UNITS - USE CM!

Base rotation control code.

calibration - add a bump switch that gets hit when it passes origin, and doesn't limit to the rotation of the base. most of the base rotation code still applies.

UPDATE - ALL BUT ONE DOF IS DONE!
I STILL NEED THE WRIST - that's not too hard...

MOVING THE ARM:

TASKS:

find x — STEP 1 - CALCULATE ~~HOW FAR THE ARM~~ WHERE THE ARM NEEDS TO BE.

find Δx — STEP 2 - CALCULATE HOW FAR AWAY THOSE POSITIONS ARE FROM PROPER (DELTA'S)

apply $x - \Delta x$ — STEP 3 - ~~CALCULATE~~ MOVE ~~AR~~ TO THE RIGHT POSITIONS.

STEP 1

needed: desired height & length - units?

inputs - motor structures, with encoders parts, current degrees, & ~~is~~ arm lengths.

(math transferred from a different page)

$$MA = \tan^{-1}\left(\frac{height}{length}\right) + \cos^{-1}\left(\frac{b^2 + height^2 + length^2 - a^2}{2 \cdot b \cdot \sqrt{height^2 + length^2}}\right)$$

(in Degrees)

$$TARGET\,MA = \tan^{-1}\left(\frac{target\,height}{target\,length}\right) + \cos^{-1}\left(\frac{(shoulder\,length)^2 + target\,height^2 + target\,length^2 - (elbow\,length)^2}{2 \cdot shoulder\,length \cdot \sqrt{\text{height}\ target\,height^2 + target\,length^2}}\right)$$

CURRENTMA = getCurrentAngle(~~shoulder~~ base(lift); // needs calibration first.

$$MB = \tan^{-1}\left(\frac{length}{height}\right) + \cos^{-1}\left(\frac{height^2 + length^2 + a^2 - b^2}{2 \cdot a \cdot \sqrt{height^2 + length^2}}\right) \quad + (elbow\,length)^2$$

$$TARGET\,MB = \tan^{-1}\left(\frac{target\,length}{target\,height}\right) + \cos^{-1}\left(\frac{target\,height^2 + target\,length^2 + \text{(shoulder length)}^2 - (shoulder\,length)^2}{2 \cdot shoulder\,length \cdot elbow\,Length}\right)$$

CURRENTMB = getCurrentAngle(elbow lift); // needs calibration first - uses data & returns current angle from relative origin.

$$TARGET\,ANGLE\,C = \cos^{-1}\left(\frac{elbow\,length^2 + shoulder\,length^2 - (height^2 + length^2)}{2 \cdot elbow\,length \cdot shoulder\,length}\right)$$

(the elbow)

CurrentAngleC = getCurrentAngle(elbow lift) // needs calibration first...