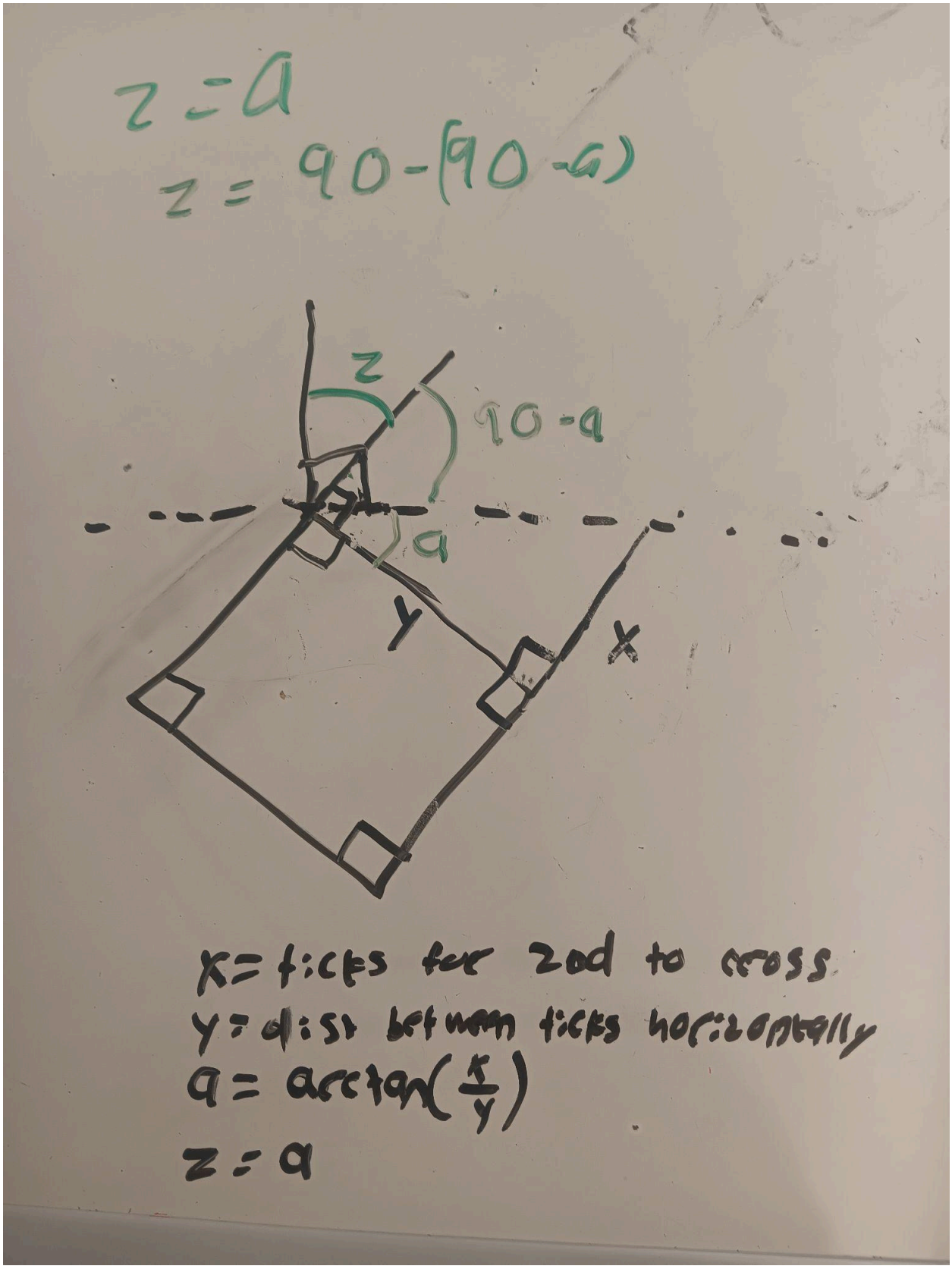
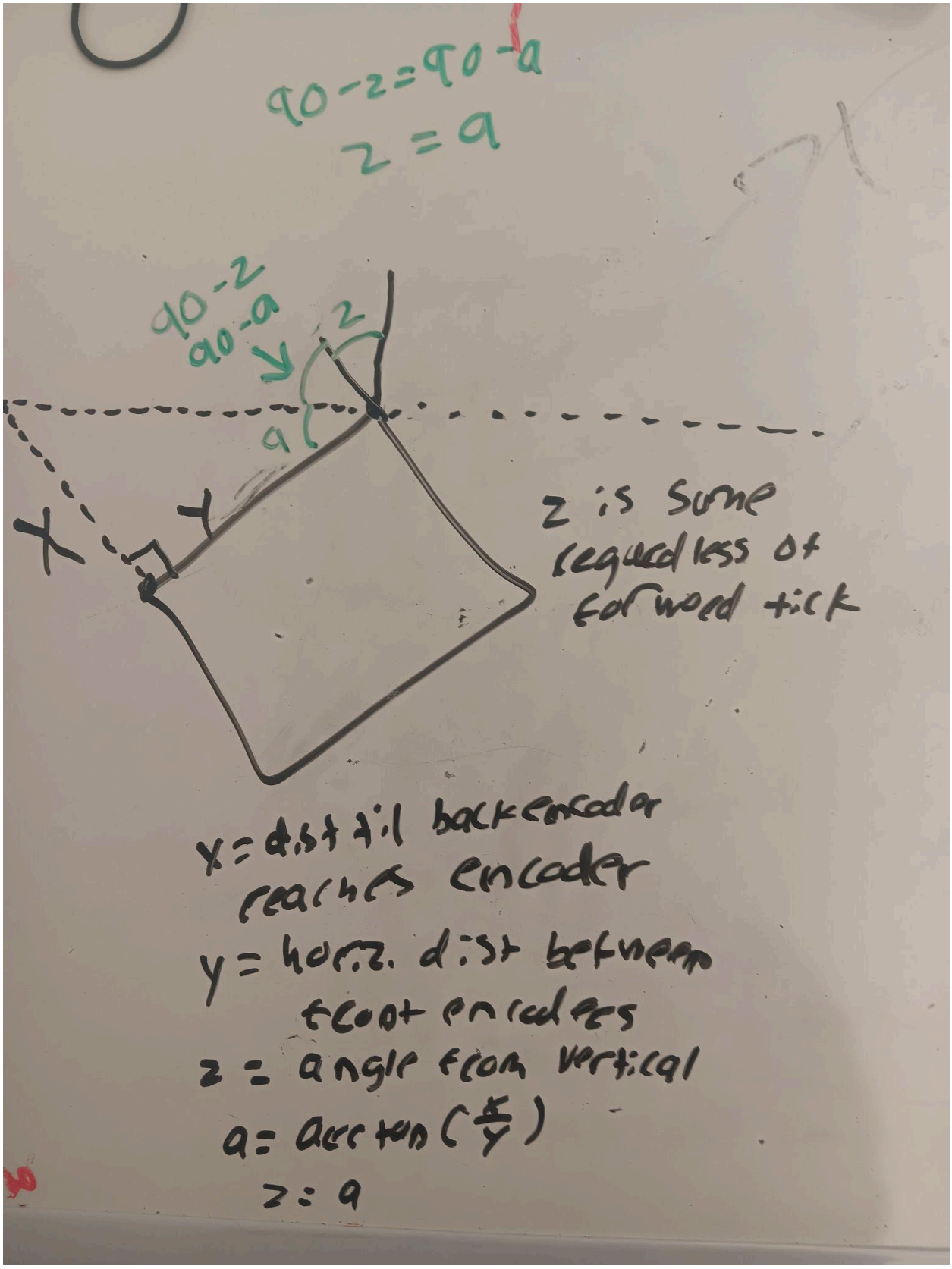

z = a
z = 90 - (90 - a)
z
90 - a
a
y
x
x = ticks for 2nd to cross
y = dist between ticks horizontally
a = arctan(x/y)
z = a


90-z=90-a
z=a
90-z
90-a
z
a
X
Y
z is same regardless of forward tick
x = dist til back encoder reaches encoder
y = horiz. dist between front encoders
z = angle from vertical
a = arctan(x/y)
z = a

- So, I'll need to first get Y, put that in for the current bot. Should be in meters.
- Next, getting X will be the issue. For now, without the PID, we'll probably map x to be the number of ticks until x reaches the new color, and maybe hope there's a linear relationship between that and distance traveled, and then do numTicks * multiplier = distance.
- Later, if the PID allows us to get distance between any given points, we can just get it by measuring when the first encoder hits and then when the second encoder hits.
- If the PID can give current velocity, we can wait for time until we reach the edge, and do dist = velocity * time. This assumes velocity will be constant, which it should ideally be.
- Then, we can get z using what's provided above. That'll just be the angle offset we'll have to compensate for.