ROBOCON-2019

RULE BOOK - click on this link to find the Rule Book

For the abstract of the ROBOCON 2019 please visit this link

MR2

SENSORS USED

1. Potentiometer (discarded)
2. Inertial Measurement Unit - MPU6050

INTEGRATING THE SENSORS ACCORDING TO THE MECHANISM

1. Potentiometer
    - We used potentiometer to measure the angles between the joints of the link in the leg . Mapping the value of resistance from the potentionmeter to the range of angles reqd for the link .
    - The main issue with the potentiometer was that it used to get damage soon . We came across both the internal and external damage of the potentiometer . External damage was caused because of the force from the coupler (this point will be more fabricated) . Internal damage occured because of some electronics issue .
    - The reason of discarding Potentiometer was the hysterisis losses generated in rotations of Potentiometer .
2. MPU6050 (IMU sensor)
    - Datasheet
    - The accelerometer in the MPU was used . The raw values generated were filtered using the arduino library available on GITHUB link
    - The filtered values were then used to calculate the same angles which potentiometer used to calculate . We didn't mapped the values as the calculated angles were in the domain of the angles generated from the raw values .

Calculating the angles for a particular point

- With the help of inverse Kinematics , we calculated the angles for a Particular point . This is somewhat similar to DH parameters but not the same .
- Considering the design of the bot , the calclations were done accordingly .
- The C code for calculating the angles is given below . The function reuqires X,Y the coordinates of the end point of the leg . The function returns the angles required for each link . `calc_angles(X,Y) { for (x>0) {r1 = sqrt(X * X + Y * Y); phi1 = acos(((a4 * a4) - (a2 * a2) - (r1 * r1)) / (-2.0 * a2 * r1)); phi2 = atan(Y / X); T[0][leg] = phi2 - phi1; phi3 = acos(((r1 * r1) - (a2 * a2) - (a4 * a4)) / (-2.0 * a2 * a4)); T[1][leg] = pi - phi3; T[0][leg] = T[0][leg] * 180 / pi; T[1][leg] = T[1][leg] * 180 / pi; } for (x<0) {X = abs(X); r1 = sqrt(X * X + Y * Y); phi1 = acos(((a4 * a4) - (a2 * a2) - (r1 * r1)) / (-2.0 * a2 * r1)); phi2 = atan(-Y / X); phi2 = pi + phi2; T[0][leg] = phi2 - phi1; phi3 = acos(((r1 * r1) - (a2 * a2) - (a4 * a4)) / (-2.0 * a2 * a4)); T[1][leg] = pi - phi3; T[0][leg] = T[0][leg] * 180 / pi; T[1][leg] = T[1][leg] * 180 / pi; } }`

MOVEMENT OF ACTUATORS

- To understand the movement of actuators , you shoud be able to understand the design of the leg thoroughly . As you can observe , if the actuator extracts it leads to change in the angle between the link . Indeed using this logic to the target angle for achieving the required coordinates .
- For the expansion and contraction of the actuators we use angles generated by the MPU6050 as the feedback sensor and use the angles which are calculated by the formulas .
- So now we have the instance of the angle by MPU and we do know the target values of the angles . Comparing those two angles , we command the actuators to either extract or contract .