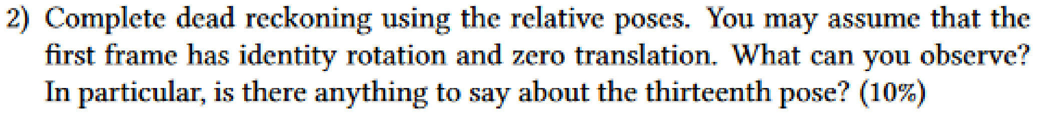
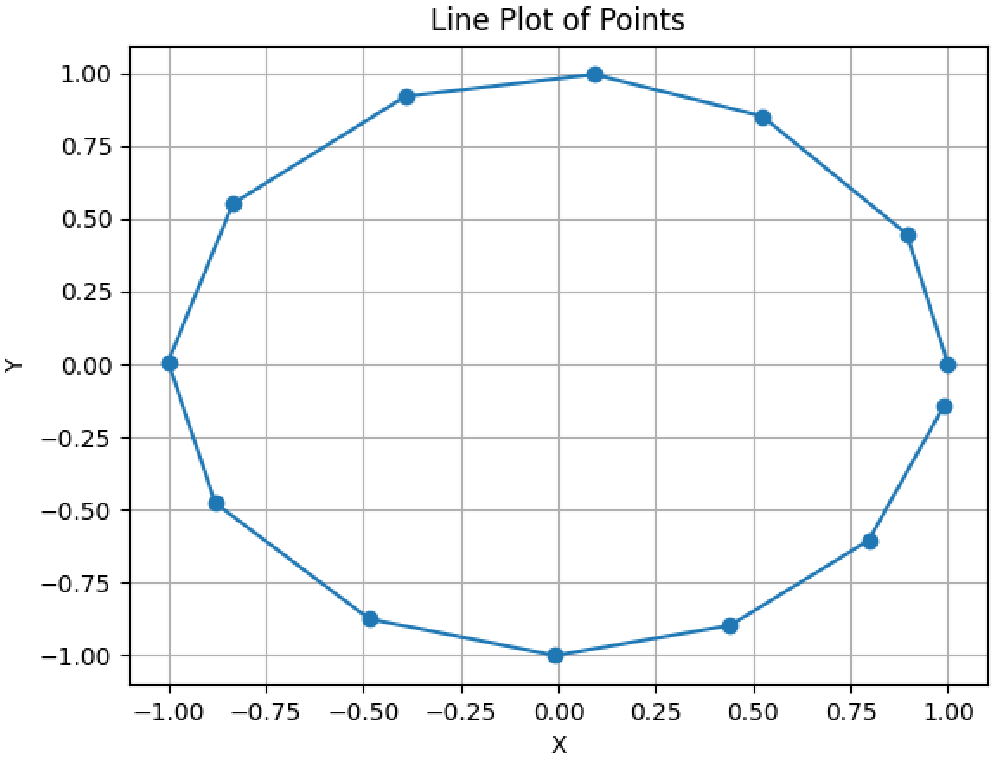
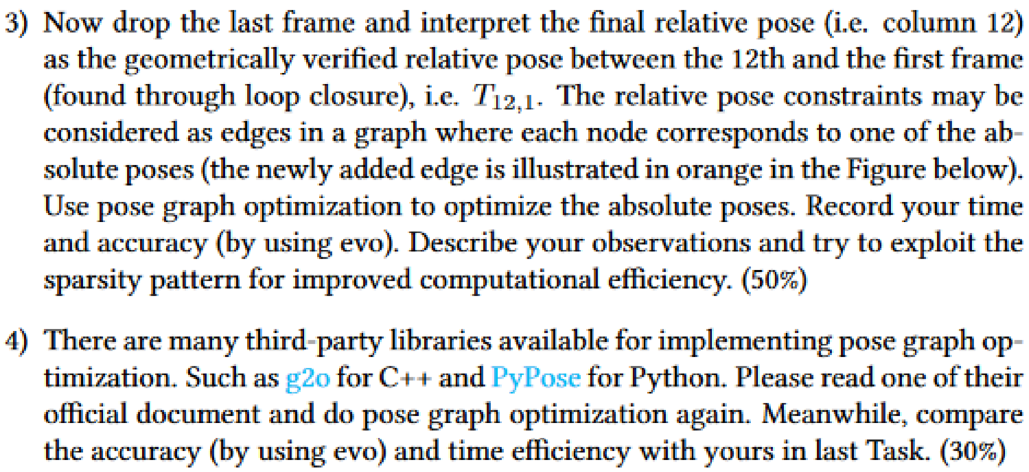


2) Complete dead reckoning using the relative poses. You may assume that the first frame has identity rotation and zero translation. What can you observe? In particular, is there anything to say about the thirteenth pose? (10%)

See Attachment Code :question2.cpp

I observed that the points form an approximate circle, but the thirteenth point does not coincide with the first point, as shown in the image.



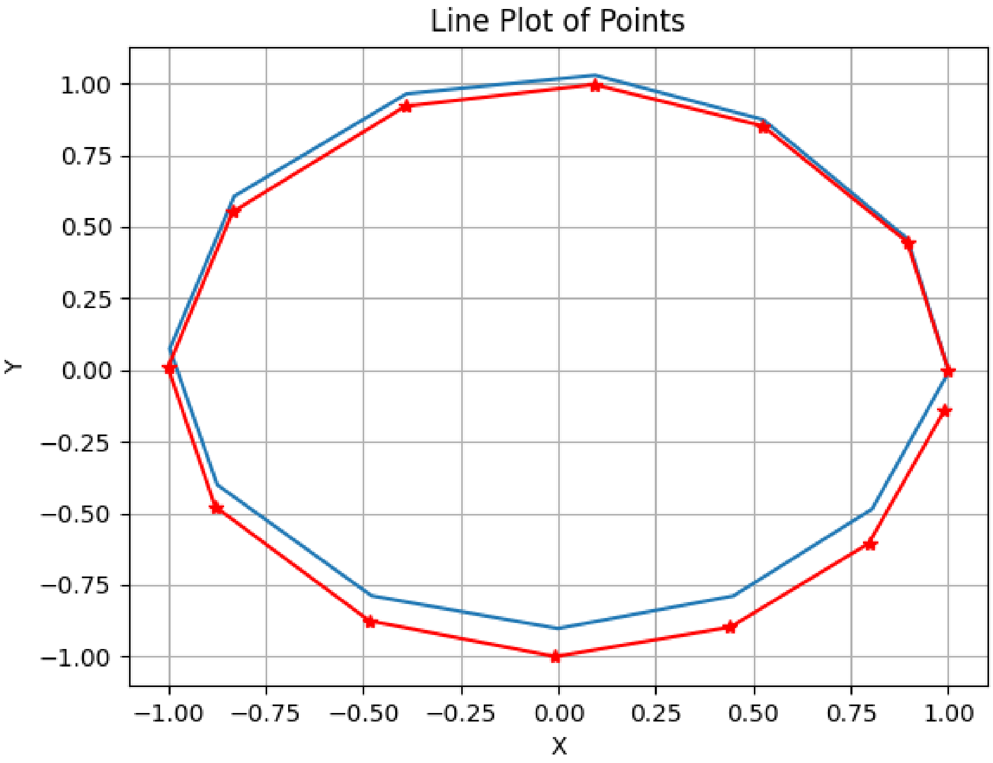


3) Now drop the last frame and interpret the final relative pose (i.e. column 12) as the geometrically verified relative pose between the 12th and the first frame (found through loop closure), i.e. $T_{12,1}$. The relative pose constraints may be considered as edges in a graph where each node corresponds to one of the absolute poses (the newly added edge is illustrated in orange in the Figure below). Use pose graph optimization to optimize the absolute poses. Record your time and accuracy (by using evo). Describe your observations and try to exploit the sparsity pattern for improved computational efficiency. (50%)

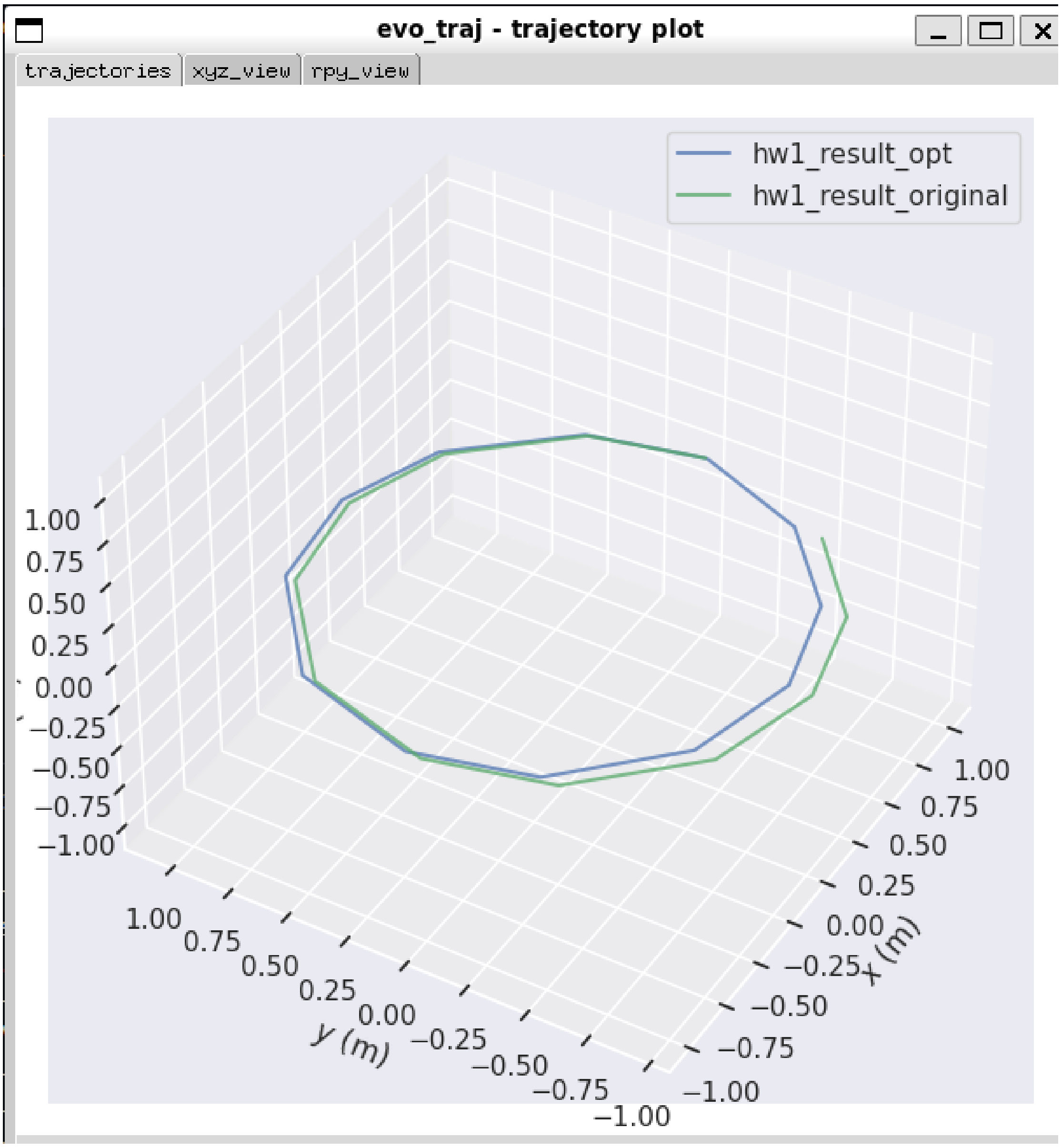
4) There are many third-party libraries available for implementing pose graph optimization. Such as g2o for C++ and PyPose for Python. Please read one of their official document and do pose graph optimization again. Meanwhile, compare the accuracy (by using evo) and time efficiency with yours in last Task. (30%)

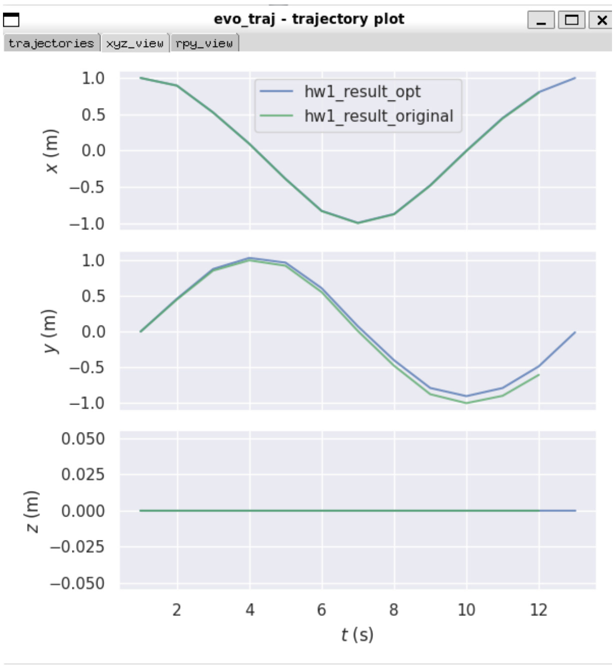
See Attachment Code :question3_4.cpp

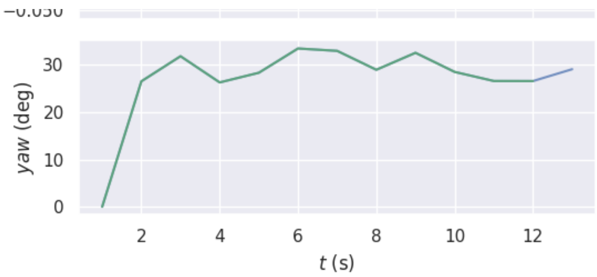
I observed that the points form an approximate circle, but the thirteenth point does not coincide with the first point, as shown in the image.



Evo plot:


evo_traj - trajectory plot
trajectories
xyz_view
rpy_view
hw1_result_opt
hw1_result_original
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
y (m)
x (m)


evo_traj - trajectory plot
trajectories
xyz_view
rpy_view
hw1_result_opt
hw1_result_original
x (m)
1.0
0.5
0.0
−0.5
−1.0
y (m)
1.0
0.5
0.0
−0.5
−1.0
z (m)
0.050
0.025
0.000
−0.025
−0.050
2
4
6
8
10
12
t (s)


−0.050
yaw (deg)
30
20
10
0
2
4
6
8
10
12
t (s)