Assignment no. 1

http://www.cs.tau.ac.il/~danha/courses/rob02.html

due: April 23rd, 2002

You may submit this assignment in pairs.

The letter **(p)** after the exercise number indicates that this exercise has a programming component.

Exercise 1.1 (p) Write a program that solves Oskar's cube. In the course's website you will find the input matrices that describe the faces of the cube. You will also find there the specifications how to output a solution path from start to goal such that the output of your program can then be run by the graphical program at the site—this way you can verify the correctness of your solution.

Exercise 1.2 (p) Solve the inverse kinematics problem for a planar arm with three rotational joints. Assume that each joint can freely rotate regardless of the location of the other joints. Also each joint is free to attain any orientation (360°).

A tool is attached to the last link of the robot. The tool frame F has its origin at the tip of the last link and its orientation is the angle α that the extension of the last link makes with the positive direction of the x-axis. See Figure 1 for an illustration. A placement F_i of the tool frame is specified by three parameters (x_i, y_i, α_i) where (x_i, y_i) is the desired location of the tool frame's origin and α_i is the desired orientation.

Write a procedure linear_motion that plans a trajectory of the tool along a straight line segment in joint values. The input to the procedure is the links' lengths l_1, l_2, l_3, the start and goal frames F_1 and F_2 and a "resolution parameter" ε. The output of the procedure is a sequence of joint values $J_1, J_2, \dots, J_n$ (each J_i is a triple of joint values $(\theta_1^i, \theta_2^i, \theta_3^i)$) that will move the robot from F_1 to F_2. (In particular in configuration J_1 the tool frame is at F_1 and in J_n the tool frame is at F_2.) The sequence should be computed such that $|\theta_k^{i+1} - \theta_k^i| < \varepsilon$ for all $1 \leq k \leq 3$, $1 \leq i < n$. All angles will be given in radians.

At the course's site you'll find the source code of an interactive program where you should insert the procedure linear_motion. Also there you'll find the exact specification of the data types to be used.

Notice: Those of you who have not studied Computational Geometry are advised to read Chapters 1 and 2 in the book *Computational Geometry Algorithms and Applications* by M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, Springer. Already in class on Tuesday 23/4 it will be assumed that you are familiar with convex hulls, the doubly-connected-edge-list (DCEL), and the sweep algorithm for computing the intersection points of a collection of segments in the plane. In the website there are suggestions for further reading.

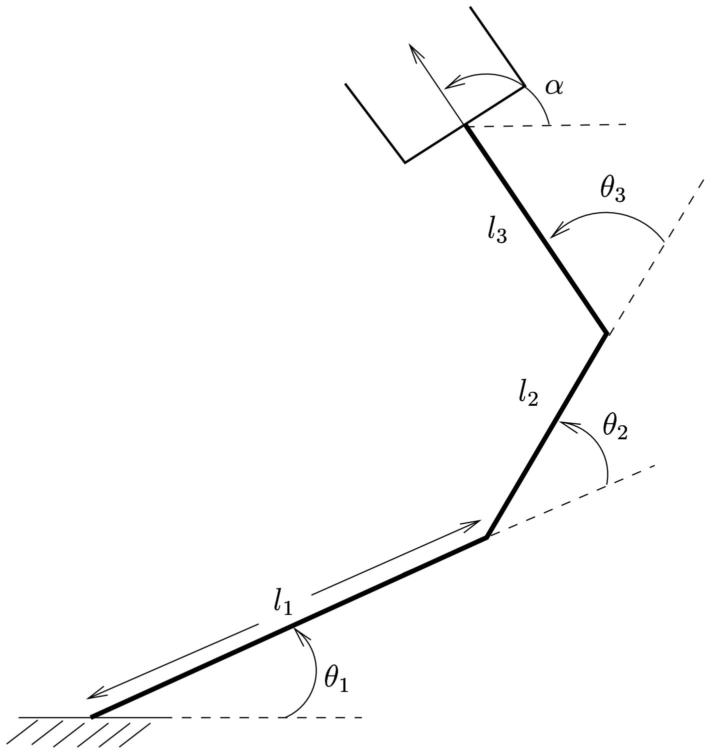

Figure 1: The three-link arm