MECH 577 Optimum Design

Project # 2: Unconstrained Optimization
The Maximum Reach of a Three-Axis Robot

Assigned: September 26, 2006 Due: October 24, 2006

We retake here Example 5.3.1 of the Lecture Notes (LN), in which the maximum reach of a three-axis robot is found via an equality-constrained optimization problem. The maximum reach is defined, with reference to Fig. 5.1 of the LN, as the maximum distance of point C from the Z_1-axis.

Some background work is needed here: First, an expression for the position vector $\mathbf{c}$ of Point C of the end-link is derived. Using this expression, the position vector $\mathbf{c}'$ of the projection C' of C onto the X-Y plane is found. The distance of point C from the Z_1 axis is the Euclidean norm of this vector. You needn't know robotics to find an expression for vector $\mathbf{c}'$. Just use the Maple worksheet `RRRorthoBkgd.mws` that is available at the course website.

(a) It is apparent from `RRRorthoBkgd.mws` that the problem reduces to finding the minimum of function $f(\theta_2, \theta_3)$, which does not involve θ_1 as an argument. The reason behind is that the maximum reach is independent of θ_1 by virtue of the axial symmetry of the robot workspace. Set up the normal equations of $f(\theta_2, \theta_3)$. Now you can find all stationary points of this function *by inspection*: To this end, plot the two functions derived from the normality conditions as implicit functions of θ_2 and θ_3 on the θ_2-θ_3 plane. These are two contours whose intersections give you all stationary points, i.e., all pairs of values (θ_2, θ_3) that verify the normal equations. These values can be accurate to a couple of digits only, but this accuracy may be all you need in engineering practice. With these rough estimates of the foregoing pairs of values, determine the nature of each stationary point by means of the second-order normality conditions. Moreover, with the aid of a 3D plot of the objective function you can find the desired maximum. Give a rough estimate of the maximum reach and compare it with the value found in Example 5.3.1.

(b) The above rough estimates can be refined to 16 digits by means of nonlinear-equation solving. Indeed, the two normal equations can be solved for the design variables θ_2 and θ_3 using the Newton-Raphson (NR) method[1]. In doing this, try to use as much information and insight as you can, by using the results of item (b): Use the rough estimates as initial guesses in the NR method and the Hessian of f as the Jacobian in the NR method. Verify the nature of each stationary point.

(c) As a means to avoid the normal equations—these can lead to wrong results in more complex problems—try a quasi-Newton method. To this end, write a piece of code either in C or in Matlab, and implement, e.g., the Broyden-Fletcher-Goldfarb-Shanno method; if the implementation of this method prevents you from submitting your report on time, try a gradient method, e.g., the one based on the Fletcher-Reeves algorithm. Verify your results using Matlab's implementation of the Nelder-Mead method.

[1]Not that this approach is sound. Remember that the normal equations are a source of ill-conditioning and the NR method requires second-order partial derivatives.