

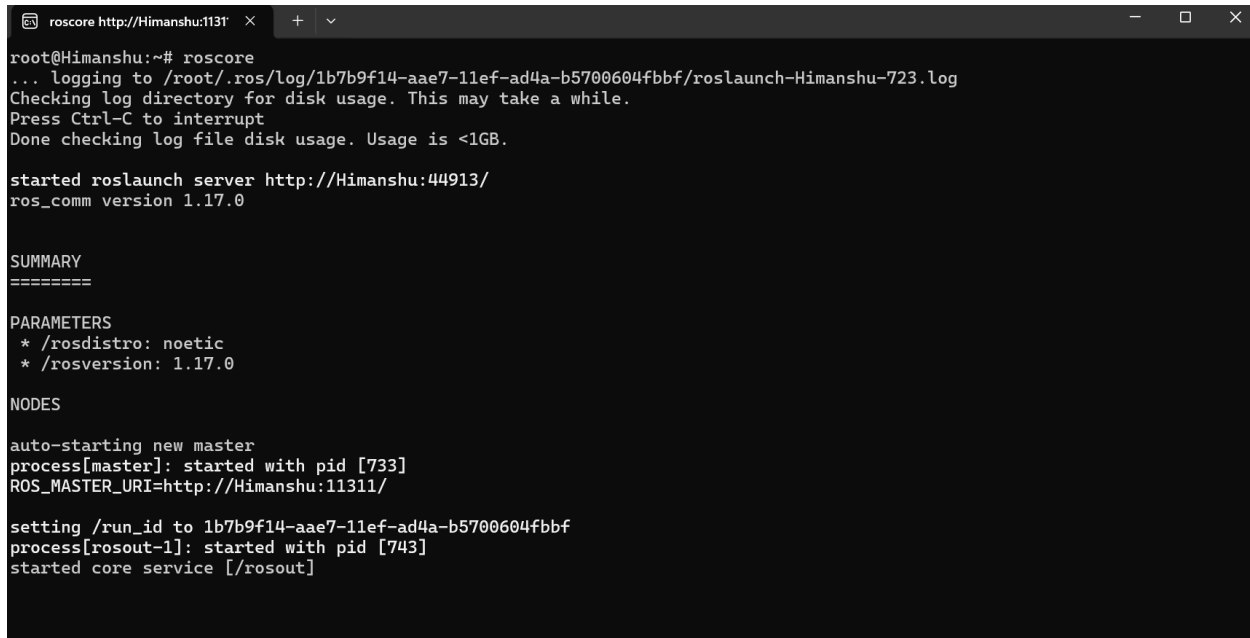
# Design and Development

## 1. Implementation Plan

### 1.1 Launching 3 Turtles (one leader (Turtle 1) and two followers (Turtle 2 and Turtle 3))

- Use a launch file to spawn three turtles in the turtlesim simulator:

**Figure 1.** Launching the Turtle simulator with ROS package

A terminal window titled 'roscore http://Himanshu:1131' showing the output of the 'roscore' command. The output includes logging information, disk usage checks, and the start of a roslaunch server. It lists parameters like '/rostdistro: noetic' and '/rosversion: 1.17.0'. Under the 'NODES' section, it shows the auto-starting of a new master and the starting of 'rosout-1' and 'core service [/rosout]'.

```
root@Himanshu:~# roscore
... logging to /root/.ros/log/1b7b9f14-aae7-11ef-ad4a-b5700604fbbf/roslaunch-Himanshu-723.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://Himanshu:44913/
ros_comm version 1.17.0

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.17.0

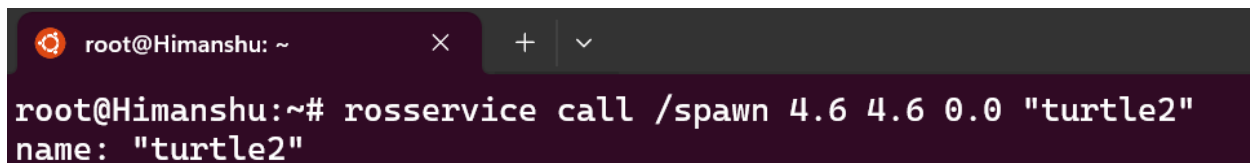
NODES

auto-starting new master
process[master]: started with pid [733]
ROS_MASTER_URI=http://Himanshu:11311/

setting /run_id to 1b7b9f14-aae7-11ef-ad4a-b5700604fbbf
process[rosout-1]: started with pid [743]
started core service [/rosout]
```

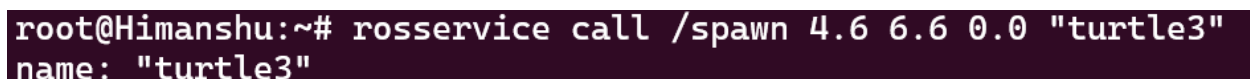
- Leader: Leader at the center (e.g., (/spawn 5.5 5.5 0.0 "turtle1"))
- Followers: FollowerA and FollowerB at random positions.

Follower 1 (**Turtle 2**, e.g., (/spawn 4.6 4.6 0.0 "turtle2"))

A terminal window showing the command 'rosservice call /spawn 4.6 4.6 0.0 "turtle2"' being executed. The output is 'name: "turtle2"'.

```
root@Himanshu:~# rosservice call /spawn 4.6 4.6 0.0 "turtle2"
name: "turtle2"
```

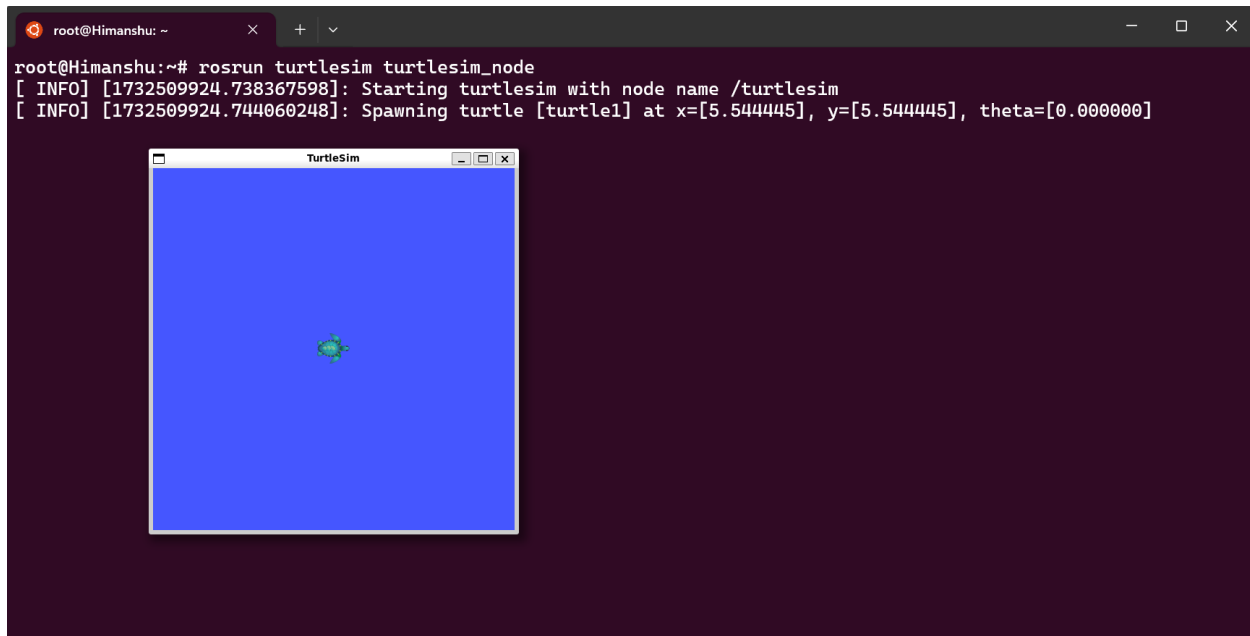
Follower 2 (**Turtle 3**, e.g., (/spawn 4.6 6.6 0.0 "turtle1"))

A terminal window showing the command 'rosservice call /spawn 4.6 6.6 0.0 "turtle3"' being executed. The output is 'name: "turtle3"'.

```
root@Himanshu:~# rosservice call /spawn 4.6 6.6 0.0 "turtle3"
name: "turtle3"
```

- Include commands to set a random background color (Blue color) for the simulator.

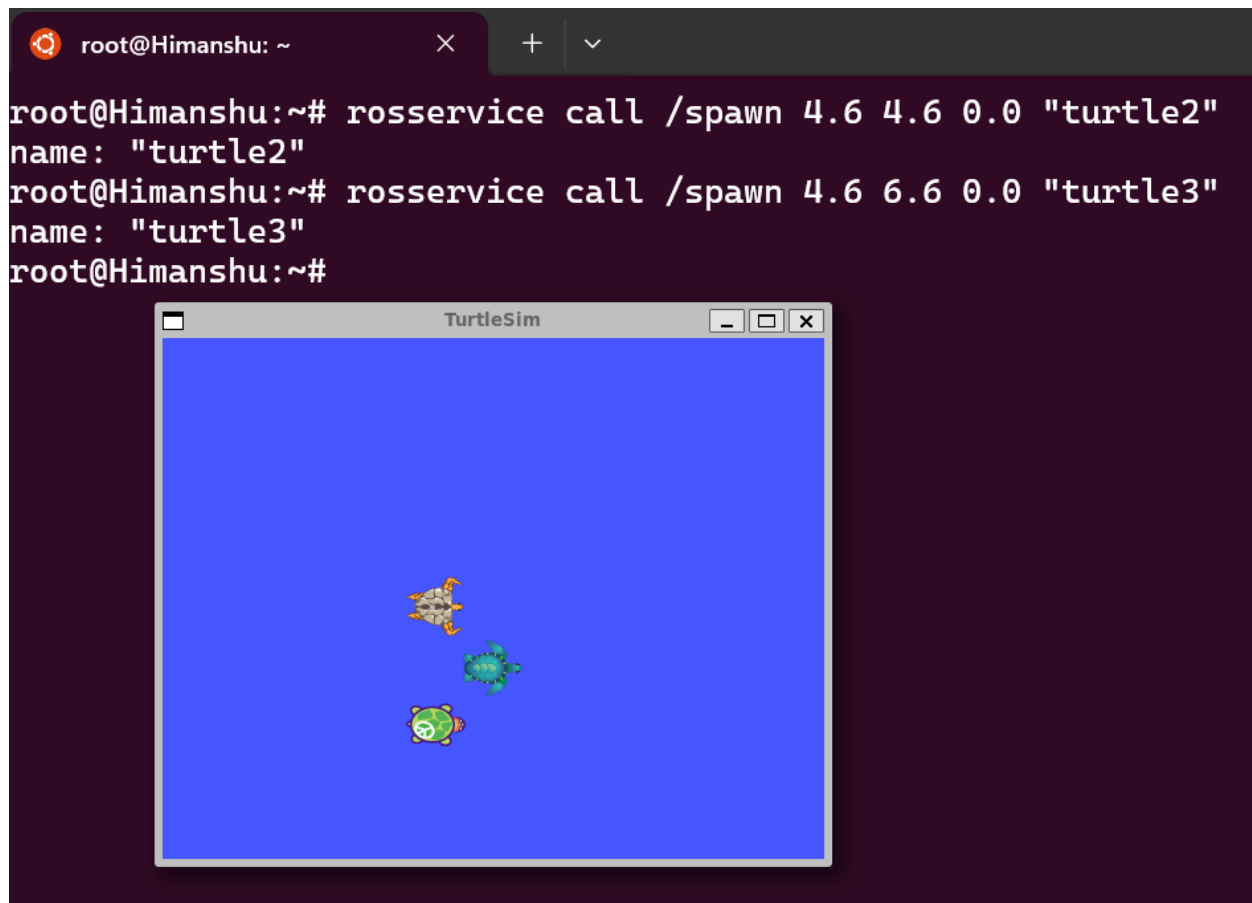
```
root@Himanshu: ~  
root@Himanshu:~# roslaunch turtlesim turtlesim_node  
[ INFO] [1732509520.086140121]: Starting turtlesim with node name /turtlesim  
[ INFO] [1732509520.093969414]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
[ INFO] [1732509535.467086901]: Spawning turtle [turtle2] at x=[4.600000], y=[4.600000], theta=[0.000000]  
[ INFO] [1732509548.130883686]: Spawning turtle [turtle3] at x=[4.600000], y=[6.600000], theta=[0.000000]
```



```
root@Himanshu:~# rosservice call /spawn 4.6 4.6 0.0 "turtle2"  
name: "turtle2"  
root@Him
```

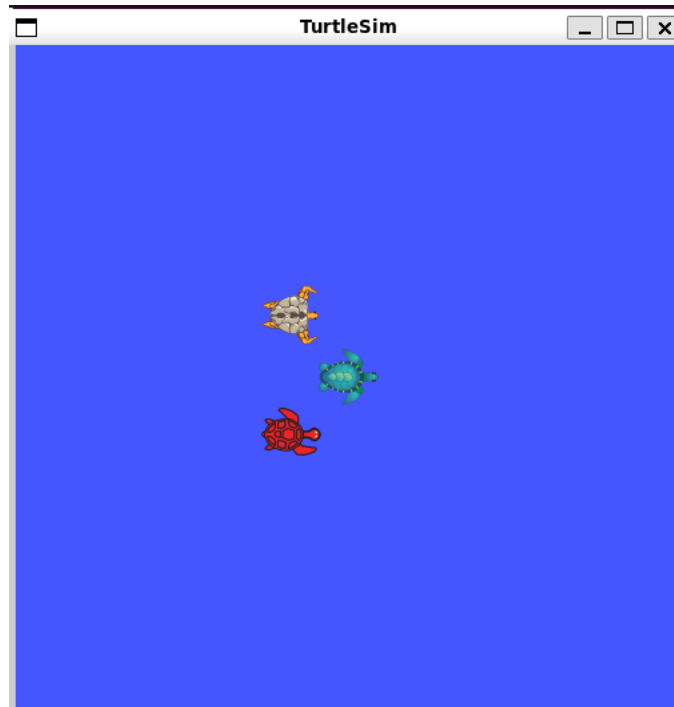


```
root@Himanshu:~# rosservice call /spawn 4.6 4.6 0.0 "turtle2"  
name: "turtle2"  
root@Himanshu:~# rosservice call /spawn 4.6 6.6 0.0 "turtle3"  
name: "turtle3"  
root@Himanshu:~#
```



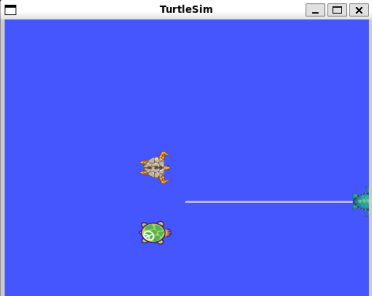
## 1.2 Achieving Formation Control

- Define a **custom message** (YourBcodeLeaderS1Command.msg) to send instructions from the leader to the followers. The message includes:
  - float32 target\_x – Desired x-position.
  - float32 target\_y – Desired y-position.
  - float32 orientation – Desired orientation (Linear).
- Use the **TF2 library** to define reference frames:
  - **Leader Frame**: At the leader's position.
  - **FollowerA Frame**: Offset by (-1, 0) from the leader.
  - **FollowerB Frame**: Offset by (1, 0) from the leader.



```
[ WARN] [1732510481.775387538]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.791638623]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.807364627]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.823209469]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.839164139]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.854835291]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.871590245]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.887209366]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.902792417]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.919482751]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.935066184]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.950611582]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.967248672]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.982837817]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510481.999357316]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.015054865]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.031462515]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.047000922]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.063915991]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.079830467]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.095390166]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.111188619]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.126769401]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.143502701]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.159691853]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.175325107]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.190878596]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
[ WARN] [1732510482.206435553]: Oh no! I hit the wall! (Clamping from [x=11.112889, y=6.488889])
```

```
root@Himanshu: ~  
root@Himanshu:~# rosservice call /spawn 4.6 4.6 0.0 "turtle2"  
name: "turtle2"  
root@Himanshu:~# rosservice call /spawn 4.6 6.6 0.0 "turtle3"  
name: "turtle3"  
root@Himanshu:~# rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 10 '{linear: {x: 1.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}'
```



The TurtleSim window displays a blue environment with three turtles. Turtle 1 (orange) is at the top left, Turtle 2 (green) is at the bottom left, and Turtle 3 (blue) is on the right. White lines indicate their movement paths: Turtle 1 moves right, Turtle 2 moves right, and Turtle 3 moves left.

