There are 2 parts to the problem. First is to generate the IMU data and the second to publish it as time varies. The values of the angular and linear acceleration must vary with time.

So, I first defined a function called imu_data_generator() which takes time as a parameter. Basically, this function instantiates the class Imu from the sensor_msgs module. This will help us store the data in Imu format.

Then we set the values of linear acceleration and angular velocity using the Vector3 module. Here, we use the time parameter passed to the function to make the values vary with time. Essentially this function will only generate one value at a time but we will call it from the publisher function in a time dependent loop. I have set the values of quaternion to a default value.

The function publisher_script() is similar to how we defined publishers in the earlier task. Here, as stated earlier we will call the imu_data_generator() and publish Imu data in a while loop till time reaches 10 seconds.

A problem here is that I am not able to publish the position. From what I read about imu, we cannot directly get the position of the object from imu since we don't get the initial position, linear velocities. One way to find position is to use another sensor like GPS and then combine data from both to get position. This combining process can be done with the help of Kalman filter. I haven't really understood how Kalman filter works even though I tried to go through online resources but it basically uses a series of measurements done over time to find the estimate for position.

References:

1. sensor_msgs/Imu Documentation
2. geometry_msgs/Vector3 Documentation
3. std_msgs/Header Documentation
4. IMU et robotique : ce qu'il faut connaître
5. Get the current time - ROS Answers: Open Source Q&A Forum
6. ROS/Tutorials/WritingPublisherSubscriber(python) - ROS Wiki
7. https://en.wikipedia.org/wiki/Kalman_filter