Reverse Kinematics on a General Four-Bar Linkage using a Least Squares Curve Fit Algorithm–A Work In Progress

Joshua Holbrook

November 19th 2008

The Idea

Using the analytical equations for a four-bar linkage isn't particularly difficult going forwards. Going backwards–finding a set of linkage parameters that result in a four-bar able to trace a required path–is harder.

There are many ways to do this. For example, I've heard legend of an algebraic approach similar to Gaussian Elimination that can be used to solve the general problem when the trigonometric functions are substituted for variables, then constrained by the trigonometric identities. For something as simple as a four-bar, this should work fine.

There are also a fair number of numerical approaches. One, which I don't think is particularly common but has been used (I did find an example for a robot arm), is to solve a related curve fitting problem:

$$\min \sum_{i=1}^{n} \left|\left| \begin{pmatrix} \bar{p} \\ \bar{\gamma} \end{pmatrix}_i - \begin{pmatrix} p(L, \theta_i) \\ \gamma(L, \theta_i) \end{pmatrix} \right|\right|^2$$

where the linkage dimension/configuration parameters L and all the θ_i are free, and $\bar{p}_i$ and $\bar{\gamma}_i$ are points one wants the curve to go through. This ends up being a sort of least squares curve fitting problem, and can be solved using the Gauss-Newton method assuming the constraints on θ never come into play.

One advantage of using a curve fit over other methods is that, even if there isn't a possible exact solution, the problem should still converge on a

minimum that is, by a measure (likely a familiar measure), the closest path to all the points.

Currently Unresolved Issues

Will This Really Work?

Even for a relatively simple application of curve fitting, this seems very complex to me, and I wouldn't be surprised if I missed a way in which this could "break."

Dealing with constraints

Unconstrained optimization is fairly easy. Constrained optimization is a much harder problem, especially when the equations involved become non-linear. I *do* have constraints to deal with, and as of yet I'm not sure how. One of my ideas is to use a penalty function–it is this approach that I will likely adopt.