**Delivery bot**

Objective:

To deliver an object from one position to another in the same room using TX1 without colliding into other objects.

Hardware:

Motors, motor drivers, Camera, SONAR, motor encoders, Arduino, Jetson TX1, tires, wires,etc.

Abstract and algorithm:

For the coordinates of the location we wish to deliver the object we have 3 options:

1. We can give tx1 the exact coordinates and by using an motor encoders readings along with the sense and avoid algorithm we can get the work done
2. If we wish to move the object in the same room to a specific person we can train a DIGITS model for specific facial recognition then upload the trained model to detectnet on tx1 to use it on detectnet-camera to find the coordinates of the location and further more do the same as above.
3. If we wish to move the object to another object in the same room we can train a DIGITS model for the object or use the pre trained models from imagenet then upload the trained model to jetson tx1 to use it on detectnet-camera command to find the coordinates of the location and further more do the same as above.

Train a DIGITS model with a few source images, or by predesigning a gazebo model of the destination object. Uploading this model on tx1 and using it as the source for comparison, we can get the coordinates of the destination using detectnet-camera for real time object detection.

Next we need to reach the coordinates without crashing into obstacles.

We’ll have to include the ROS library, geometry\_msgs/Twist.h and std\_msgs/Int16 first to create a topic for sense and avoid to initialize the starting position and the ending position, to control the geometrical movement of the bot and to subscribe to various other topics being the motor encoders data respectively.

Furthermore, we initialize the safety values for maximum safe distance from an object, the linear speed, the angular speed, etc.

Next we initialize the velocity variable to publish the velocity value to the motor drivers while initializing the motor encoders and SONAR variables to subscriber to the same via the Arduino nodes.

We also initialize enumeration for the states of the bot, being “FORWARD”, “REVERSE” and “TURN” and initialize the variable value for the same as FORWARD first.

Now while the Data from the motor encoders in not the same as that of the coordinate values that came from “detect-net” command.

{

If (state is FORWARD & data from SONAR < minimum object distance initialized) {

Publish ROS\_INFO("REVERSE")

state changes to REVERSE.

velocity linear in x coordinate becomes –LINEAR SPEED initialized previously.

velocity angular in z coordinate becomes 0.

publish(above velocity changes) to the motor drivers

}

else if (state is FORWARD) {

velocity linear in x coordinate becomes LINEAR SPEED initialized previously.

velocity angular in z coordinate becomes 0.

publish(above velocity changes) to the motor drivers

}

else if(state is REVERSE & data from sonar > minimum object distance initialized ) {

Publish ROS\_INFO("TURN").

state changes to TURN.

velocity linear in x coordinate becomes 0.

velocity angular in z coordinate becomes ANGULAR\_SPEED initialized previously.

publish(above velocity changes) to the motor drivers

}

else if (state is TURN & left\_count > turn duration limit initialized earlier) {

state changes to FORWARD

Publish ROS\_INFO("FORWARD")

velocity linear in x coordinate beco­­­­mes LINEAR SPEED initialized previously.

velocity angular in z coordinate becomes 0.

publish(above velocity changes) to the motor drivers

}

}

Uses:

As a stockroom bot in industries, domestic help, as a waiter, etc.