CORRECTIONS SO THAT ROBOT MOVES IN A STRAIGHT LINE

One of the main focus of this project is to find a efficient solution so that our bot moves in a straight line with minimal error. Here we have tried to do it without using encoders, kangaroo. Thus, only using MPU (for angle and displacement measurements) and sabertooth for motor voltage controls.

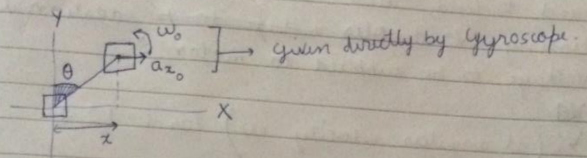
In this report I have devised a simple model and then implemented it in a code so that our bot moves in a straight line using ICU only.

Regarding our progress on the ICU front, we have learnt its use, and successfully calibrated it (thanks to Tanmay sir!). Today we'll try to complete the chassis and mount motors.

So below is the model I am proposing.

Suppose our bot was moving in a straight line along y-axis with no velocity or acceleration along x-axis and no angular velocity or acceleration(moving ideally).Now, suppose after sometime the bot starts moving with Ax=Ao and Wz=Wo. Firstly, we have calculated its displacement along x-axis and the angular deviation from y-axis. Rest is explained below.

→ We want the bot to again come to $\theta = 0$ and $v_x = 0$ in short duration of time, so that we do not miss readings in between.



→ Assuming constant acceleration and ω.

Now, we will rectify by giving V_{x_0} and $-a$ and ω_0 and $-\alpha$. Let us assume bot comes to its original position in t time. During this time we would assume that no external force acts on the bot.

So, $\theta = \frac{1}{2}\alpha t^2$ -(1)

$x = \frac{1}{2} a t^2$ -(2)

$V_x - V_{x_0} = at$ -(3)

$\omega - \omega_0 = \alpha t$ -(4)

We keep ω and α fixed, and find v and a.

Using (1) and (2),

$$\frac{\theta}{\alpha} = \frac{x}{a} \quad -(5)$$

and, using (3) and (4),

$$\frac{V_x - V_0}{\omega - \omega_0} = \frac{a}{\alpha} \quad -(6)$$

By solving we get,

$$\boxed{V_x = \frac{x(\omega - \omega_0)}{\theta} + V_0}$$

and, $\boxed{a = \frac{\alpha x}{\theta}}$

$x = \frac{1}{2} a_0 t_0^2$.

and, $\theta = \omega_0 t_0$

↓

where, a_0, ω_0 will be readings of IMU and t_0 can be found using the millis() function.

NOW, as we are using sabertooth only.

So, for implementation of this model we need to find these 6 functions through testing. V_i denotes ~~voltage to b~~ additional voltage to be supplied to motor i.

V_i (at angular velocity ω) for $i = 1,2,3$

and, V_i (at $v_x = v$) for $i = 1,2,3$.

Rest everything i've tried to explain in the code itself.

For the graph part I stated in the end of the code , I know ML and am thinking of using linear regression to find relation between the variables. Why am I assuming that their relationship will be linear? See below.

Date __/__/__
Page ______
Puffin

→ Consider a circuit, Voltage applied = IR + Back EMF

$$V = IR + E \quad -(1)$$

Now, Back emf (E) $\propto$ ω.

and, Current through the coil (I) $\propto$ Torque applied (T)

Now, T applied will be constant. Putting all this in equation (1),

~~V =~~

$$V = \underbrace{k_1 T \cdot R}_{\text{constant}} + k_2 \omega$$

Thus, linear relation.