1. ROS 토픽 메시지를 이용해 드론의 현재 위치를 받아 목적지 근처에 도착했는지를 확인하는 ROS 응용 패키지를 작성해 보시오.

■ 구현 요구사항

- Publisher (**drone.py**)
  - ✓ node name: drone
  - ✓ 'drone' publisher 노드 초기화 후 초기화가 되었다는 메시지 출력
  - ✓ 1초마다 3차원 좌표값으로 구성된 드론의 현재 위치를 담은 "/drone_position" 토픽 메시지를 발행
  - ✓ 드론의 현재 위치 x, y, z 값은 random.uniform() 함수를 이용하여 40에서 60사이 임의의 값으로 생성
  - ✓ 터미널 화면에도 현재 위치 좌표값 출력
- Subscriber (**monitor.py**)
  - ✓ node name: monitor
  - ✓ 'monitor' subscriber 노드 초기화 후 초기화가 되었다는 메시지 출력
  - ✓ "/dron_position" 토픽 메시지를 수신한 후 타겟 위치와의 거리를 계산하여 threshold 값 이내면 도착했다는 메시지를 출력하고, 그렇지 않으면 현재 드론의 위치값만 출력
  - ✓ 타겟 위치 좌표는 (Target_x, Target_y, Target_z) = (50.0, 50.0, 50.0)으로 설정
  - ✓ 거리 계산: $distance = \sqrt{(Target_x - x)^2 + (Target_y - y)^2 + (Target_z - z)^2}$
  - ✓ 목적지 인근에 도착했는지 여부를 판단하는 임계값 (Threshold) 는 8.0으로 설정
- 토픽 메시지 (**/dron_position**)
  - ✓ 3차원 좌표 공간에서 좌표값을 표현해야 하므로 geometry_msgs/Point 메시지 타입 사용
- rosbag 파일 (**hw2_1.bag**)
  - ✓ drone.py, monitor.py를 실행한 후 "/drone_position" 토픽 데이터를 hw2_1.bag 파일에 기록
  - ✓ 명령어: rosbag record -O hw2_1.bag /drone_position
  - ✓ 기록한 데이터에는 최소 10줄 이상의 내용이 포함되어야 함.

■ Hint

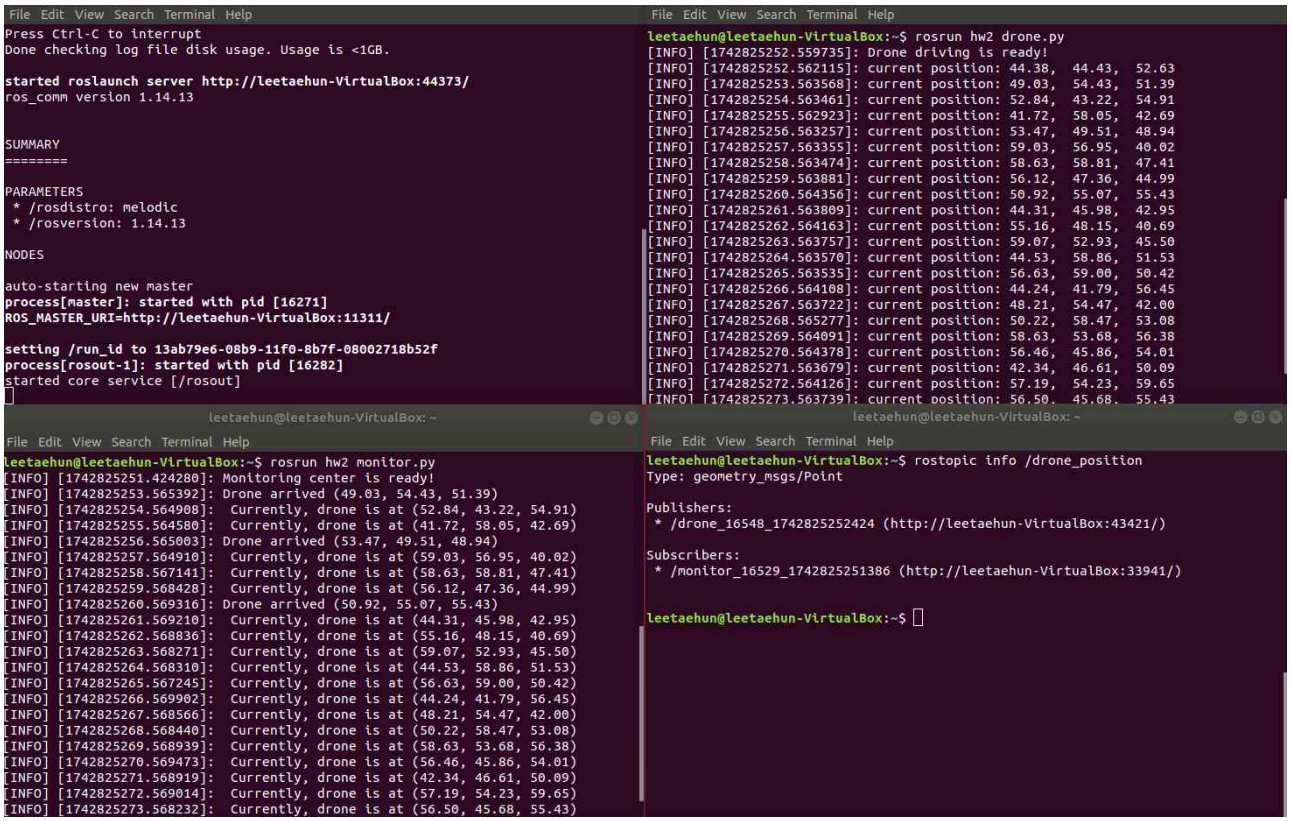
- 아래 주어진 pseudo 코드와 4주차 실습 노트의 pub.py, sub.py 구현 내용 참고
- drone.py

```
#!/usr/bin/env python2
import rospy
import random
from geometry_msgs.msg import Point

def report_position():
    # 'drone' publisher 노드 초기화
    #  - Point 타입 메시지를 담은 /drone_position 토픽을 1초마다 발행하도록 설정
    ...
    # 발신 노드가 초기화되었다는 메시지 출력
    ...
    while not rospy.is_shutdown():
       # random.uniform() 함수를 이용해 임의의 값을 가진 x, y, z 좌표 생성
       #  - 예: pos = random.uniform(40, 60)
         ...
       # 현재 드론 좌표값 출력
       rospy.loginfo(.....)
       # 토픽 메시지 발행

if __name__ == '__main__':
    try:
        report_position()
    except rospy.ROSInterruptException:
        pass
```

- monitor.py

```
#!/usr/bin/env python2
import rospy
import math
from geometry_msgs.msg import Point

# 타겟 위치와 위치 임계치 변수 선언
    ...
    ...

def callback(data):
    # 토픽 메시지 내용인 data를 참조하여 목적지와 현재 드론의 위치 사이 거리 계산
    # 거리가 임계치 이하면 drone이 근처에 도착했다는 메시지 출력,
    #  그렇지 않으면, 드론의 좌표값만 출력
    ...
    ...

def monitor():
    # 'monitor' subscriber 노드 초기화
    #  - /drone_position 토픽 수신 후 callback 호출하도록 설정
    ...
    # 수신 노드가 초기화되었다는 메시지 출력
    ...
    rospy.spin()

if __name__ == '__main__':
    try:
        monitor()
    except rospy.ROSInterruptException:
        pass
```

■ 예시 실행 출력 결과


File Edit View Search Terminal Help
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://leetaehun-VirtualBox:44373/
ros_comm version 1.14.13
SUMMARY
PARAMETERS
* /rosdistro: melodic
* /rosversion: 1.14.13
NODES
auto-starting new master
process[master]: started with pid [16271]
ROS_MASTER_URI=http://leetaehun-VirtualBox:11311/
setting /run_id to 13ab79e6-08b9-11f0-8b7f-08002718b52f
process[rosout-1]: started with pid [16282]
started core service [/rosout]
leetaehun@leetaehun-VirtualBox: ~
File Edit View Search Terminal Help
leetaehun@leetaehun-VirtualBox:~$ rosrun hw2 monitor.py
[INFO] [1742825251.424280]: Monitoring center is ready!
[INFO] [1742825253.565392]: Drone arrived (49.03, 54.43, 51.39)
[INFO] [1742825254.564908]: Currently, drone is at (52.84, 43.22, 54.91)
[INFO] [1742825255.564580]: Currently, drone is at (41.72, 58.05, 42.69)
[INFO] [1742825256.565003]: Drone arrived (53.47, 49.51, 48.94)
[INFO] [1742825257.564910]: Currently, drone is at (59.03, 56.95, 40.02)
[INFO] [1742825258.567141]: Currently, drone is at (58.63, 58.81, 47.41)
[INFO] [1742825259.568428]: Currently, drone is at (56.12, 47.36, 44.99)
[INFO] [1742825260.569316]: Drone arrived (50.92, 55.07, 55.43)
[INFO] [1742825261.569210]: Currently, drone is at (44.31, 45.98, 42.95)
[INFO] [1742825262.568836]: Currently, drone is at (55.16, 48.15, 40.69)
[INFO] [1742825263.568271]: Currently, drone is at (59.07, 52.93, 45.50)
[INFO] [1742825264.568310]: Currently, drone is at (44.53, 58.86, 51.53)
[INFO] [1742825265.567245]: Currently, drone is at (56.63, 59.00, 50.42)
[INFO] [1742825266.569902]: Currently, drone is at (44.24, 41.79, 56.45)
[INFO] [1742825267.568566]: Currently, drone is at (48.21, 54.47, 42.00)
[INFO] [1742825268.568440]: Currently, drone is at (50.22, 58.47, 53.08)
[INFO] [1742825269.568939]: Currently, drone is at (58.63, 53.68, 56.38)
[INFO] [1742825270.569473]: Currently, drone is at (56.46, 45.86, 54.01)
[INFO] [1742825271.568919]: Currently, drone is at (42.34, 46.61, 50.09)
[INFO] [1742825272.569014]: Currently, drone is at (57.19, 54.23, 59.65)
[INFO] [1742825273.568232]: Currently, drone is at (56.50, 45.68, 55.43)
File Edit View Search Terminal Help
leetaehun@leetaehun-VirtualBox:~$ rosrun hw2 drone.py
[INFO] [1742825252.559735]: Drone driving is ready!
[INFO] [1742825252.562115]: current position: 44.38, 44.43, 52.63
[INFO] [1742825253.563568]: current position: 49.03, 54.43, 51.39
[INFO] [1742825254.563461]: current position: 52.84, 43.22, 54.91
[INFO] [1742825255.562923]: current position: 41.72, 58.05, 42.69
[INFO] [1742825256.563257]: current position: 53.47, 49.51, 48.94
[INFO] [1742825257.563355]: current position: 59.03, 56.95, 40.02
[INFO] [1742825258.563474]: current position: 58.63, 58.81, 47.41
[INFO] [1742825259.563881]: current position: 56.12, 47.36, 44.99
[INFO] [1742825260.564356]: current position: 50.92, 55.07, 55.43
[INFO] [1742825261.563809]: current position: 44.31, 45.98, 42.95
[INFO] [1742825262.564163]: current position: 55.16, 48.15, 40.69
[INFO] [1742825263.563757]: current position: 59.07, 52.93, 45.50
[INFO] [1742825264.563570]: current position: 44.53, 58.86, 51.53
[INFO] [1742825265.563535]: current position: 56.63, 59.00, 50.42
[INFO] [1742825266.564108]: current position: 44.24, 41.79, 56.45
[INFO] [1742825267.563722]: current position: 48.21, 54.47, 42.00
[INFO] [1742825268.565277]: current position: 50.22, 58.47, 53.08
[INFO] [1742825269.564091]: current position: 58.63, 53.68, 56.38
[INFO] [1742825270.564378]: current position: 56.46, 45.86, 54.01
[INFO] [1742825271.563679]: current position: 42.34, 46.61, 50.09
[INFO] [1742825272.564126]: current position: 57.19, 54.23, 59.65
[INFO] [1742825273.563739]: current position: 56.50, 45.68, 55.43
leetaehun@leetaehun-VirtualBox: ~
File Edit View Search Terminal Help
leetaehun@leetaehun-VirtualBox:~$ rostopic info /drone_position
Type: geometry_msgs/Point
Publishers:
* /drone_16548_1742825252424 (http://leetaehun-VirtualBox:43421/)
Subscribers:
* /monitor_16529_1742825251386 (http://leetaehun-VirtualBox:33941/)
leetaehun@leetaehun-VirtualBox:~$

2. 아래 요구사항을 참고하여, 실습수업 때 작성한 my_msg 패키지를 수정하여 실행한 결과를 보이시오.

■ 구현 요구사항

- **launch 파일 작성 (m_send.launch 파일을 복사한 후 수정하여 m_send_param.launch 파일 작성)**
  - ✓ <node> 태그 안에 <param> 태그를 추가하여 my_msg 메시지 형식 필드 중 아래 내용의 값을 launch 파일에서 지정해서 실행하도록 함.
    - ○ string first_name, string last_name, string phone_number, int32 id_number
  - ✓ 동일한 코드를 사용하는 2개의 노드를 서로 다른 이름으로 실행하도록 설정(예시 화면 참고)
    - ○ 1번째 노드('sender1'): **자기 자신**의 이름, 성, 핸드폰 번호, 학번 정보를 전송
    - ○ 2번째 노드('sender2'): 스마트 모빌리티 설계 과목에 대한 정보를 전송
- **publisher 코드 작성 (msg_send.py 파일을 복사, 수정하여 msg_send_param.py 파일 작성)**
  - ✓ rospy.get_param() 함수를 이용해 launch 파일에서 지정한 파라미터 값을 publish 할 수 있도록 수정
- **rosbag 파일 (hw2_2.bag) 생성**
  - ✓ m_send_param.launch를 실행한 후 "msg_to_xycar" 토픽 데이터를 hw2_2.bag 파일에 기록
  - ✓ 명령어: rosbag record -O hw2_2.bag msg_to_xycar
  - ✓ 기록한 데이터에는 최소 10줄 이상의 내용이 포함되어야 함.

■ 힌트

- 실습 때 다루었던 Custom message 활용 부분과 launch 기능 활용시 <param> 태그 사용 참조

■ 예시 실행 출력 결과

■ 과제 제출 양식

- 아래와 같이 폴더를 만들고 압축하여 [학번_이름_과제번호].tar로 teams에 제출

```
2020430050_HongGildong_HW2
  └HW2_1
      └drone.py
      └monitor.py
      └실행화면 캡쳐본(png, jpg)
      └hw2_1.bag
  └HW2_2
      └m_send_param.launch
      └msg_send_param.py
      └실행화면 캡쳐본(png, jpg)
      └hw2_2.bag
```

- 실행화면 캡쳐본 요구사항은 다음과 같음.(예시 실행 결과 참고)
  - ✓ 1번: 4개의 터미널이 모두 한 화면에 나오도록 캡쳐(한 화면에 나오기 어렵다면 각 터미널을 캡쳐한 후 여러 개의 캡쳐본 제출)
    - ○ 터미널 #1: roscore 실행
    - ○ 터미널 #2: drone.py 실행
    - ○ 터미널 #3: monitor.py 실행
    - ○ 터미널 #4: rostopic info /drone_position 실행
  - ✓ 2번: roslaunch my_msg m_send_param.launch 실행 후 캡쳐