Implementation of a Robot Behaviour Learning Simulator

Kushagra Singh BISEN

Ecole des Mines de Saint Etienne

kushagrasingh.bisen@etu.emse.fr

July 29, 2021

Work Updates

- Log files are working and I can produce the log files.
- Method for Dynamic Obstacle Avoidance is done.
- Shift from Turtlebot2 to Turtlebot3 is done.

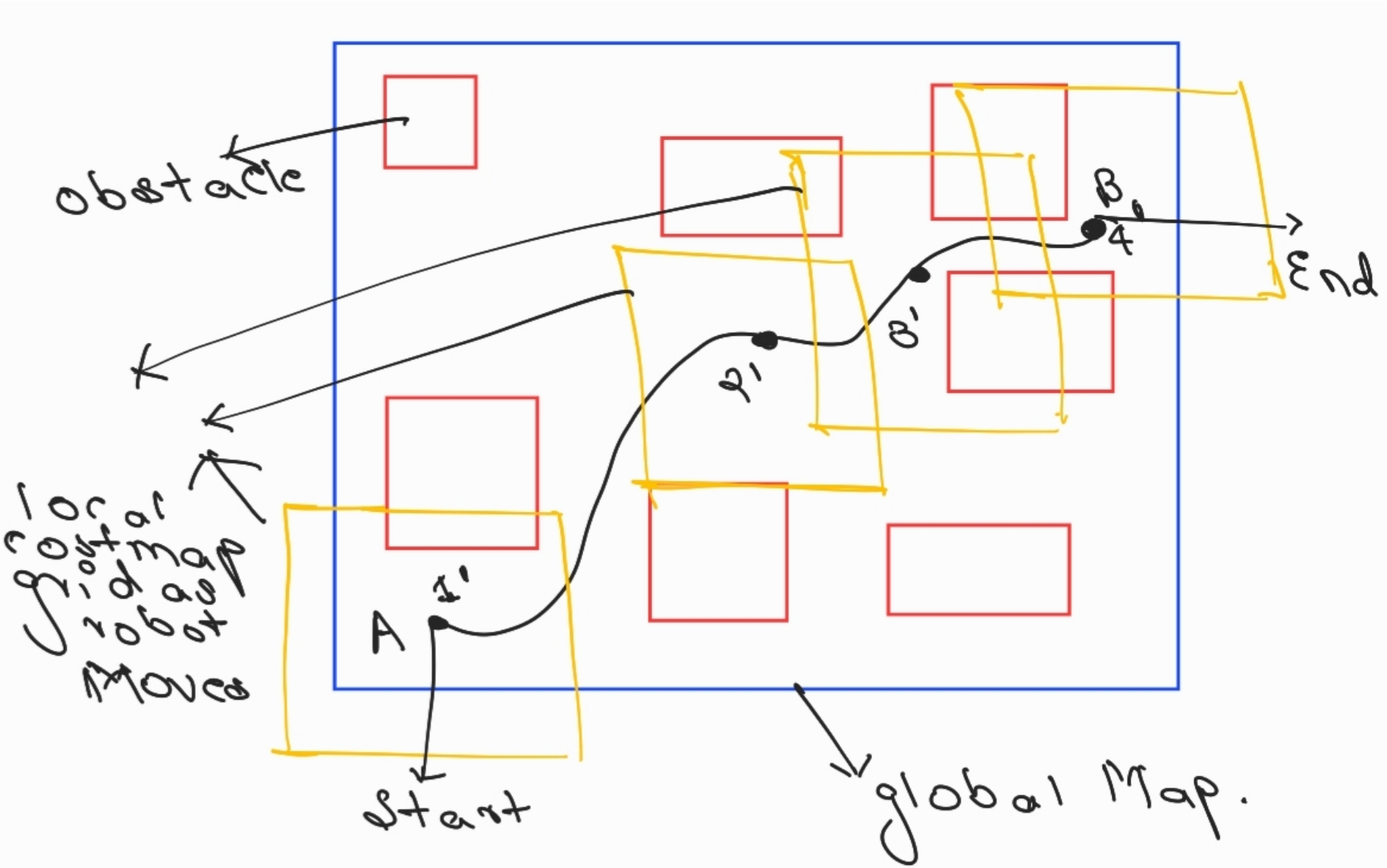
In the subsequent sections, I will talk about these different sections

On the Log File

Regarding the log file, we are using the odometry topic to produce the log for detailing the 2D position of the robot. This particular part has been executed.
Now, when we talk about the robot's neighbouring space, then some problems arise which we are trying to solve from last 3 meetings.

- We wish to know the neighbouring grid space and the obstacle of the robot as it is moving.
- I tried with calculating the path and then realized that solution to the problem is not 'scalable' at all.
- The particular solution is not accurate either, and will provide a bad result in our experiment.

Continued — On the Log File



Continued — On the Log File

Regarding the simulation with log file, I know that,

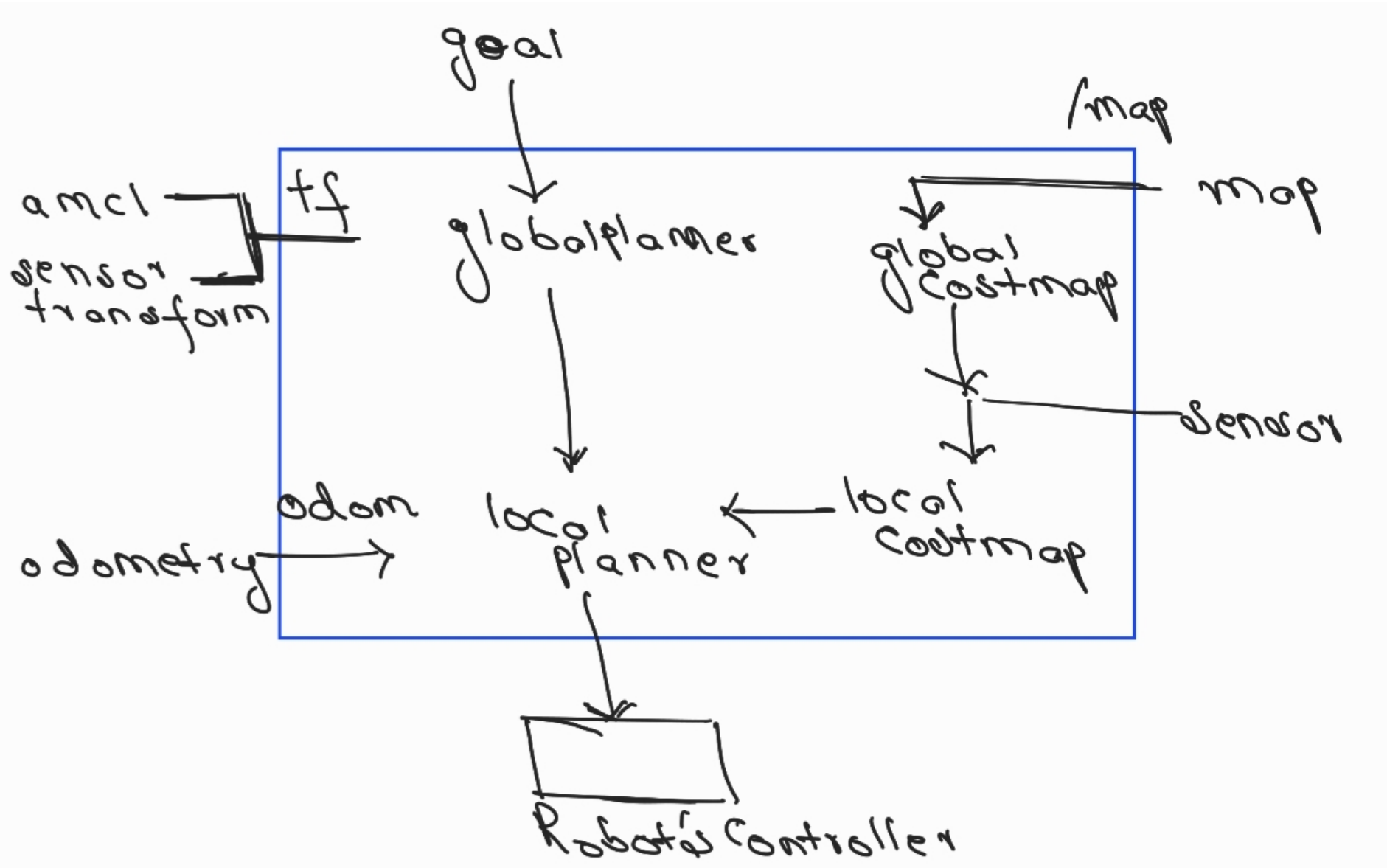
- We wish to predict the future trajectory of the robot.
- The previous simulator (with Java i.e before my internship) could not consider more than 20 obstacles.
- The reinforcement learning decision tree based model for predicting the trajectory is done.
- I come to generate the data for the machine learning model to train on, on various different environments.

Continued — On the Log File

Now, the idea is to generate the data for the machine learning model 'from the environment'. But there are some problems with it,

- The environment can not generate any data, the map can be used for the data.
- The global costmap is made by using the 360 degree LIDAR sensor.
- The robot has some inertia and some ability to 'cut the corners' when it is going from A to B.
- The method I was making (3 by 3) grid is not accurate as the robot can not follow the path totally.
- The old method is not very accurate, and to 'predict' the trajectory and as a point of error is not acceptable. We can not use that method.

On the Robot Simulation



On the Robot Simulation

So, the current method can choose the grid size, and the data will be accurate to the movement of the robot and it doesn't exactly take the data from the robot or the robot doesn't publish the data but the ROS architecure uses the sensor on the robot because there is no other way to do the same.

Dynamic Obstacle Avoidance

I have included the method for avoiding dynamic obstacle or the other robot, but since we want to predict the trajectory using a pre-existing approach is not good (I think) so I have no implemented. (But I can show you a simple demonstration)

Shift from Turtlebot2 to Turtlebot3

I did this because we have Turtlebot3 in our lab and they should run on the same software otherwise there will be a driver/controller issue. Also, for the obstacle avoidance, a 360 degree LIDAR sensor is available in the Turtlebot3.

Thank you for your time.