

4-1. Turtlesim이 특정 각도로 이동하는 단순 코드 작성 해보기

터미널 1

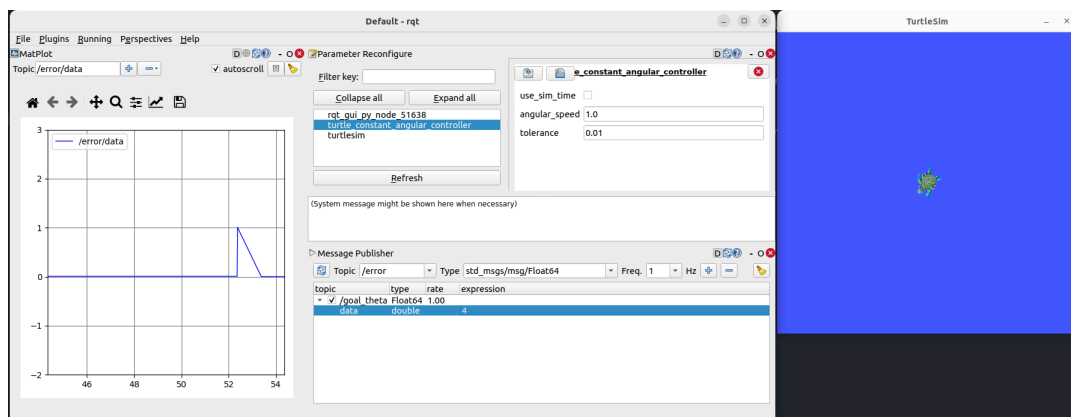
```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg simple_rotate
```

터미널 3

```
rqt -> Plot rqt -> Parameter Reconfigure rqt -> Message Publisher
```



[그림 4-1] 실행 결과 예시

4-2. PID 제어기로 turtlesim 각도 제어하기

터미널 1

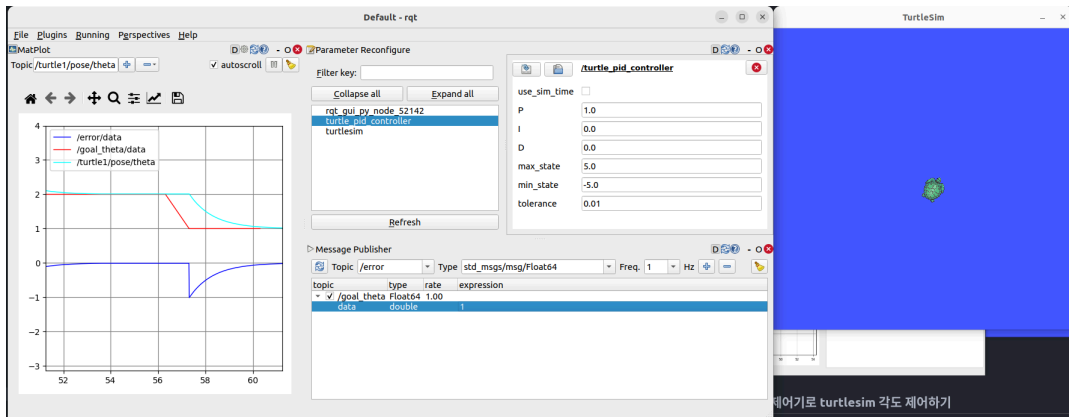
```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg control_rotate
```

터미널 3

```
rqt -> Plot rqt -> Parameter Reconfigure rqt -> Message Publisher
```



[그림 4-2] 실행 결과 예시

4-3. Turtlesim을 활용해 두 개의 PID 제어기로 로봇 위치 제어 연습하기

터미널 1

```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg pose_dual_controller
```

터미널 3

```
ros2 topic pub --once /goal_pose turtlesim/msg/Pose "{x: 9.,y: 7.,theta: 0.}"
```

4-4. PID로 구현된 로봇 구동 제어기의 상태를 멋지게 모니터링해보자

터미널 1

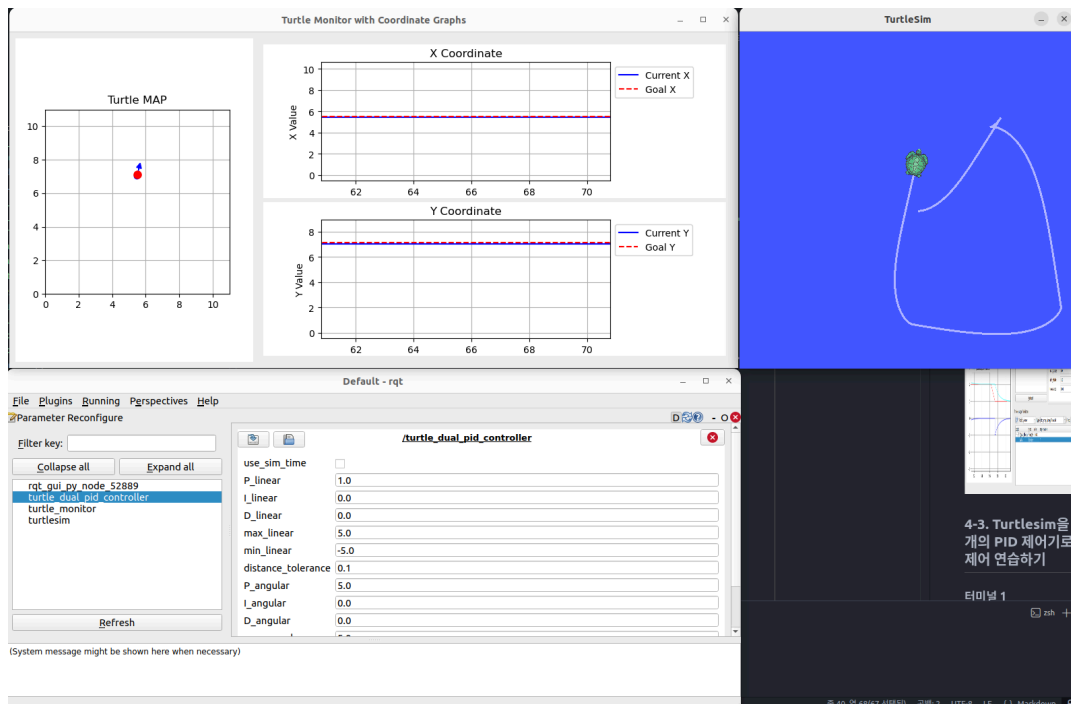
```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg pose_dual_controller
```

터미널 3

```
ros2 run controller_tutorials_pkg qmonitor_for_pose_dual_controller
```



[그림 4-4] 실행 결과 예시

4-5. 로봇 주행 제어기를 state로 나눠서 구현하기

터미널 1

```
ros2 run turtlesim turtlesim_node
```

터미널 2

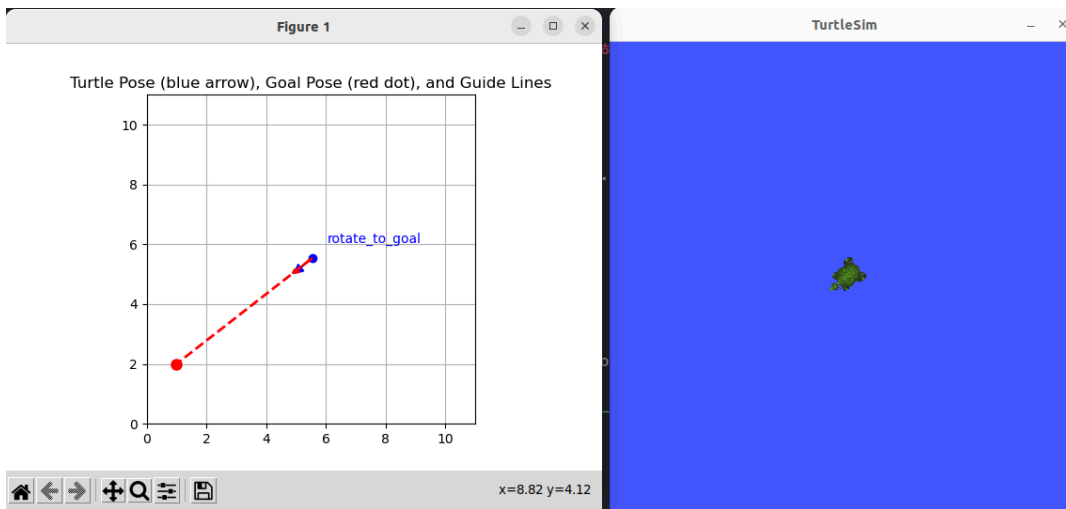
```
ros2 run controller_tutorials_pkg move_turtle
```

터미널 3

```
ros2 run controller_tutorials_pkg monitor_for_move_turtle
```

터미널 4

```
ros2 topic pub --once /goal_pose turtlesim/msg/Pose "{x: 1.,y: 2.,theta: 0.}"
```



[그림 4-5] 실행 결과 예시

4-6. 간단한 주행 제어기를 StateMachine으로 구현해 보기

터미널 1

```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg move_turtle_state_machine
```

터미널 3

```
ros2 topic pub --once /goal_pose turtlesim/msg/Pose "{x: 1.,y: 2.,theta: 0.}"
```

터미널 4

```
ros2 topic echo /state
```

4-7. StateMachine으로 구현된 주행 제어기를 QT로 모니터링 해보자

터미널 1

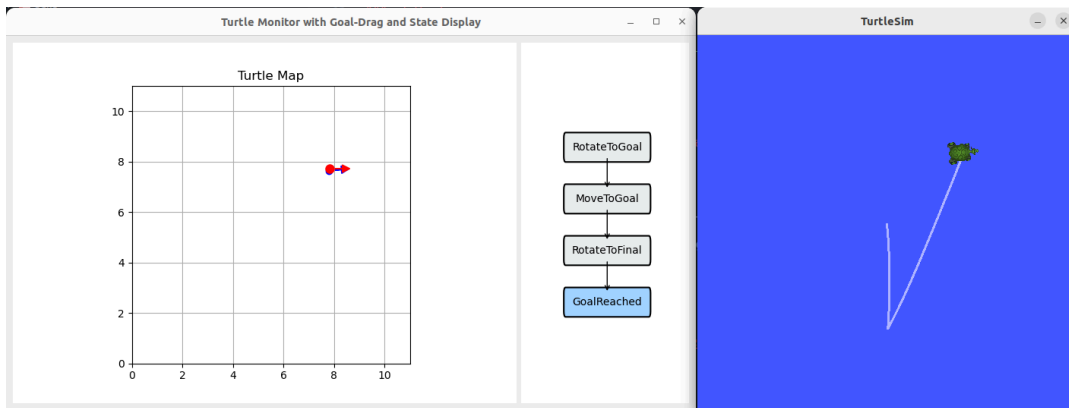
```
ros2 run turtlesim turtlesim_node
```

터미널 2

```
ros2 run controller_tutorials_pkg move_turtle_state_machine
```

터미널 3

ros2 run controller_tutorials_pkg qmonitor_state_machine



[그림 4-7] 실행 결과 예시