, q_3, q_4]^T = f(([x_{P_j}]^0, [p_{P_j}]^0)), \quad j = 1, 2, 3, 4, 5$$

where $([x_{P_j}]^0, [p_{P_j}]^0)$ are first and last column, respectively, of $T_{P_j}^0, j = 1, 2, 3, 4, 5$.

PART 2: TRAJECTORY PLANNING

•**Problem 4:** The (inverse) transformed knot-points found in problem 3:

$$q = [q_1, q_2, q_3, q_4]^T = f\big(\big([x_{P_j}]^0, [p_{P_j}]^0\big)\big), \quad j = 1, 2, 3, 4, 5$$

are now to be connected by means of 5. order polynomials. In the knot-points the joint speed and acceleration are 0 radians/sec and 0 radians/sec^2, respectively. Each segment has a duration time of 2 seconds for each run trough. The polynomials are defined by:

Segment 1: $\{1\} \rightarrow \{2\}$, $0 \leq t \leq 2$

$$\begin{aligned}
q_1(t) &= f1(t) = a15 \cdot t^5 + a14 \cdot t^4 + a13 \cdot t^3 + a12 \cdot t^2 + a11 \cdot t + a10, \\
q_2(t) &= f2(t) = a25 \cdot t^5 + a24 \cdot t^4 + a23 \cdot t^3 + a22 \cdot t^2 + a21 \cdot t + a20 \\
q_3(t) &= f3(t) = a35 \cdot t^5 + a34 \cdot t^4 + a33 \cdot t^3 + a32 \cdot t^2 + a31 \cdot t + a30 \\
q_4(t) &= f4(t) = a45 \cdot t^5 + a44 \cdot t^4 + a43 \cdot t^3 + a42 \cdot t^2 + a41 \cdot t + a40
\end{aligned}$$

Segment 2: $\{2\} \rightarrow \{3\}$, $0 \leq t \leq 2$

$$\begin{aligned}
q_1(t) &= f5(t) = a55 \cdot t^5 + a54 \cdot t^4 + a53 \cdot t^3 + a52 \cdot t^2 + a51 \cdot t + a50 \\
q_2(t) &= f6(t) = a65 \cdot t^5 + a64 \cdot t^4 + a63 \cdot t^3 + a62 \cdot t^2 + a61 \cdot t + a60 \\
q_3(t) &= f7(t) = a75 \cdot t^5 + a74 \cdot t^4 + a73 \cdot t^3 + a72 \cdot t^2 + a71 \cdot t + a70 \\
q_4(t) &= f8(t) = a85 \cdot t^5 + a84 \cdot t^4 + a83 \cdot t^3 + a82 \cdot t^2 + a81 \cdot t + a80
\end{aligned}$$

Segment 3: $\{3\} \rightarrow \{4\}$, $0 \leq t \leq 2$

$$\begin{aligned}
q_1(t) &= f9(t) = a95 \cdot t^5 + a94 \cdot t^4 + a93 \cdot t^3 + a92 \cdot t^2 + a91 \cdot t + a90 \\
q_2(t) &= f10(t) = a105 \cdot t^5 + a104 \cdot t^4 + a103 \cdot t^3 + a102 \cdot t^2 + a101 \cdot t + a100 \\
q_3(t) &= f11(t) = a115 \cdot t^5 + a114 \cdot t^4 + a113 \cdot t^3 + a112 \cdot t^2 + a111 \cdot t + a110 \\
q_4(t) &= f12(t) = a125 \cdot t^5 + a124 \cdot t^4 + a123 \cdot t^3 + a122 \cdot t^2 + a121 \cdot t + a120
\end{aligned}$$

Segment 4: $\{4\} \rightarrow \{5\}$, $0 \leq t \leq 2$

$$\begin{aligned}
q_1(t) &= f13(t) = a135 \cdot t^5 + a134 \cdot t^4 + a133 \cdot t^3 + a132 \cdot t^2 + a131 \cdot t + a130 \\
q_2(t) &= f14(t) = a145 \cdot t^5 + a144 \cdot t^4 + a143 \cdot t^3 + a142 \cdot t^2 + a141 \cdot t + a140 \\
q_3(t) &= f15(t) = a155 \cdot t^5 + a154 \cdot t^4 + a153 \cdot t^3 + a152 \cdot t^2 + a151 \cdot t + a150 \\
q_4(t) &= f16(t) = a165 \cdot t^5 + a164 \cdot t^4 + a163 \cdot t^3 + a162 \cdot t^2 + a161 \cdot t + a160
\end{aligned}$$

Determine the coefficients a_{ij} where $i \in \{1, 2, \ldots, 16\}$ and $j \in \{0, 1, 2, 3, 4, 5\}$, and the time is $t = 0$ seconds at the beginning of each segment.

The robot manipulator is now supposed to be moved along straight lines between the knot-points (Cartesian path control) in the same sequence as before. Again, the duration time for each segment is 2 seconds. The accelerations and speeds are always zero (0) in the via-points. The time is $t = 0$ seconds at the beginning of each line segment.

For the Cartesian path planner we use 5. order polynomials similar to the ones we used for the joint path planner in Problem 4. These polynomials are defined by:

Segment 1: $\{1\} \rightarrow \{2\}$, $0 \le t \le 2$

$$p_X(t) = f1(t) = a15 \cdot t^5 + a14 \cdot t^4 + a13 \cdot t^3 + a12 \cdot t^2 + a11 \cdot t + a10$$
$$p_Y(t) = f2(t) = a25 \cdot t^5 + a24 \cdot t^4 + a23 \cdot t^3 + a22 \cdot t^2 + a21 \cdot t + a20$$

Segment 2: $\{2\} \rightarrow \{3\}$, $0 \le t \le 2$

$$p_X(t) = f3(t) = a35 \cdot t^5 + a34 \cdot t^4 + a33 \cdot t^3 + a32 \cdot t^2 + a31 \cdot t + a30$$
$$p_Y(t) = f4(t) = a45 \cdot t^5 + a44 \cdot t^4 + a43 \cdot t^3 + a42 \cdot t^2 + a41 \cdot t + a40$$

Segment 3: $\{3\} \rightarrow \{4\}$, $0 \le t \le 2$

$$p_X(t) = f5(t) = a55 \cdot t^5 + a54 \cdot t^4 + a53 \cdot t^3 + a52 \cdot t^2 + a51 \cdot t + a50$$
$$p_Y(t) = f6(t) = a65 \cdot t^5 + a64 \cdot t^4 + a63 \cdot t^3 + a62 \cdot t^2 + a61 \cdot t + a60$$

Segment 4: $\{4\} \rightarrow \{5\}$, $0 \le t \le 2$

$$p_X(t) = f7(t) = a75 \cdot t^5 + a74 \cdot t^4 + a73 \cdot t^3 + a72 \cdot t^2 + a71 \cdot t + a70$$
$$p_Y(t) = f8(t) = a85 \cdot t^5 + a84 \cdot t^4 + a83 \cdot t^3 + a82 \cdot t^2 + a81 \cdot t + a80$$

where p_x, p_y are the x, y coordinates of the five knot-points in Fig. 1 with regard to the robot base frame.

•**Problem 5:** Determine the coefficients a_{ij} where $i \in \{1, 2, \ldots, 8\}$ and $j \in \{0, 1, 2, \ldots, 5\}$.

Part 3: SINGULARITIES (DIFFERENTIAL MOTION AND STATICS)

•**Problem 6:** Determine the Jacobian matrix of the manipulator. Using the Jacobian matrix to determine, whether there are any singularities along the path in Cartesian space found in Problem 5.

PART 4: ROBOT MANIPULATOR DYNAMICS

The following assumptions can be done in order to determine the center of gravity and the mass moments of inertia of each of the links:
Link 1 is a cylinder with radius r_1, length L_1, and mass m_1. The matrix of inertia (in $\mathbf{C}M^1$) for link 1 in L_1 is defined by:

$$\bar{D}_1 = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_{1yy} & 0 \\ 0 & 0 & I_1 \end{bmatrix}, \quad \text{where } I_1 = I_{1xx} = I_{1zz}.$$

[1] CM=center of Mass

Link 2 is considered as a box with sides $b_2 \times b_2$, length L_2, and a mass of m_2. The matrix of inertia (in CM) for link 2 described in L_2 is defined by:

$$\bar{D}_2 = \begin{bmatrix} I_2 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_{2zz} \end{bmatrix}, \quad \text{where } I_2 = I_{2xx} = I_{2yy}.$$

Link 3 is considered as a box with sides $b_3 \times b_3$, length L_3 and mass m_3. The matrix of inertia (in CM) for link 3 in L_3 is defined by:

$$\bar{D}_3 = \begin{bmatrix} I_3 & 0 & 0 \\ 0 & I_{3yy} & 0 \\ 0 & 0 & I_3 \end{bmatrix}, \quad \text{where } I_3 = I_{3xx} = I_{3zz}.$$

Link 4 (the high pressure unit) can be considered as an infinitely thin rod with mass m_4 and length a_4. The matrix of inertia in CM for link 4, described in L_4 is:

$$\bar{D}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_4 & 0 \\ 0 & 0 & I_4 \end{bmatrix}, \quad \text{where } I_4 = I_{4yy} = I_{4zz}.$$

•**Problem 7:** Find the following moments of inertia $I_1, I_{1yy}, I_2, I_{2zz}, I_3, I_{3yy}$ and I_4 , respectively, when

$$\begin{aligned} L_1 &= 0.67\,[\text{m}], r_1 = 0.04\,[\text{m}], L_2 = 1.7\,[\text{m}], b_2 = 0.22\,[\text{m}], \\ L_3 &= 1.65\,[\text{m}], b_3 = 0.18\,[\text{m}], a_4 = 0.98\,[\text{m}] \\ m_1 &= 4.9\,[\text{kg}], m_2 = 8.1\,[\text{kg}], m_3 = 4.9\,[\text{kg}], m_4 = 2.2\,[\text{kg}] \end{aligned}$$

(for use in *Problem 9*: $\Delta_2 = 0.34\,[\text{m}]$)

PART 5: ROBOT CONTROL AND DYNAMIC SIMULATION

MAXON DC-servo motors of type RE and gears with ratios $n = 53$ (53:1Maxon gear) are mounted in each of the joints of the manipulator for control of the joints $q = [\theta_1, \theta_2, d_3, \theta_4]^T$ (the prismatic joint 3 achieves the gear reduction ratio through a ball lead screw with steepness $n = 53\,[\text{rad/m}]$). Assume that there are no losses and backlashes in the gears (losses are not considered in this assignment at all). The masses of all the motors are also neglected.

•**Problem 8:** The actuator model is given by

$$u_i \cdot k_{T,i} - \frac{1}{n_i}\tau_i = J_{M,i} \cdot \ddot{q}_{M,i} + f_{M,i}\dot{q}_{M,i}, \quad i = 1, 2, 3, 4$$

where

u_i: control voltage [V].

$k_{T,i}$: torque constant [Nm/V].

τ_i : load torque (on the load side of the gear) .

n_i: gear ratio.

$J_{M,i}$: mass moment of inertia of the motor.

$f_{M,i}$: viscous friction coefficient for motor .

$q_{M,i}$: motor angle [radians].

Derive a SIMULINK block diagram for closed-loop motor angle control.

•**Problem 9:** To implement 'Independent Control' we need to determine the effective moments of inertia as seen from the motor axes. Find the effective moments of inertia "seen" from the motor axes when (use the values from *Problem 7*):

$$J_{\text{eff},i} = \frac{1}{n_i^2} \cdot \sup_q(D_{ii}(q)) + J_{M,i}, \quad i = 1, 2, 3, 4.$$

where

$$q = [\theta_1 \quad \theta_2 \quad d_3 \quad \theta_4]^T.$$

$\sup_q(D_{ii}(q))$ means determine the largest numerical value of each diagonal element ($D_{ii}(q)$) in the $D(q)$ matrix. i is index for link i. $n_i = 53$ is the gear reduction ratio, and $D(q)$ is the robots mass/inertia tensor (refer to page 1 of Annex B). J_M is the mass moment of inertia of the motor

$$J_M = 1320 \times 10^{-7}\text{kgm}^2.$$

(Please refer to Annex C for further description of the 'Independent Control' Principle)

For use in following problems:

$$g = 9.8\,[\frac{\text{m}}{\text{s}^2}], \quad k_T = \ 0.17\,[\frac{\text{Nm}}{\text{V}}],$$
$$f_{\text{eff}} = 2.4 \cdot 10^{-5}\,[\frac{\text{Nm}}{\text{rad/s}}], \quad n_i = 53.$$

where f_{eff} is effective viscous friction coefficient, which is defined by $f_{\text{eff}} = f_M + f_L/n_i^2$ (f_L is

viscous friction coefficient on the link side, here assume $f_L = 0$).
The following PD position control law is added to each robot axis

$$u_i = n_i \cdot [K_{p,i} \cdot (q_i^r - q_i) + K_{D,i} \cdot (\dot{q}_i^r - \dot{q}_i)], \quad i = 1, 2, 3, 4.$$

•**Problem 10:** Consider the actuator model in *Problem 8*, the 'independent control' in *Problem 9* and Annex C, and the PD control law above. Express the angle positions in the Laplace domain by the transfer functions:

$$q_i(s) = \frac{1}{N_i(s)} \left[F_i(s) q_i^r(s) - G_i(s) T_{L,i}(s)\right]$$

where all $f(s)$ are polynomials in the operator s.

•**Problem 11:** Show that the PD control parameters can be expressed as follows

$$K_{P,i} = \frac{\omega_{n,i}^2 \cdot J_{\text{eff},i}}{k_{T,i}} \qquad K_{D,i} = \frac{2\zeta_i}{\omega_{n,i}} \cdot K_{P,i} - \frac{f_{\text{eff},i}}{k_{T,i}}, \quad i = 1, 2, 3, 4$$

where ω_n and ζ are the natural frequency and the damping of the closed loop system, $s^2 + 2\omega_{n,i}\zeta_i s + \omega_{n,i}^2$. Be aware that a PD controller can not be applied in practice. (Study also a LEAD controller relative to compare with PD controller - only for those who are interested in LEAD controller)

•**Problem 12:** Derive a model in SIMULINK for each joint servo axis with $q_i^r(s)$ as input and $q_i(s)$ as output. Use the closed loop control described above with $T_{L,i} = 0$. Determine $K_{P,i}$ and $K_{D,i}$, when $\omega_{n,i} = 15\left[\frac{\text{rad}}{\text{s}}\right]$ and $\zeta_i = 1$. Simulate the closed loop model response of a reference step at the size $q_i^r = 0.34\,[\text{rad}]$ for $i = 1, 2, 4$ and $q_3^r = 0.34\,[\text{m}]$ posed at time $t = 0\,[\text{s}]$.
Plot q_i^r, q_1, q_2, q_3 and q_4 as a function of time t, on the same plot, denoted A_1 (use MATLAB's 'hold' command). Make comments on the results. Simulate the closed loop system for each joint axis at $\zeta_i = 0$ by use of the same reference step as used for plot **A1** above, and show q_i^r, q_1, q_2, q_3 and q_4 on the same plot, **A2**, as a function of time. Comment the results.

•**Problem 13:** The masses of the manipulator are now introduced to the model by means of a load torque T_L.

Please determine the joint values when the torque/force load T_L is maximum for each joint (use the $h(q)$ matrix in Annex B), and give the relevant configuration of the robot arm. Add these constant load torques T_L to the simulation model. Simulate a response of a reference step input, $q_i^r = 0.35$, damping ratio $\zeta_i = 1.0$ and natural frequency $\omega_{n,i} = 15$. On the same plot (**A3**), show q_i^r, q_1, q_2, q_3 and q_4 as a function of time. Make comments on the results. Determine the steady-state error (calculate and compare).

•**Problem 14:** For each robot-axis, introduce a PID controller of the type (analogue to *Problem 10*)

$$u_i = n_i \cdot \big[K_{p,i} \cdot (q_i^r - q_i) + K_{D,i} \cdot (\dot{q}_i^r - \dot{q}_i) + K_{I,i} \int (q_i^r - q_i) dt\big], \quad i = 1,2,3,4$$

•**Problem 15:** Find $K_{P,i}$, $K_{D,i}$, $K_{I,i}$ expressed by the natural frequency ω, damping ζ and the time constant τ' of the closed loop system $(s^2 + 2\zeta_i\omega_{n,i}s + \omega_{n,i}^2)(1 + \tau_i' s)$ (use the results from *Problem 14* and put $\tau_i' = |J_{\text{eff},i}|$ (in seconds)).

Determine $K_{P,i}$, $K_{D,i}$, $K_{I,i}$ at $\omega_{n,i} = 15\,[\text{rad/s}]$ and $\zeta_i = 1$ for $i = 1,2,3,4$. Simulate a reference step response, $q_i^r = 0.35\,[\text{rad}]$ or $[\text{m}]$. Plot q_i^r, q_1, q_2, q_3 and q_4 as a function of time on the same plot (**A4**). Make comments on the results. Also comment the overshoot (if any). Determine the steady state error (calculate and compare).

•**Problem 16:** Add the following pre-filter to the simulation model:

$$F_{P,i}(s) = \frac{1}{\frac{K_{D,i}}{K_{I,i}}s^2 + \frac{K_{P,i}}{K_{I,i}}s + 1}, \quad i = 1,2,3,4$$

As shown on Figure 2.
Simulate a reference step response. Plot the results on the same plot (**A5**), as a function of time. Make comments on the result. Draw a symbolic block diagram (no numeric values) for the closed loop system for link 1 including the control-object, the PID-controller, and the pre-filter (plot **A6**).

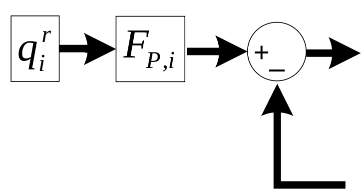


Figure 2: Prefilter

•**Problem 17:** Add the SIMULINK block named Bane Generator (path generator), and the Block named Direkte kinematik (*Direct kinematics*) to model (see Annex D, use SIMULINKS *demux* and *mux* blocks). Add the GM-matrix (found in *Problem 4*) to MATLABs workspace. The GM-matrix is a 16×6 matrix: $\mathrm{GM}(i,j) = a_{i,6-j}$, $i = 1, 2, ..., 16, j = 1, 2, ..., 6$. Execute the m-file with the initializing constants. Simulate now the point to point movement (the duration time is 8 sec.) (*NB! Without the pre-filter*).
Plot q_i^r, $q_i = 1, 2, 3, 4$ as a function of time on the same plot (**A7**).
Show p_y as a function of p_x on plot **A8**. Make comments on the results.
The movements of the robot manipulator can be animated by the inclusion of the block named *Gem til vis4link (save to show 4link)* into the simulation model (see Annex D).

•**Problem 18:** Concerning the inverse kinematic transformer: Add to the simulation model the module named *inverse kinematik* (inverse kinematics) from the robot library named robtek1 (see Annex D). Add the GM-matrix found in *Problem 5* to MATLABs workspace. The GM-matrix is a 8×6 matrix: $\mathrm{GM}(i,j) = a_{i,6-j}$, $i = 1, 2, ..., 8, j = 1, 2, ..., 6$. Execute the m-file with initializing constants.

Thereafter, simulate the total movement in the Cartesian space (duration 8 seconds). Please plot the wanted Cartesian path and the measured Cartesian path (p_y as a function of p_x) on the same plot (**A9**). Make comments on the results.

We will now include an "exact" model of the robot dynamics in the simulation model. Note, that this "exact" model is ideal and does not include dry friction, backlash, etc. One way to do this is to include the part of the robot model dynamics, which is not included as a part of the effective inertia as a load torque T_L . However, a more sufficient way to do this is to amalgamate the manipulator dynamics:

$$\begin{aligned} &D(q) \cdot \ddot{q} + v(q, \dot{q}) = \tau = \begin{bmatrix} \tau_1 \ \tau_2 \ f_3 \ \tau_4 \end{bmatrix}^T \\ &\text{where } q = \begin{bmatrix} q_1 \ q_2 \ q_3 \ q_4 \end{bmatrix}^T \end{aligned} \tag{1}$$

With the DC-actuator models:

$$u_i \cdot k_{T,i} - \frac{1}{n_i} \cdot \tau_i = J_{M,i} \cdot \ddot{q}_{M,i} + f_{M,i} \cdot \dot{q}_{M,i}, \quad i = 1, 2, 3, 4 \tag{2}$$

•**Problem 19:** Show that equation (1) and (2) can be reduced to:

$$\widetilde{D}(q) \cdot \ddot{q} + \widetilde{v}(q, \dot{q}) = u \tag{3}$$

where $\widetilde{D}(q)$ is a 4x4 matrix, $\widetilde{v}(q, \dot{q})$ is a 4x1 vector and $u = [u_1 \ u_2 \ u_3 \ u_4]^T$ is a 4x1 vector.

A PID-controlled robot system with the "exact" model and a square generator are included in the SIMULINK file named *trc_dyn.mdl*.

NB! In this exact model, the gravity load is dependent of the joint variables as shown in, $h(q(t))$ (refer to Annex B). Consequently, the gravitational load is no longer approximated by the constant "worst-case" load torque T_L , as found in *Problem 13*.

•**Problem 20:** Add the control parameters determined in *Problem 15* and the GM-matrix used in *Problem 18* to MATLAB's *workspace*. Execute the m-file with initializing constants. Simulate the movements. Plot the wanted Cartesian path and the measured Cartesian path (p_y as a function of p_x) on the same plot (**A10**). Compare the results with the results obtained in *Problem 18*.

Problem 21: Add the control parameters determined in *Problem 11*. Simulate without using the integration control (that is, by using a PD controller). On the same plot (**A11**), plot the wanted Cartesian path and the measured Cartesian path (p_y as a function of p_x). Compare the results with the results obtained in *Problem 20*.

Annex A: Kinematic description of the Alto robot used in the project

The figure below shows a sketch of the robot used in this project, as well as a table of Denavit-Hartenberg parameters corresponding to the robot. The robot has three revolute joints and one prismatic joint; that is a so-called RRPR-robot (also refer to next page).

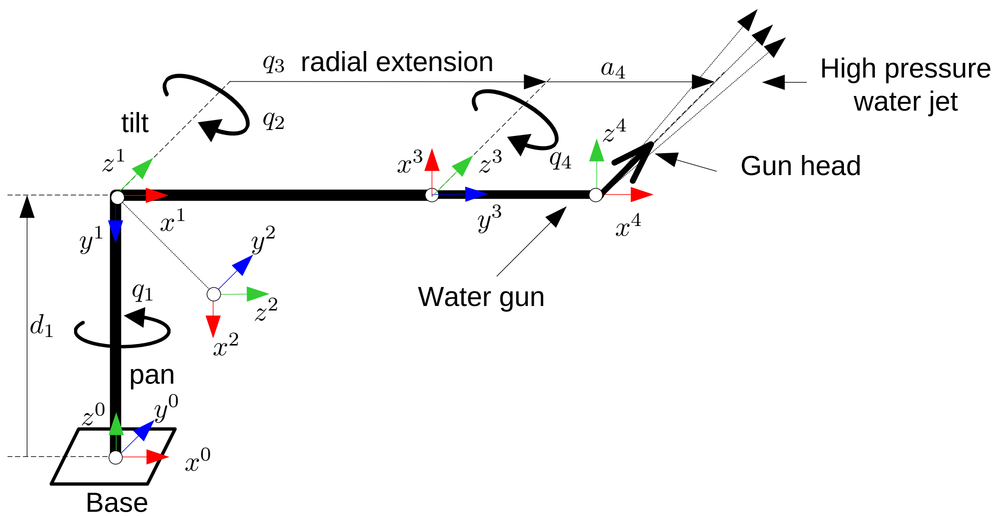
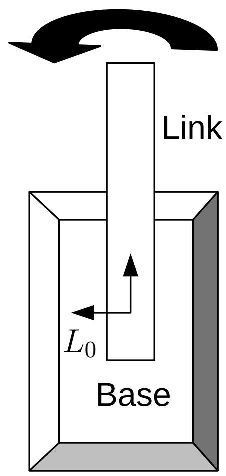


Figure 3: The RRPR-robot used in the project assignment.

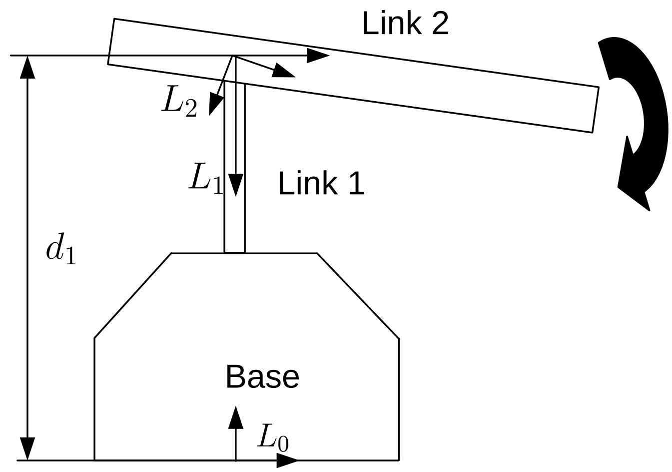
Joint	type	a	α	d	θ	initial	min	max
1							$-\pi$	π
2							$\pi/6$	$3\pi/4$
3						1.35[m]	1.35[m]	3.00[m]
4							$-\pi/2$	$5\pi/4$

The DH- table for the RRPR-robot, where $\mathbf{d_1 = 1.5}$ [**m**], $\mathbf{a_4 = 1.02}$ [**m**].
Gun head coordinates with respect to L_4 : $\left[\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}\right]^T$ [m].

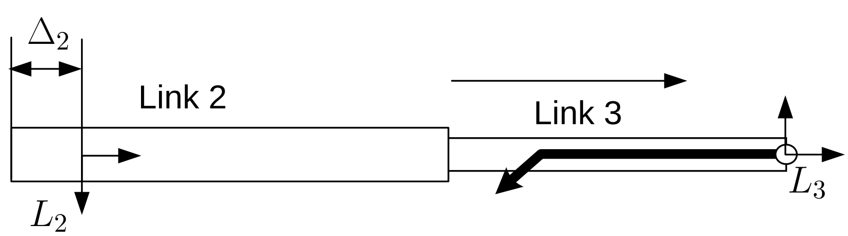
Kinematic description of the Alto robot (continued)



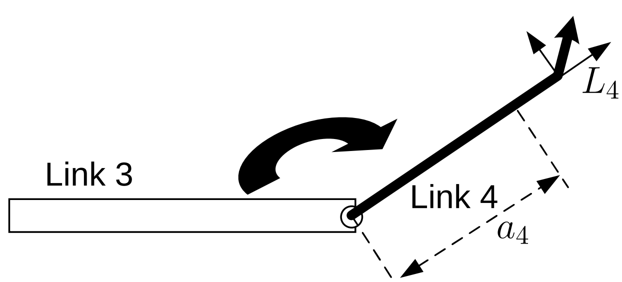
1: Rotation of tower



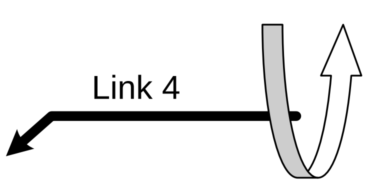
2: Rotation of arm



3: Radial extension of arm



4: Rotation of high pressure gun



5: Rotation of gun head (not addressed in the project)

Annex B. Dynamics of the Alto robot

The inverse dynamic equation related to the Alto robot can be written:

$$\tau = D(q)\ddot{q} + c(q,\dot{q}) + h(q)$$

Where $c(q,\dot{q})$ is the inter-axis velocity coupling vector which includes Coriolis and centrifugal forces (not addressed further in Annex B).
$D(q)$ is the symmetric manipulator inertia tensor:

$$D(q) = \begin{bmatrix} D_{11} & 0 & 0 & 0 \\ 0 & K_3 + 2f_1(q_3) + f_2(q_3)\sin(q_4) & f(q_4) & 2K_1 + \frac{1}{2}f_2(q_3)\sin(q_4) \\ 0 & f(q_4) & K_4 & f(q_4) \\ 0 & 2K_1 + \frac{1}{2}f_2(q_3)\sin(q_4) & f(q_4) & 2K_1 \end{bmatrix}$$

with

$$D_{11} = K_5 + f_1(q_3) - (K_2 + f_1(q_3))\cos(2q_2) + K_1\cos(2(q_2+q_4)) - f_2(q_3)\cos(q_2+q_4)\sin(q_2)$$

and

$$\begin{aligned} K_0 &= m_2(\frac{1}{2}L_2 - \Delta_2)^2 + m_3(\frac{1}{2}L_3)^2 \\ K_1 &= \frac{1}{2}\big[I_4 + \frac{1}{4}m_4a_4^2\big] \\ K_2 &= \frac{1}{2}\big[I_2 - I_{2zz} + I_3 - I_{3yy} + K_0\big] \\ K_3 &= I_2 + I_3 + K_0 + 2K_1 \\ K_4 &= m_3 + m_4 \\ K_5 &= \frac{1}{2}\big[2I_{1yy} + I_{2zz} + I_{3yy} + K_3\big] \\ f_1(q_3) &= \frac{1}{2}K_4q_3^2 - \frac{1}{2}m_3L_3q_3 \\ f_2(q_3) &= m_4a_4q_3 \\ f(q_4) &= \frac{1}{2}a_4m_4\cos(q_4) \end{aligned}$$

$h(q)$ is the gravity load vector:

$$h(q) = g\begin{bmatrix} 0 \\ \frac{1}{2}m_4a_4\cos(q_2+q_4) + [m_2(\Delta_2 - \frac{1}{2}L_2) + m_3(\frac{1}{2}L_3 - q_3) - m_4q_3]\sin(q_2) \\ (m_3+m_4)\cos(q_2) \\ \frac{1}{2}a_4m_4\cos(q_2+q_4) \end{bmatrix}$$

Δ_2 is given in Problem 7.

Annex C: Independent joint control

Independent joint control is a well-proven control principle for robot control. By this kind of control, the control signal for link i is generated exclusively by considering the position, (and sometimes also the derivative of the position for link i i.e. the speed of link i), that is $u_i = u_i(q_i)$.

The independent joint control is often implemented by the use of PD and PID control combined with a proper pre-filtering and feed-forward control. These classic control concepts have proven to be sufficient for use with large gear reduction ratios whereby the nonlinear and coupled structure of the robot become of minor importance.

There exist methods to handle the nonlinear closed control loop systems (e.g. Lyaponov functions, hyper stability). However, these methods for handling the non-linearities are typically concentrated on the stability of the system rather than on the system performance.

A very much-applied method is to use classic linear single input single output control theory (known from a basic control theory courses) on a linearized or otherwise simplified control object.

A simplified control scheme will be examined in the following. The dynamics model of the robot manipulator can be described by the following equation:

$$D(q)\ddot{q} + C(q, \dot{q}) + h(q) = \tau$$

This equation may be rewritten to:

$$\hat{D}(q)\ddot{q} = \tau + (\hat{D}(q) - D(q))\ddot{q} - C(q, \dot{q}) - h(q) = \tau - T_L$$

where

$$\hat{D}(q) = \mathrm{diag}(I_{1\max}, \ldots, I_{m\max})$$

is a constant diagonal matrix determined by

$$I_{i\max} = \sup_q(d_{ii}),$$

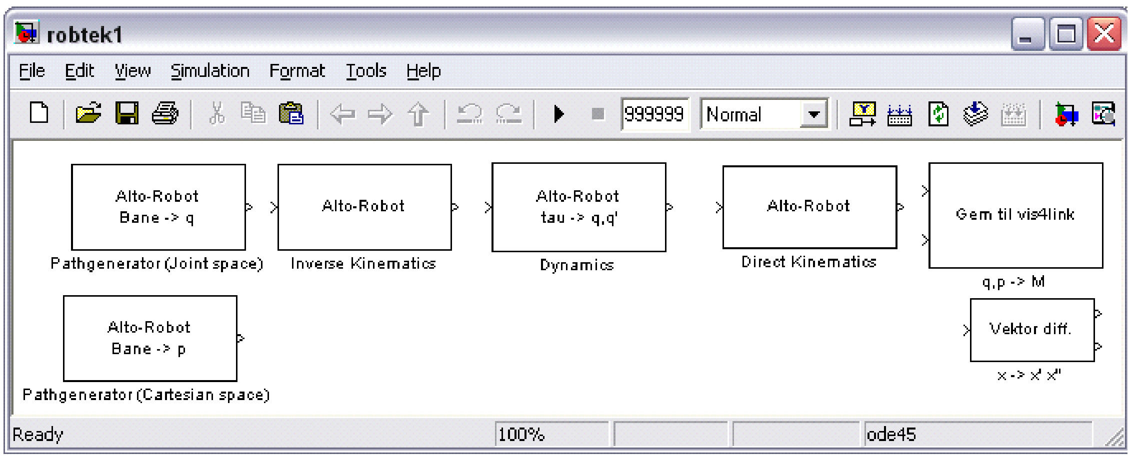
$I_{i\max}$ is the largest moment of inertia for joint axis . A critical damped or over-damped closed loop system is secured which is desirable for robot control by choosing these maximum moments of inertia. Critically damped or over-damped closed loop systems are preferred for robot control because overshoots cannot be accepted.

T_L can be considered as a disturbance and is considered independent of q when used in simple stability analysis.

Annex D

Description of the Robot Library for solving the questions 15-19.

There have been developed a number of blocks, which provide the possibility to simulate kinematics, dynamics and trajectory generation for the four-link Alto robot in Matlab. These blocks are stored as Matlab m-files located in the '31383 Robotteknik' download area at the DTU Campus net. Therefore, you can use an Internet browser to download these m-files from the DTU Campus net to your local computer.
When you write **robtek1** in the MATLAB command window you will get access to the following window.



The individual blocks exchange data denoted p and q. p is a 6x1 vector containing the origin, $p^4 = \left[[p]_1^4\ [p]_2^4\ [p]_3^4\right]^T$ and x-axis, $x^4 = \left[[x]_1^4\ [x]_2^4\ [x]_3^4\right]^T$ of the Cartesian tool frame, (i.e. the first three rows of the first and the last column of the transformation matrix). Similarly, q is a 4x1 vector containing the robot joint coordinates q_1, q_2, q_3 and q_4.

NOTE: In order to run the Robot library with your data for the Alto robot, you must include the following global statement in your initializing m-script file and use these global names for your variables and constants:

global $GM\ m1\ m2\ m3\ m4\ L1\ L2\ dd2\ L3\ a4\ I2\ I3\ I4\ I1yy\ I2zz\ I3yy\ g\ n\ k_T\ f_{eff}\ d_1$;

The individual blocks are described in the subsequent paragraphs. $dd2$ is Δ_2 (given in Problem 7).

Pathgenerator (Joint space planner).

The 'Pathgenerator' (Joint space) block generates joint space references as four to-seconds segments. For this purpose the Matlab function J_path defined in the script-file J_path.m is called by the Pathgenerator.

The references in the individual segments are generated from fifth order polynomials (one polynomial per axis). Input to the 'Pathgenerator' block is the GM matrix containing the

4×4 polynomials. The first four rows is the first segment, the next four rows is the second segment, etc. (The time starts at zero at the first segment):

$$\begin{aligned} & f_1(t) = a_{15}t^5 + a_{14}t^4 + a_{13}t^3 + a_{12}t^2 + a_{11}t + a_{10} \\ & f_2(t) = a_{25}t^5 + a_{24}t^4 + a_{23}t^3 + a_{22}t^2 + a_{21}t + a_{20} \\ & f_3(t) = \dots \\ \Rightarrow\ & GM = [a_{15}\ a_{14}\ a_{13}\ a_{12}\ a_{11}\ a_{10}; \\ & \qquad a_{25}\ a_{24}\ a_{23}\ a_{22}\ a_{21}\ a_{20}; \\ & \qquad \dots \end{aligned}$$

Pathgenerator (Cartesian space planner).

The 'Pathgenerator' (Cartesian space) block generates Cartesian space references as four to-seconds segments. For this purpose the Matlab function C_path defined in the script-file C_path.m is called by the Pathgenerator.

The references in the individual segments are generated from fifth order polynomials (one polynomial per coordinate). Input to the 'Pathgenerator' block is the GM matrix containing the 4×2 polynomials. The first two rows is the first segment, the next two rows is the second segment, etc. (The time starts at zero at the first segment):

$$\begin{aligned} p_X(t) =& f_1(t) = a15 \cdot t^5 + a14 \cdot t^4 + a13 \cdot t^3 + a12 \cdot t^2 + a11 \cdot t + a10 \\ p_Y(t) =& f_2(t) = a25 \cdot t^5 + a24 \cdot t^4 + a23 \cdot t^3 + a22 \cdot t^2 + a21 \cdot t + a20 \\ p_X(t) =& \dots \text{etc.} \\ GM =& [a_{15}\ a_{14}\ a_{13}\ a_{12}\ a_{11}\ a_{10}; \\ & a_{25}\ a_{24}\ a_{23}\ a_{22}\ a_{21}\ a_{20}; \\ & \dots \\ & \text{etc.}] \end{aligned}$$

Inverse kinematics

This block calculates the inverse kinematics for the revolute robot. Inputs are the Cartesian coordinates p_X, p_Y, and p_Z, as well as the x^4 vector, and output are the four corresponding joint coordinates.

Dynamics

This block provides a facility for simulating the dynamics behaviour of the robot manipulator. Input to the block is a vector containing the wanted armature currents (danish: ankerstrømme) u_1, u_2, u_3 and u_4. Output are q and $\dot{q}$. The dynamic calculations are carried out by the Matlab function *id4link* defined in the script-file *id4link.m*.

Direct kinematics

This block calculates the direct kinematics. Input is q, and outputs are the p^4 and x^4 vectors.

Gem til vis4link (save to show4link).

This block saves the simulation results in a matrix M. This matrix is used when the function **vis4link** is used. **vis4link** shows a 3D animation of the of the robot movements.

Enter **help vis4link** in the MATLAB window for instructions on how to use the function. Example: Try **vis4link(M,1,18)**

<u>Vektor diff.:</u> Block for calculation of vector differentiation, not used in the project.