**Project Report**

**Project Topic: Dynamics and Path Planning of an Underwater Bot**

**The Bot:**

As seen in the picture, the bot has one central motor which gives it a thrust in the vertical direction, 3 motors which provide thrusts along the 3 axes. Translation occurs in the plane of the bot because of the thrust provided by the motors and the direction of rotation of the omni wheels while rotation occurs due to the motion of the flippers which are connected to the shaft connecting the motor to the omni wheels.

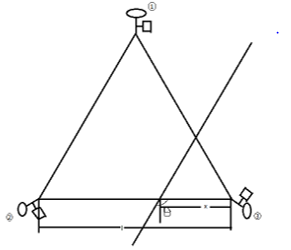


**Tasks completed:**

1. Came up with an equation for the forces on each flipper of the bot to bring about rotation about any arbitrary axis. I started off with considering specific cases and then mathematically induced the general case.

**General case:**Consider a triangle of length l. The axis is inclined at an arbitrary angle to the base and a distance of x from one of the vertices​.

|  |  |  |
| --- | --- | --- |
| **1(+)​** | **2(+)​** | **3(-)​** |
| Sin(30+*θ)\*(l-sin(30+θ)/x\*sin(90-θ))\*F*​  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*​  *(sin(90-θ)+ sin(30+θ)\* (l-sin(30+θ)/x\*sin(90-θ))*​  ​ | (l-x)\*sin(90-*θ)\*F*​  *\_\_\_\_\_\_\_\_\_\_\_*​  *(sin(90-θ)+ sin(30+θ)\* (l-sin(30+θ)/x\*sin(90-θ))*​  ​ | x\*sin(90-*θ)\*F*​  *\_\_\_\_\_\_\_\_\_\_\_*​  *(sin(90-θ)+ sin(30+θ)\* (l-sin(30+θ)/x\*sin(90-θ))*​  ​ |



In order to validate these equations I constructed a 3D model of the bot( a very basic one) on Simscape Multibody where you can model 3D components give them mechanical constraints with respect to each other and simulated motions (both translation and rotation) from the force ratios given by my equation. I have attached the files containing the simulations of the bot model in the **Simulations** folder.

1. Secondly I looked into **trajectory optimization** where I considered a cubic and quintic profile and found the quintic profile to be the most efficient. I have attached the code for both these cases in the ‘**Traj Planning**’ folder.
2. Thirdly I considered the problem of **path planning**. Initially I wrote a code for a Probabilistic Road Map. Turns out that a PRM is computationally expensive, therefore I considered going with an RRT. The approach involved while constructing the RRT involved Sampling ‘n’ random points at a maximum distance of ‘d’ from the point/node under consideration(n is given by maxsamples and d is given by maxdist both of which can be fixed by the user in the beginning). Each of these ‘n’ points are connected to the point under consideration. A heuristic is the calculated for each of the ‘n’ points- that is, distance from the point under consideration to the sampled point added to the distance between the sampled point to the goal and the sample with the least heuristic is picked as the best sample. The path between the point under consideration and the best sample is marked in red. For the next iteration, the ‘best sample’ point is taken to be the point under consideration. If the goal is reached before the maximum number of iterations has been reached, the path planning process is stopped there and then. I wrote the RRT for a 2D case and then modified it for 3D. I have also drawn a small triangular shape to indicate the ‘bot’ which is shown going to all these sample points. I tried performing obstacle avoidance too, but given the time crunch could not perfect it. I have attached the code I have used nevertheless. Obstacle avoidance has been performed using the fact that the distance between the center of the obstacle and the start/end node should exceed the hypotenuse of the obstacle radius and half the distance between the start and end node. Code for this has been attached in the **Path Plan folder**.