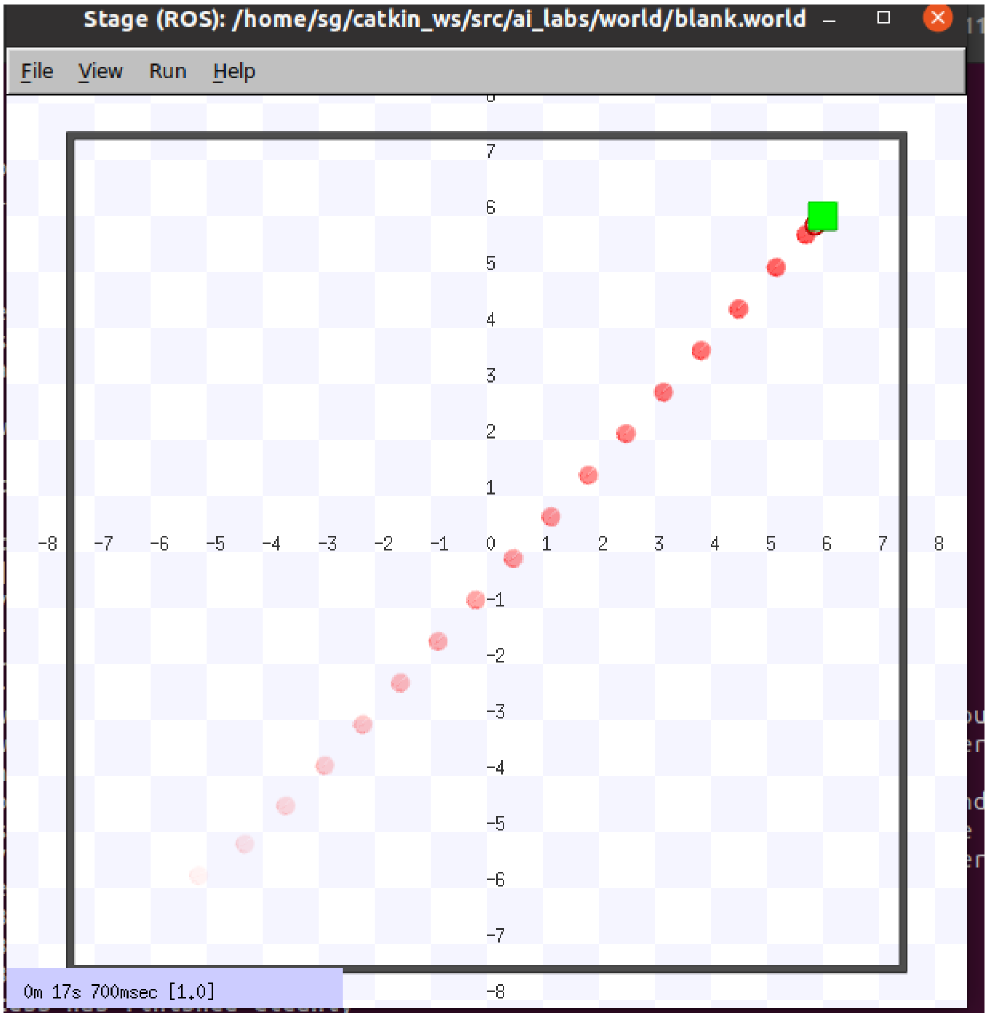
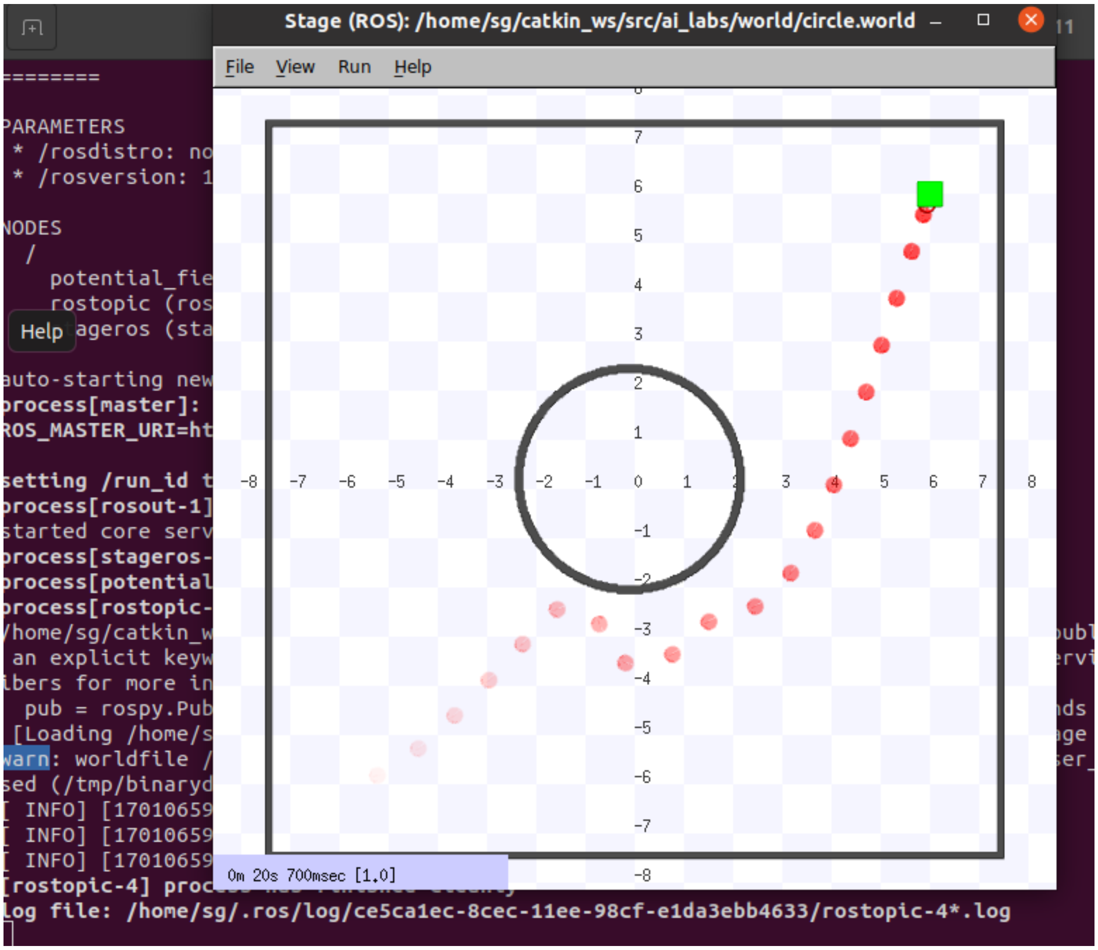
Lab 4 – Navigating with Potential Fields

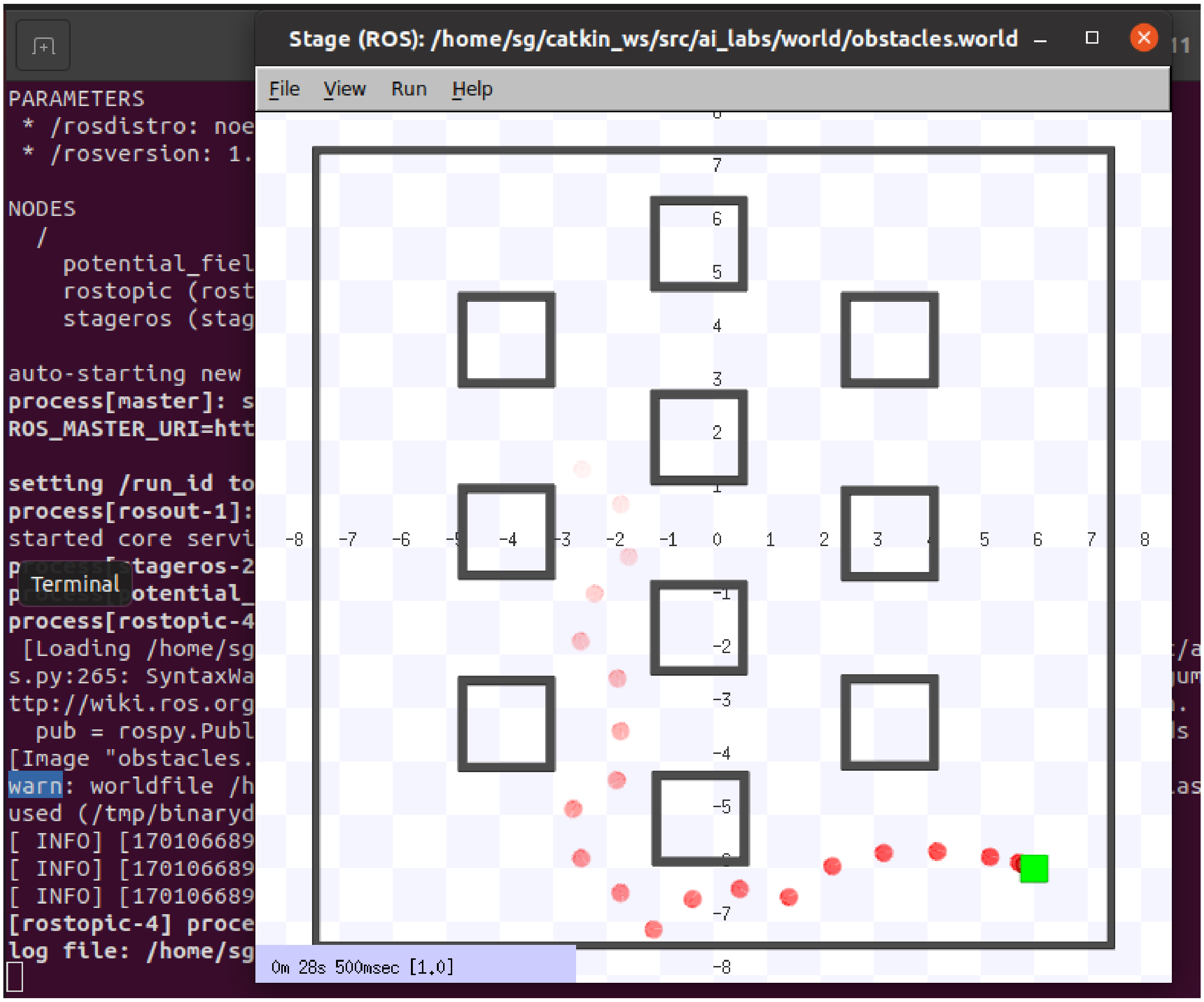
PartA:



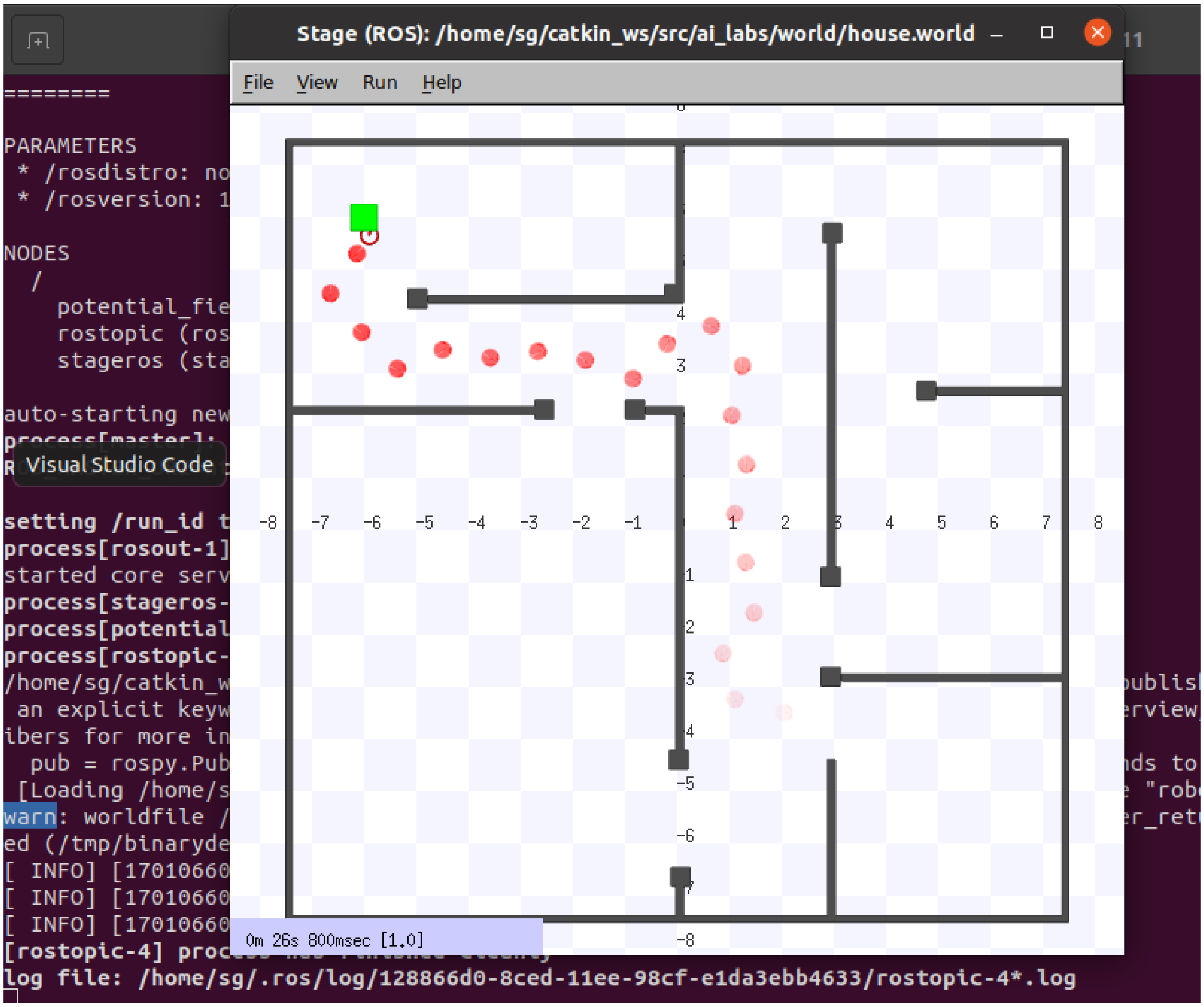
PART B:


Stage (ROS): /home/sg/catkin_ws/src/ai_labs/world/circle.world
File View Run Help
0m 20s 700msec [1.0]
log file: /home/sg/.ros/log/ce5ca1ec-8cec-11ee-98cf-e1da3ebb4633/rostopic-4*.log

Part C:


Stage (ROS): /home/sg/catkin_ws/src/ai_labs/world/obstacles.world
File View Run Help
PARAMETERS
* /rosdistro: noe
* /rosversion: 1.
NODES
/
potential_fiel
rostopic (rost
stageros (stag
auto-starting new
process[master]: s
ROS_MASTER_URI=htt
setting /run_id to
process[rosout-1]:
started core servi
process[stageros-2
process[potential_
process[rostopic-4
[Loading /home/sg
s.py:265: SyntaxWa
ttp://wiki.ros.org
pub = rospy.Publ
[Image "obstacles.
warn: worldfile /h
used (/tmp/binaryd
[INFO] [170106689
[INFO] [170106689
[INFO] [170106689
[rostopic-4] proce
log file: /home/sg
Terminal
0m 28s 500msec [1.0]

Part D:



Part E: Discussion:

1. Describe the process you used to determine what values to set the parameters in the code to. Do you think there is a way to do this automatically? Why or why not?

Ans: Process of determining Parameter Values:

1. **Initial Theoretical Estimates**: I start with basic theoretical values based on the robot's specifications (given default values) and the expected operating environment.
2. **Empirical Testing and Observation**: I Implemented the algorithm with these initial values and observe the robot's behavior in a controlled environment. I noted down how it responds to obstacles and its ability to reach the goal.
3. **Iterative Adjustments**: Based on observations, I iteratively adjust the parameters. Increase the strength of the repulsive force if the robot gets too close to obstacles or increase the attractive force if it's not effectively moving towards the goal.

4. **Balancing Forces**: I make ensure a balance between attractive and repulsive forces. The robot should neither be overly aggressive in moving towards the goal nor too timid in avoiding obstacles.
5. **Environment Variability**: Test in different environments A, B and C world to understand how changes in obstacle density, types, and goal distances affect performance. I need to adjust how I calculate goal force, obstacle force.

Maybe there's potential for automating the process with advanced machine learning techniques, it remains a non-trivial task due to the variability and complexity involved in robot-environment interactions. Manual tuning, informed by both theory and empirical testing, often provides a more straightforward and immediate solution, especially for specific or controlled environments.

2. Describe and explain (give reasons for) the behavior of the robot when it could sense one or more obstacles and a goal was present.

Ans: With both obstacles and a goal present, the robot's behavior is influenced by both attractive and repulsive forces:

- **Towards the Goal**: The attractive force directs the robot towards the goal.
- **Avoiding Obstacles**: When an obstacle is detected within the threshold distance, the repulsive force acts in the opposite direction to the obstacle, influencing the robot to steer away from it.
- **Balancing Forces**: The robot continuously balances these forces, often resulting in a path that points towards the goal while avoiding obstacles.

3. Describe how the different repulsive force magnitude profile (linear) that you implemented changed the behavior of the robot, and why. If it doesn't change anything, explain why this is.

Ans: Using a linear repulsive force profile impacts the robot's behavior in the following ways:

- **Gradual Response to Obstacles**: As opposed to a sudden change in force, the linear profile leads to a more gradual increase in repulsive force as the robot approaches an obstacle. This can result in smoother avoidance maneuvers.
- **Distance Proportional Response**: The repulsion strength being proportional to the distance from the obstacle allows for more nuanced obstacle avoidance, particularly in environments with varying obstacle densities.
- **Behavioral Change**: If the behavior didn't change significantly, it might be due to the robot's operating environment not testing the limits of the linear profile, or the chosen parameters not being distinct enough from the constant profile scenario.

4. List some advantages of the potential fields approach as demonstrated in this lab.

Ans: Some advantages of the potential fields approach are:

- **Simplicity**: The potential fields method is conceptually straightforward and easy to implement.
- **Real-time Operation**: It is suitable for real-time navigation in dynamic environments.
- **Local Minima Avoidance**: Well-tuned parameters can help the robot avoid getting stuck in local minima, a common issue with potential fields.

5. List some disadvantages of the potential fields approach as demonstrated in this lab

Ans: Some of disadvantages of the potential fields approach are:

- **Local Minima**: The robot can get stuck in local minima where repulsive forces from obstacles and the attractive force to the goal balance each other.
- **Oscillations**: In certain scenarios, especially in narrow passages or around sharp obstacles, the robot may oscillate.
- **Lack of Path Optimization**: The method doesn't inherently optimize the path length or time; it's more about reaching the goal while avoiding obstacles.
- **Parameter Sensitivity**: The effectiveness is highly dependent on the correct tuning of parameters, which can be challenging and environment specific.