Inverse Kinematics

Forward Kinematics solves the problem “If I know the link transformation parameters, where are the links?”.

Inverse Kinematics (IK) solves the problem “If I know where I want the links to be (X*, Y*), what link transformation parameters will put them there?”

Cyclic Coordinate Descent (CCD) Method

The idea is to change Θ1 so that (X, Y) are as close to (X*, Y*) as possible.
Then change Θ2.
Then change Θ3.
Then change Θ1.
Then change Θ2.
Then change Θ3.
Then change Θ1.

...
Changing $\Theta_1$

Holding $\Theta_2$ and $\Theta_3$ constant, rotate $\Theta_1$ towards $(X^*, Y^*)$ so that the dashed purple lines line up.

Ground

$(X, Y)$

$(X^*, Y^*)$

Changing $\Theta_2$

Holding $\Theta_1$ and $\Theta_3$ constant, rotate $\Theta_2$ towards $(X^*, Y^*)$ so that the dashed purple lines line up.

Ground

$(X, Y)$

$(X^*, Y^*)$
Holding $\theta_1$ and $\theta_2$ constant, rotate $\theta_3$ towards $(X^*, Y^*)$ so that the dashed purple lines line up.

Now, do it again -- Changing $\theta_1$
Now, do it again -- Changing \( \Theta_2 \)

\[ (X, Y) \rightarrow (X^*, Y^*) \]

Now, do it again -- Changing \( \Theta_3 \)

\[ (X, Y) \rightarrow (X^*, Y^*) \]
Where we are now: \((X_3, Y_3)\)

Where we want to be: \((X_*, Y_*)\)

Computing how much to change a rotation by

\[
\theta_* = \arctan2(Y_* - Y_2, X_* - X_2);
\]

\[
\theta = \arctan2(Y_3 - Y_2, X_3 - X_2);
\]

\[
\Delta \theta_2 = \theta_* - \theta;
\]

Use the C/C++ \(\arctan2()\) function:

Do not use the C/C++ \(\arctan()\) function:

\[
\theta_* = \arctan\left(\frac{Y_* - Y_2}{X_* - X_2}\right);
\]

\[
\theta = \arctan\left(\frac{Y_3 - Y_2}{X_3 - X_2}\right);
\]

\[
\Delta \theta_2 = \theta_* - \theta;
\]