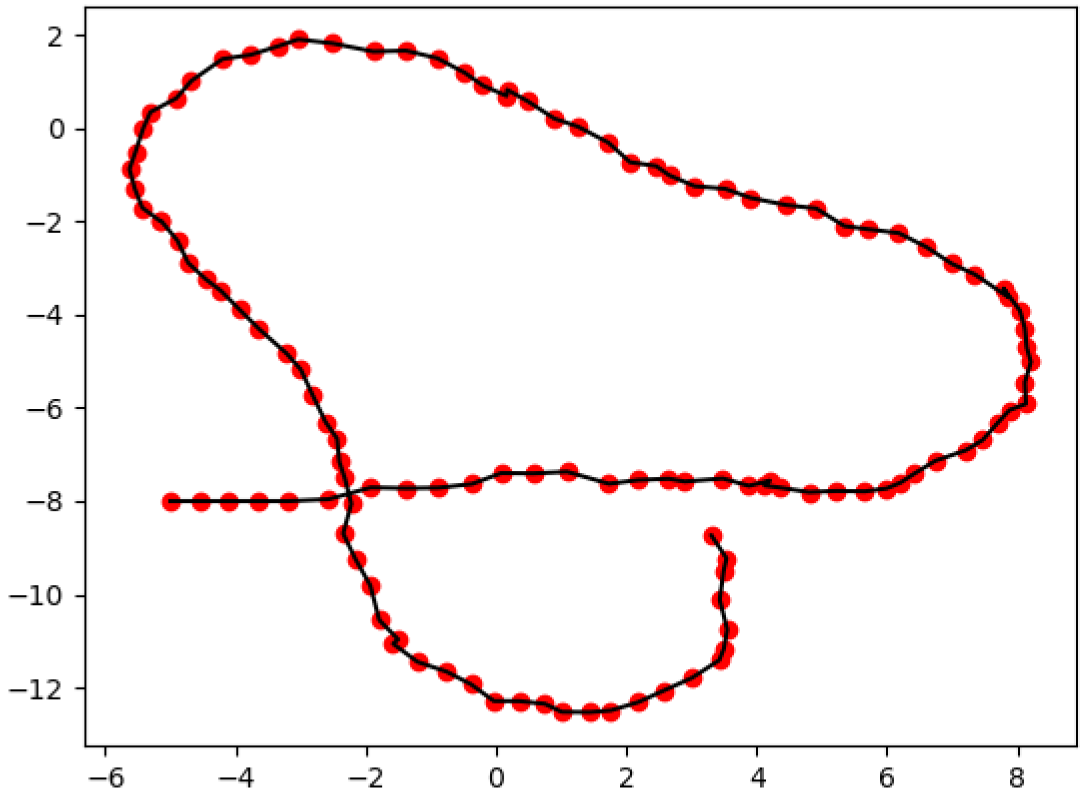
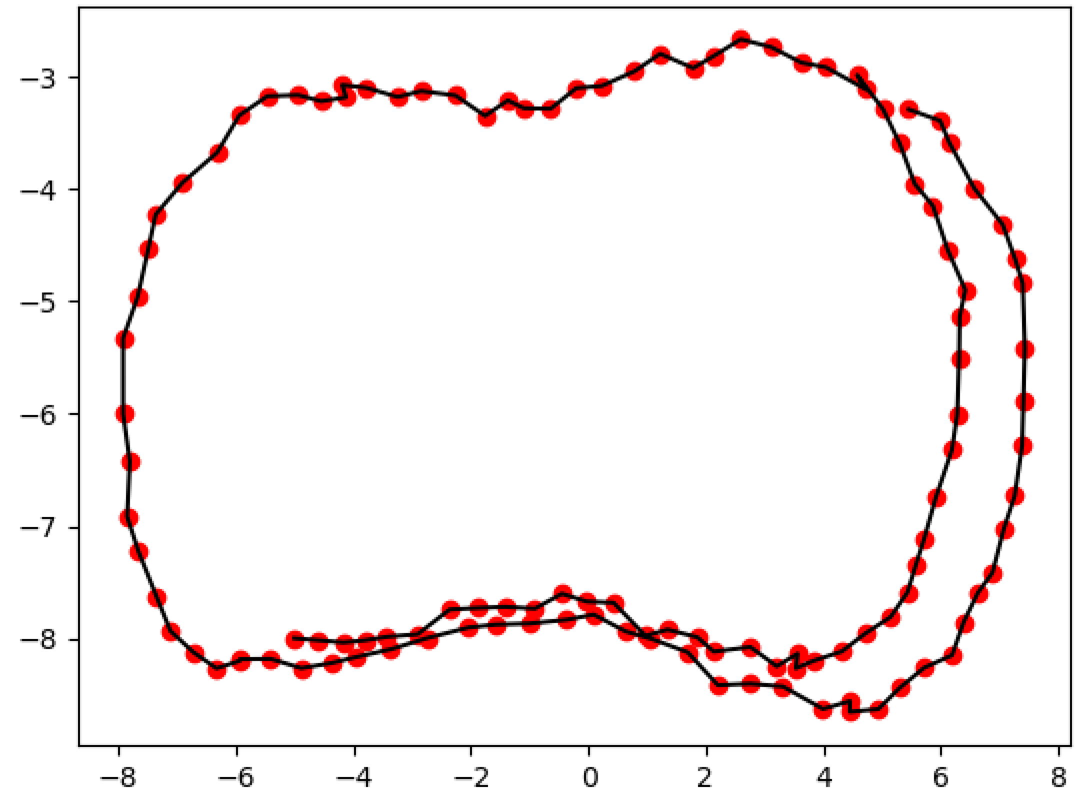
[Trajectory Evaluation using G2O]

I fed the following input to g2o:

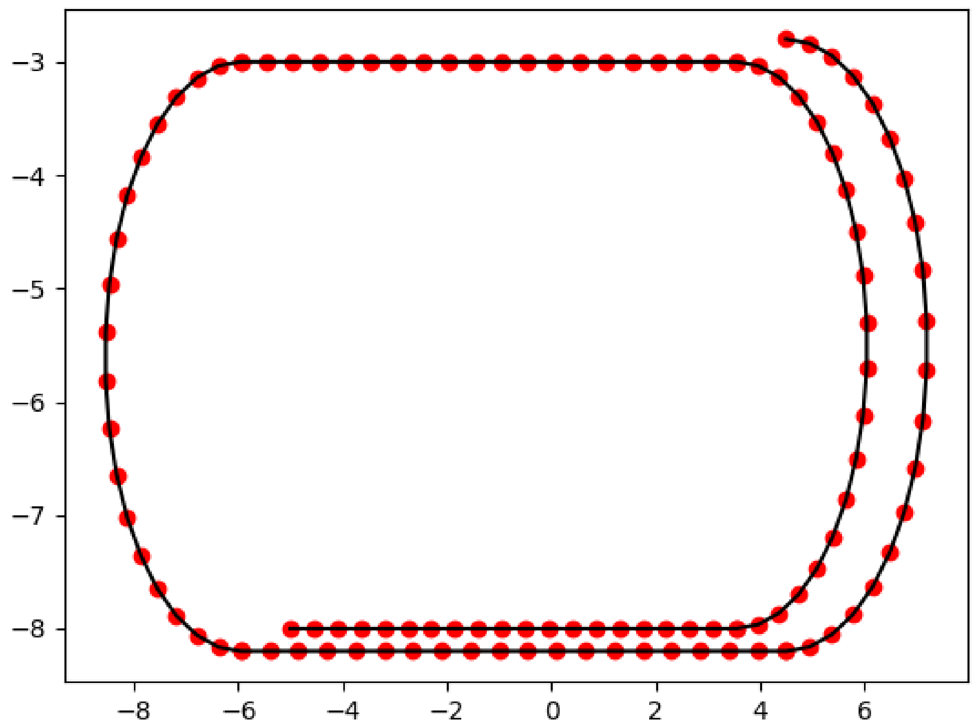
Input:
saved_data/edges-poses.g2o : This file is the one generated by my code. Here, the initial pose value of all the time steps calculated based on odometry edges is also included.



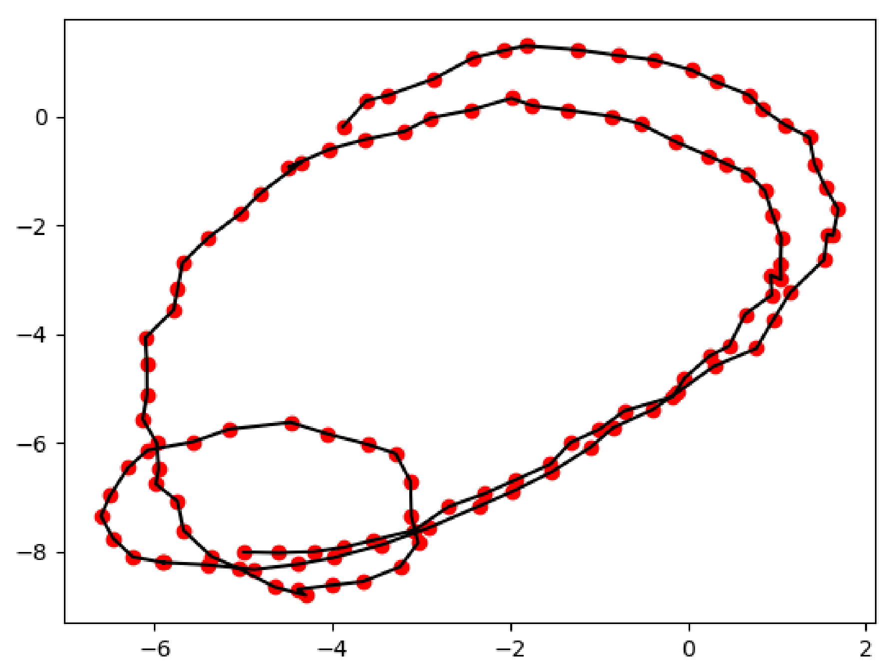
Output: saved_data/official_edges-poses.g2o



Ground truth:

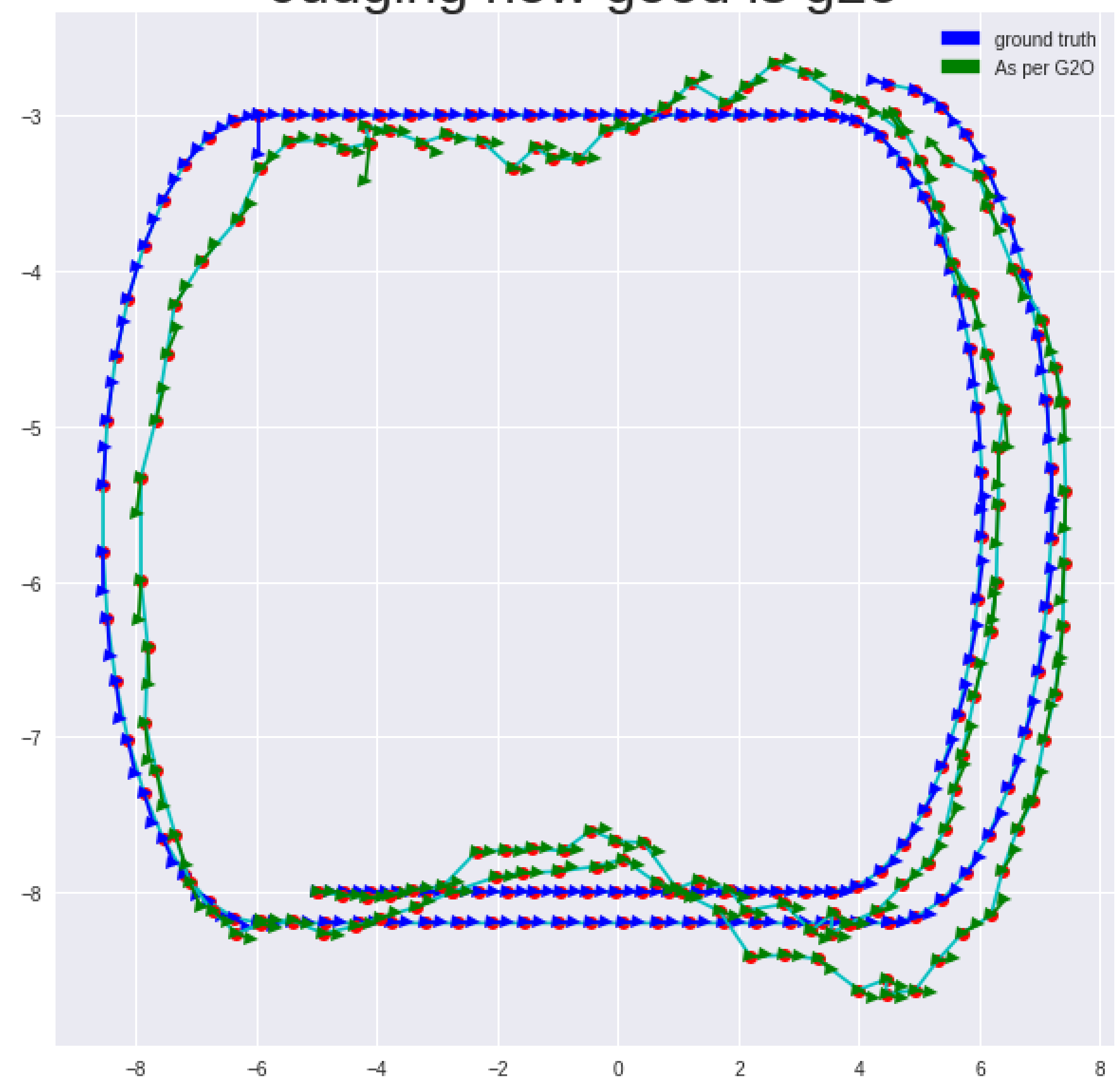


Optimized version by my code: gaussian newton (500,700,1000)



Superimposed:

Judging how good is g2o



EVO based comparison:

ATE/APE measures the difference between ground truth points and estimated trajectory, obtained by an algorithm using sensor data as input. It is referred to as a mandatory performance measure. mean absolute pose error

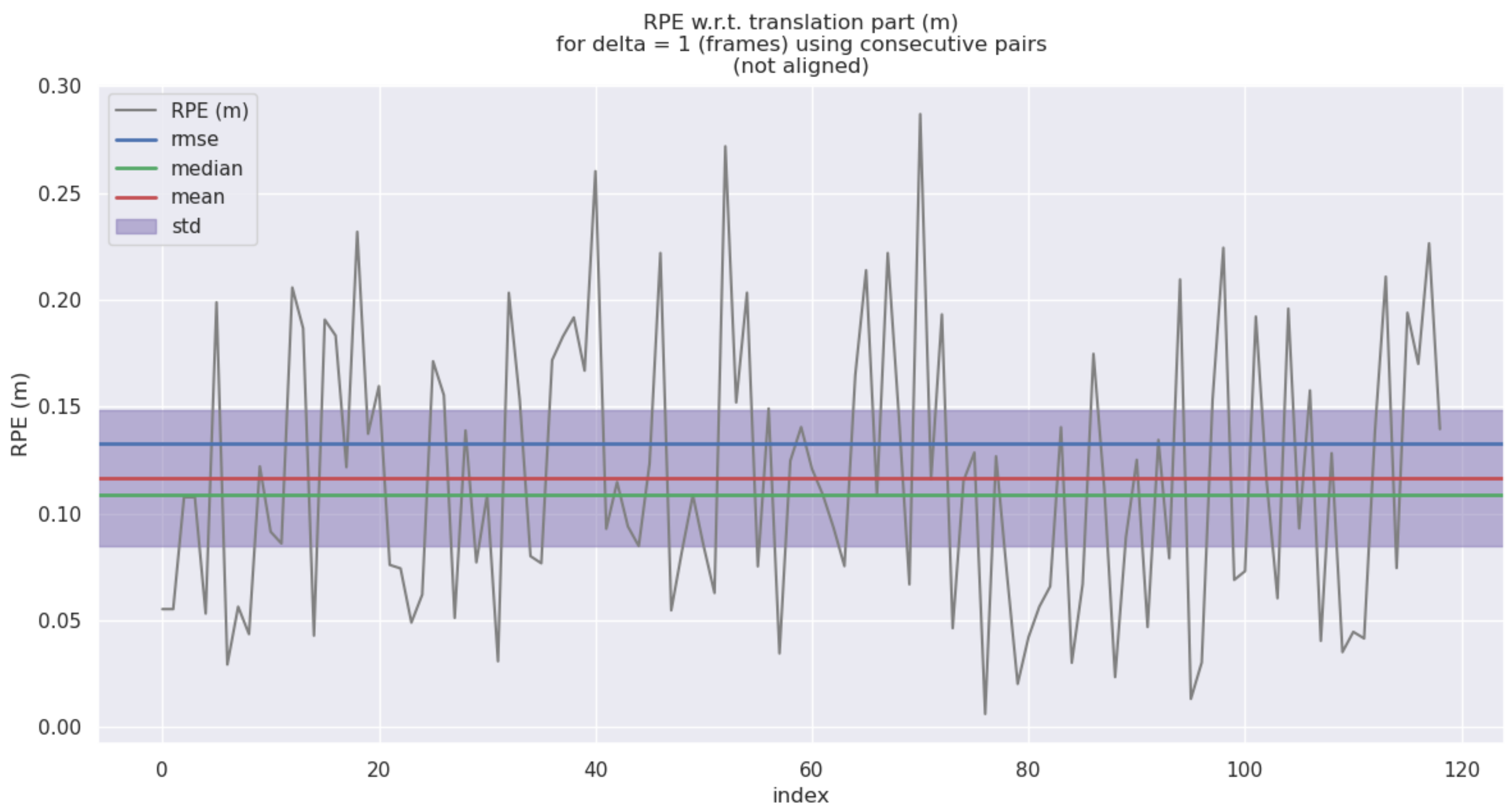
RPE accounts for measuring SLAM result i.e. estimated trajectory accuracy by comparing reconstructed relative transformation between nearby poses to actual relative transformation i.e. ground truth. It is a recommended performance measure. mean absolute pose error

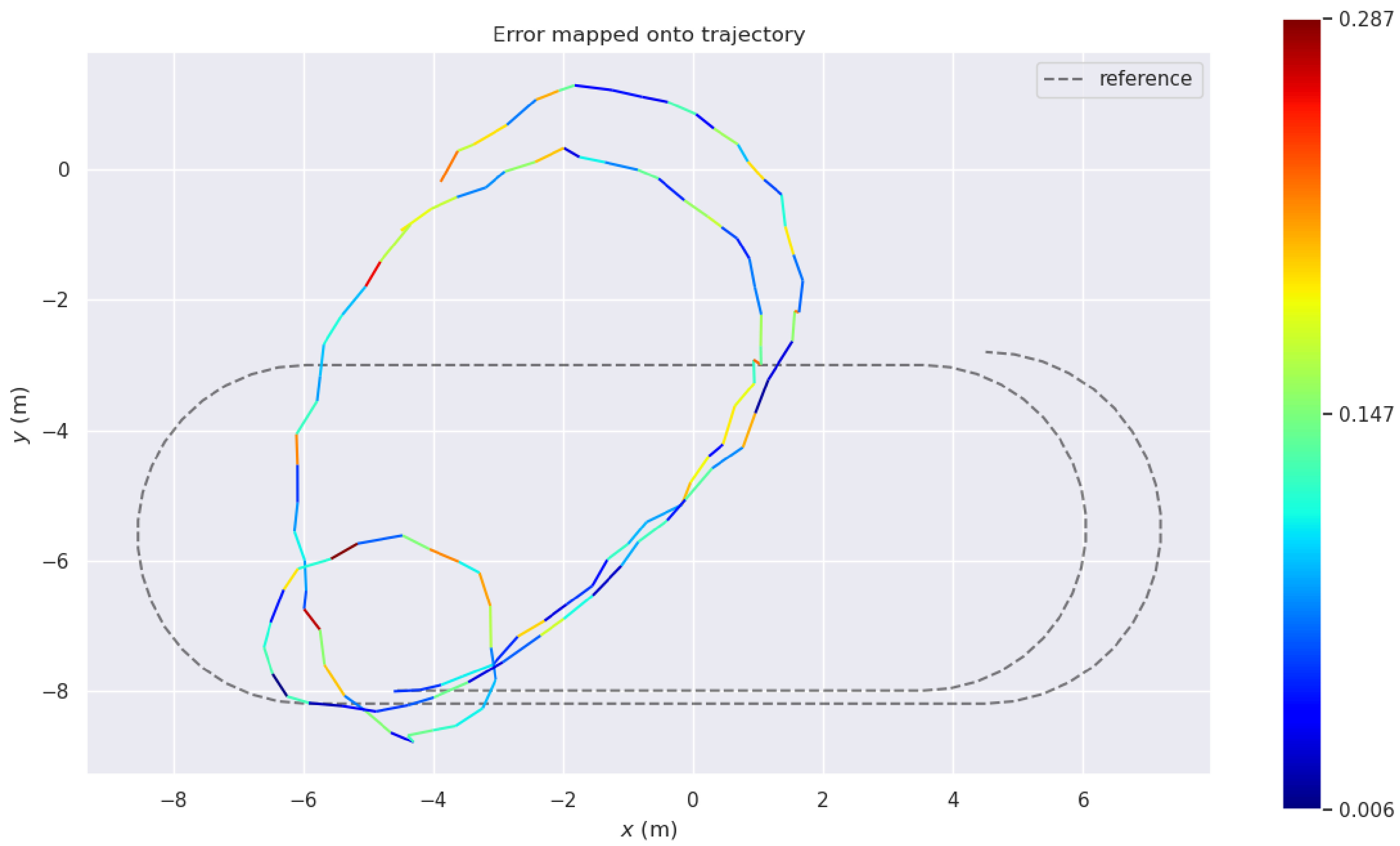
For optimal trajectory by mine code (Gaussian Newton, (500, 700,1000))

RPE
RPE w.r.t. translation part (m)
for delta = 1 (frames) using consecutive pairs
(not aligned)

max	0.286835
mean	0.116624
median	0.108873
min	0.006309

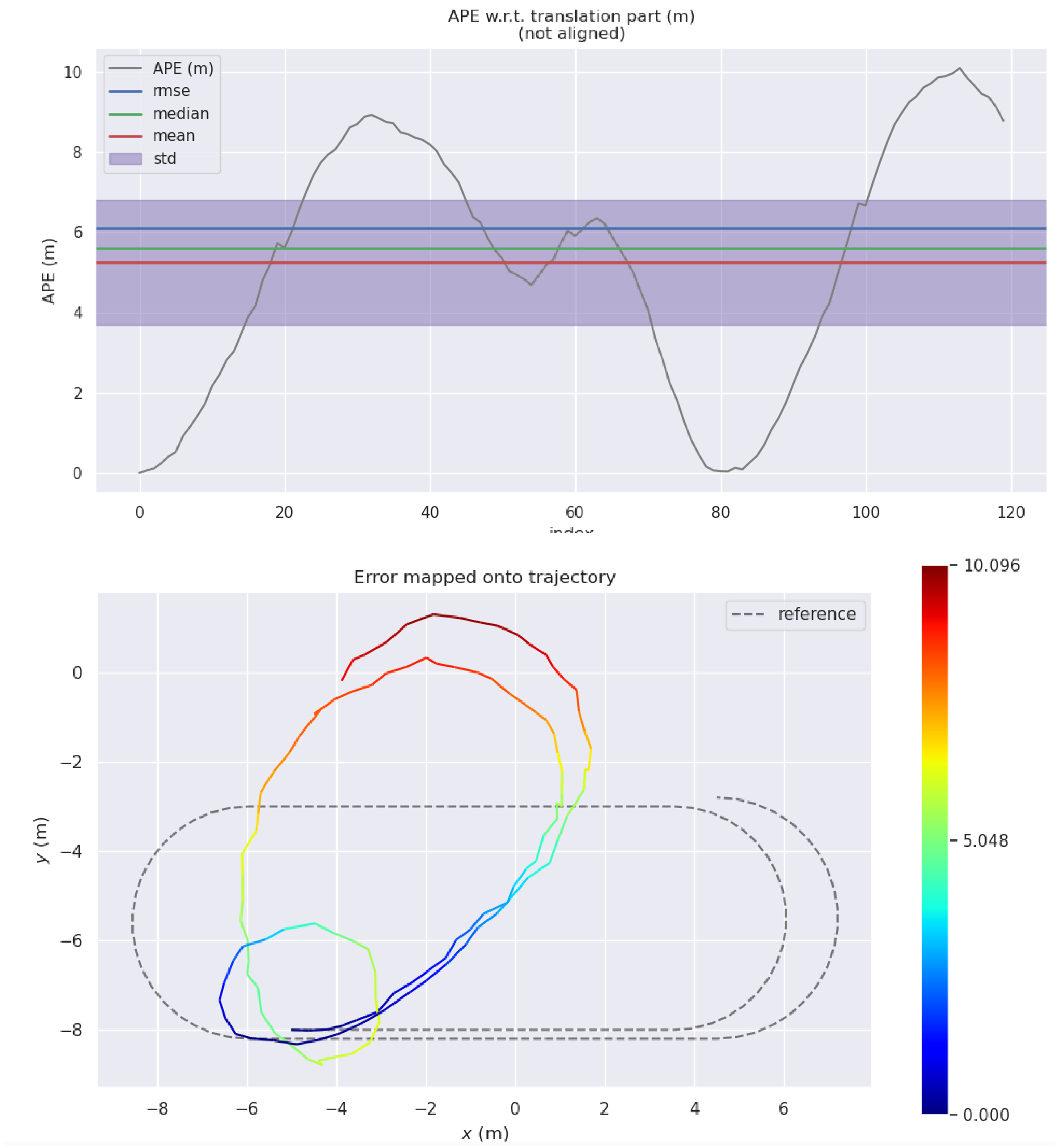
rmse 0.132588
sse 2.091982
std 0.063075



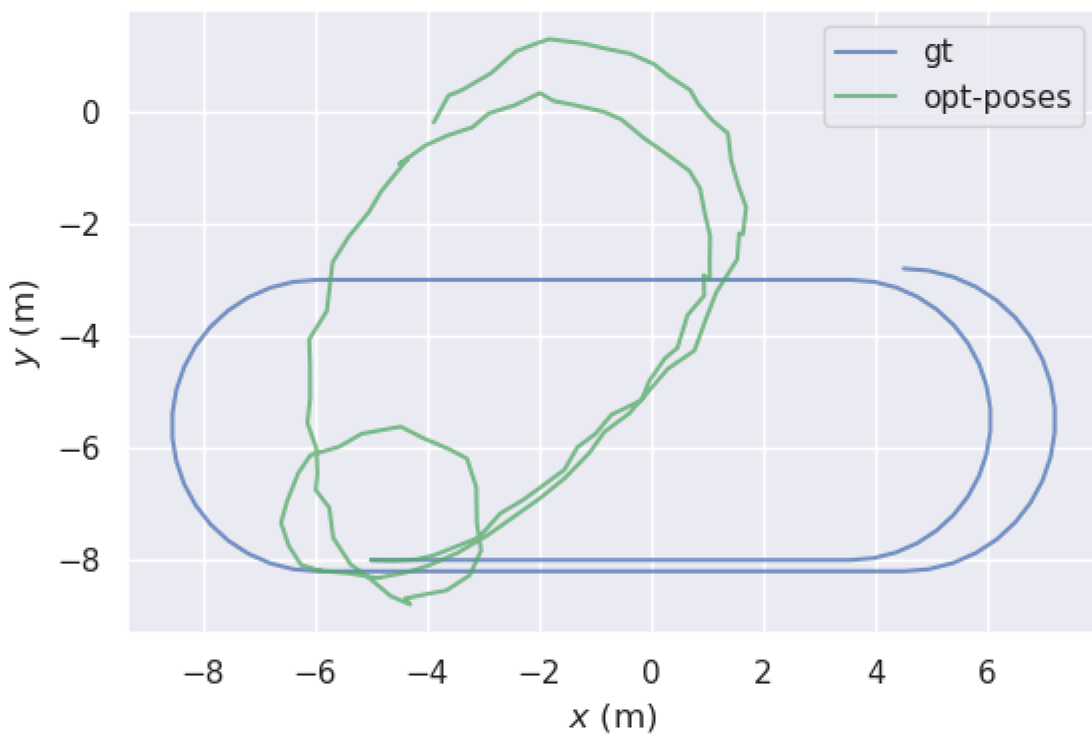


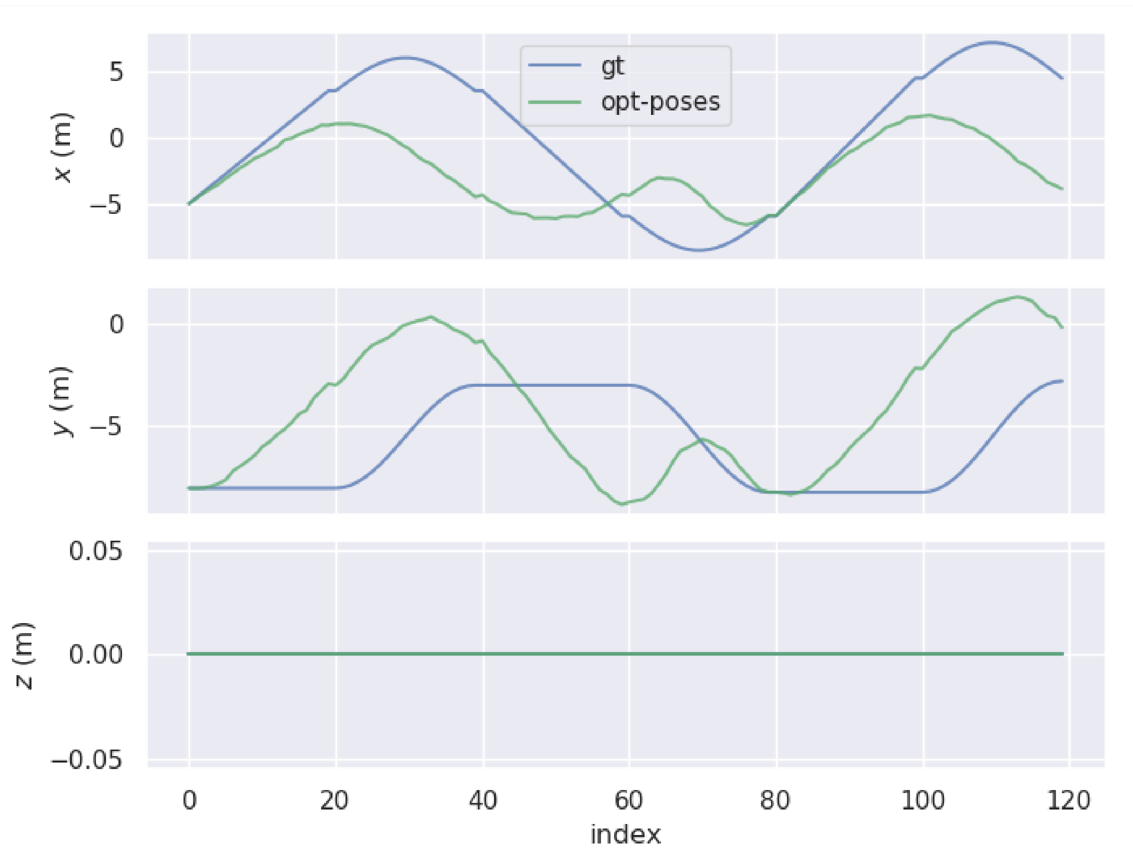
APE
APE w.r.t. translation part (m)
(not aligned)

max 10.096323
mean 5.242878
median 5.600365
min 0.000000
rmse 6.098648
sse 4463.220271
std 3.115403

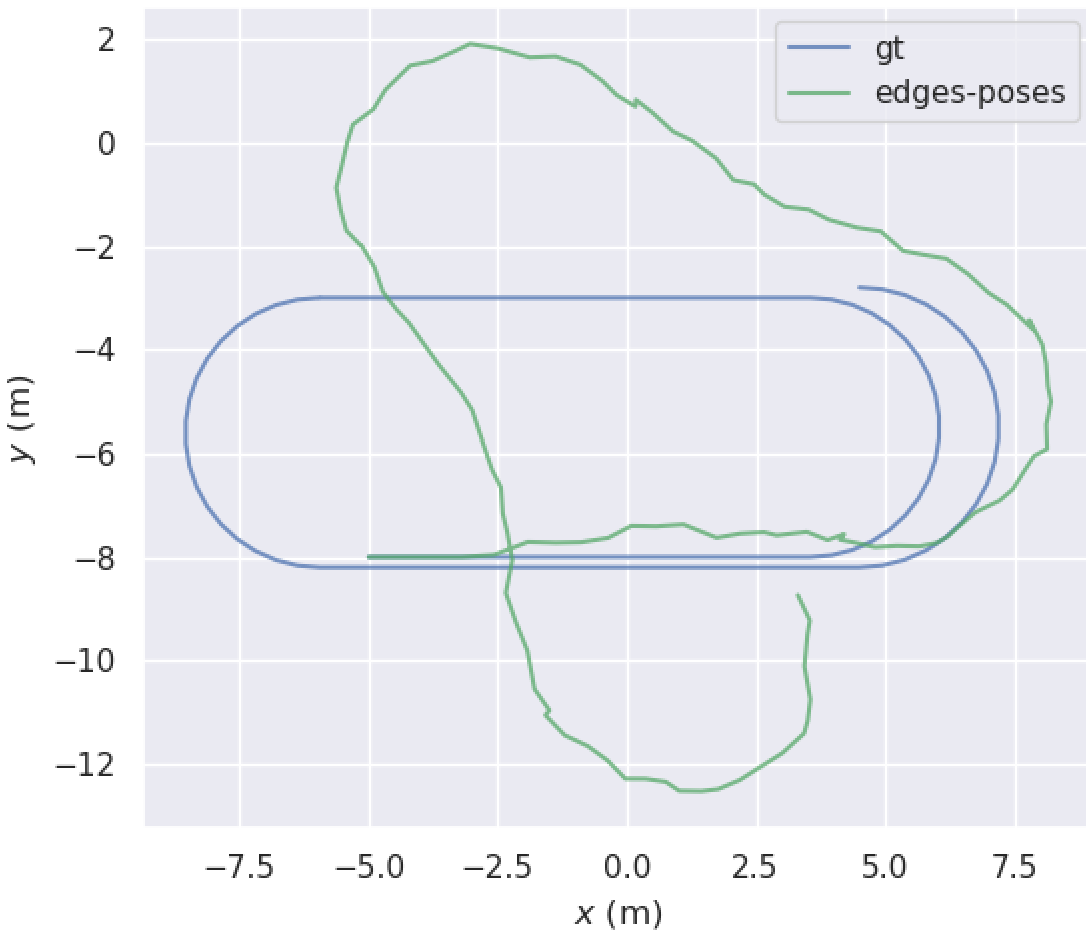


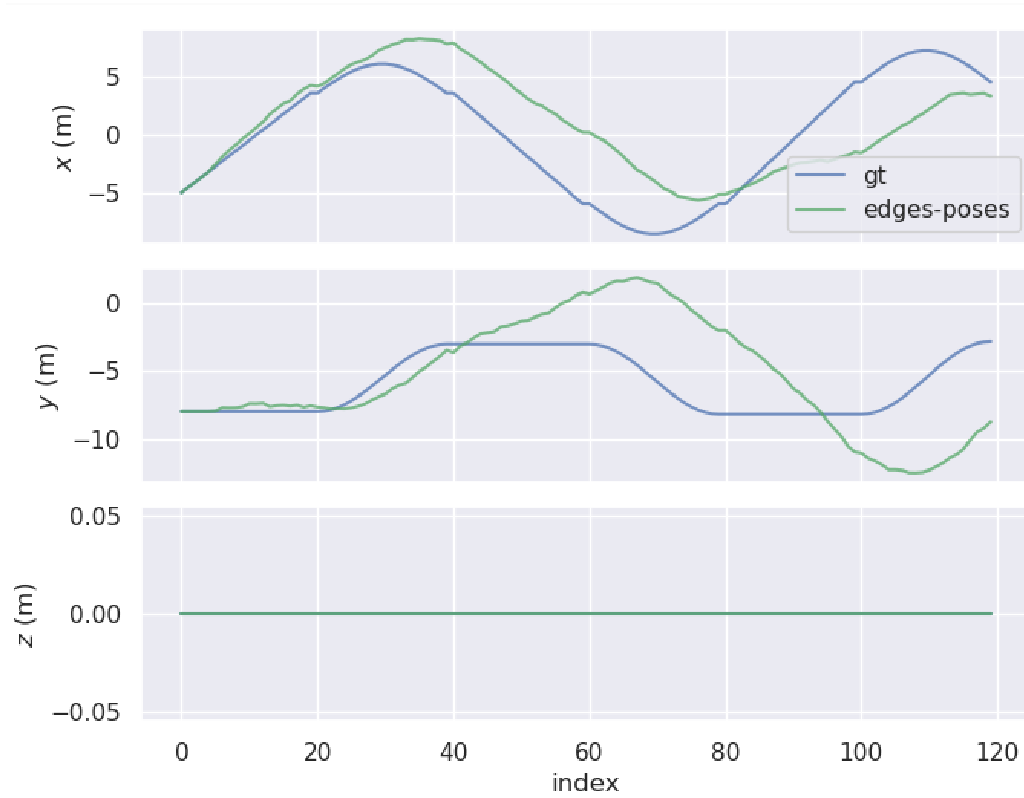
We see that due to loop closure, the error slightly decreases again.


gt
opt-poses
0
−2
−4
−6
−8
y (m)
−8
−6
−4
−2
0
2
4
6
x (m)


gt
opt-poses
5
0
−5
x (m)
0
−5
y (m)
0.05
0.00
−0.05
z (m)
0
20
40
60
80
100
120
index

For NOISY trajectory by ODOM EDGES

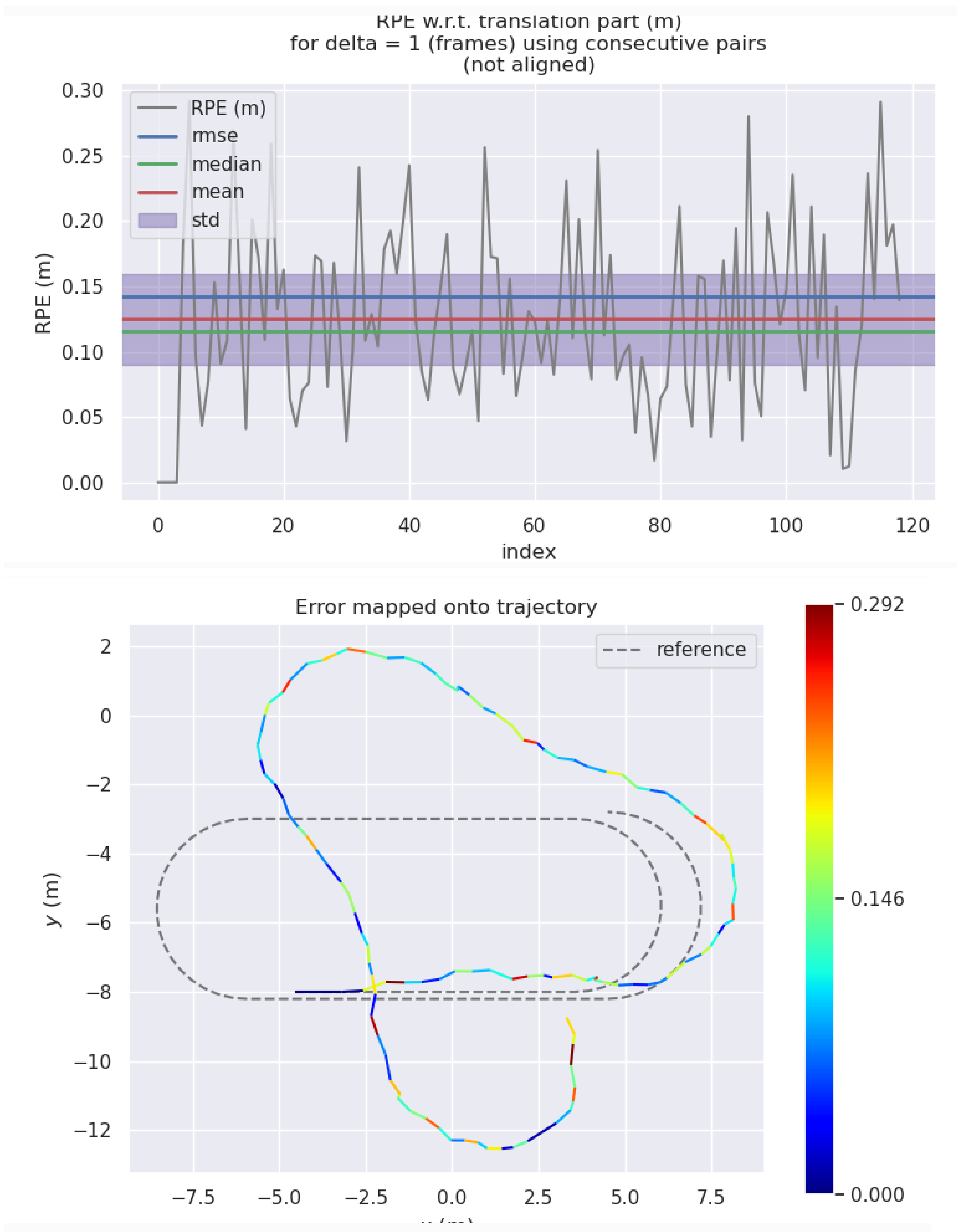




We see that by the end, the drift is very large in both X and Y direction.

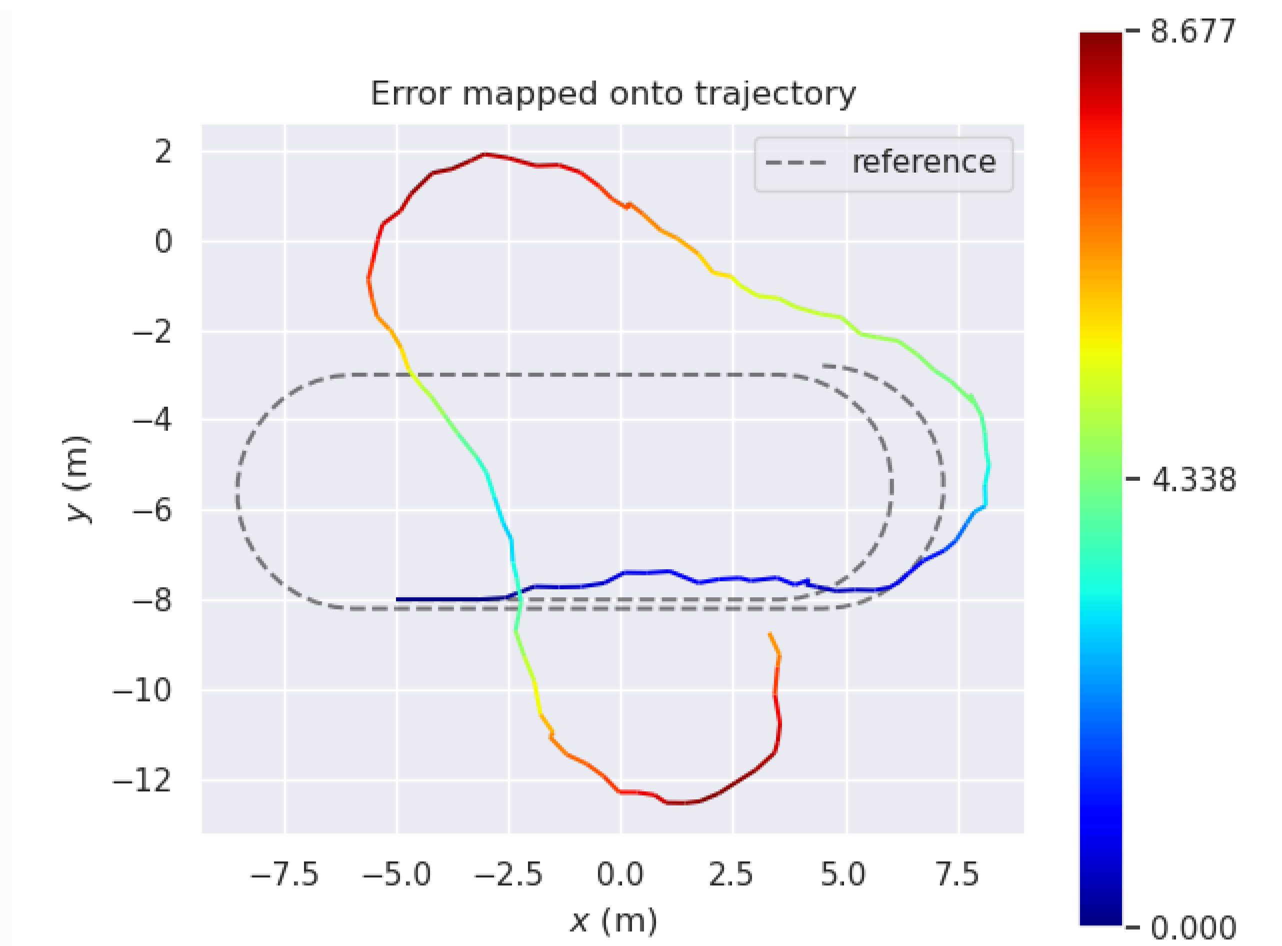
RPE
RPE w.r.t. translation part (m)
for delta = 1 (frames) using consecutive pairs
(not aligned)

max	0.291708
mean	0.124830
median	0.116206
min	0.000000
rmse	0.142610
sse	2.420184
std	0.068958



APE
APE w.r.t. translation part (m)
(not aligned)

max	8.676720
mean	4.665931
median	5.000999
min	0.000000
rmse	5.451923
sse	3566.815386
std	2.820027



EXPERIMENTS (TODO)

- **Try a different confidence matrix with reason why you choose it and observed trend.**

NOTE: It is not good to compare errors as its values depend on the MAGNITUDE OF the absolute values of INFORMATION MATRIX. As a result, I HAD NORMALIZED THE WEIGHTS BEFORE FEEDING THEM. Even then, a direct comparison must be interpreted cautiously as there ARE MORE odom edges than loop closure edges.

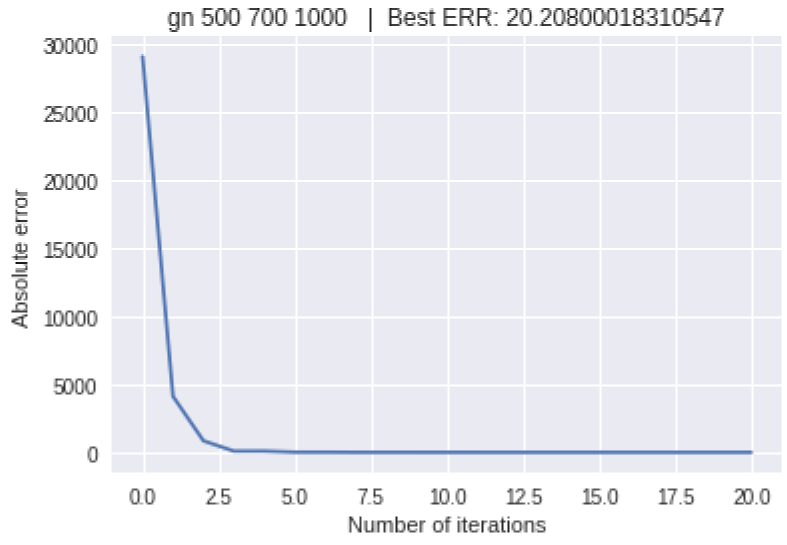
22, 31, 45 Gaussian Newton (Corresponds to 500, 700, 1000)

Reason for choosing: These were the default weights we were asked to plot for.

Best Error: 20

Comments on convergence:

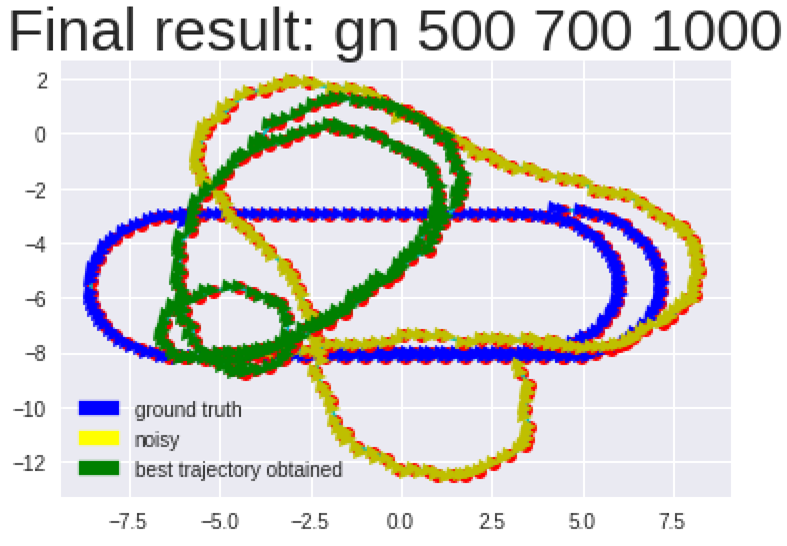
Error converged after iteration 6.



Comments on how trajectory changed over time:

The trajectory changed massively for the first 6 iterations. After that, it remained more or less the same.

Picture:



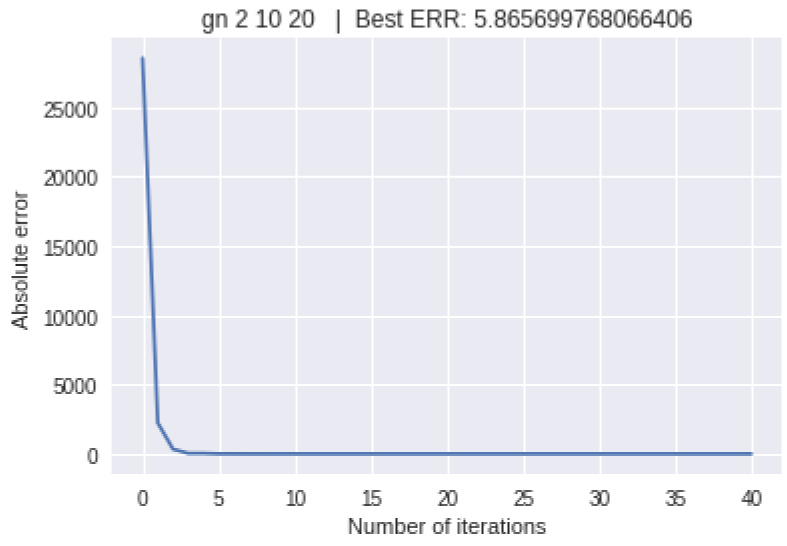
(Odom: 6, LOOP: 31, Anchor: 62) Gaussian Newton

Reason for choosing:

I wanted to see the effect of increasing the loop closure edge weights.

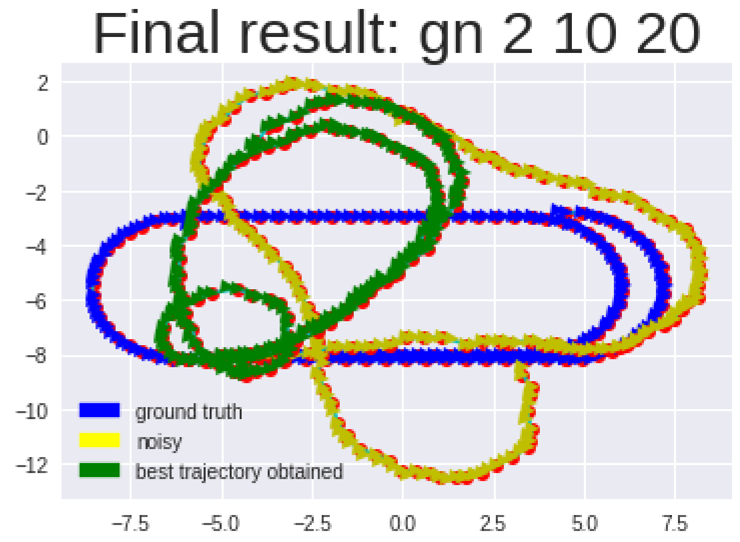
Best Error:5.8

Comments on convergence:



Comments on how trajectory changed over time:
Converged around iteration 10.

Picture:

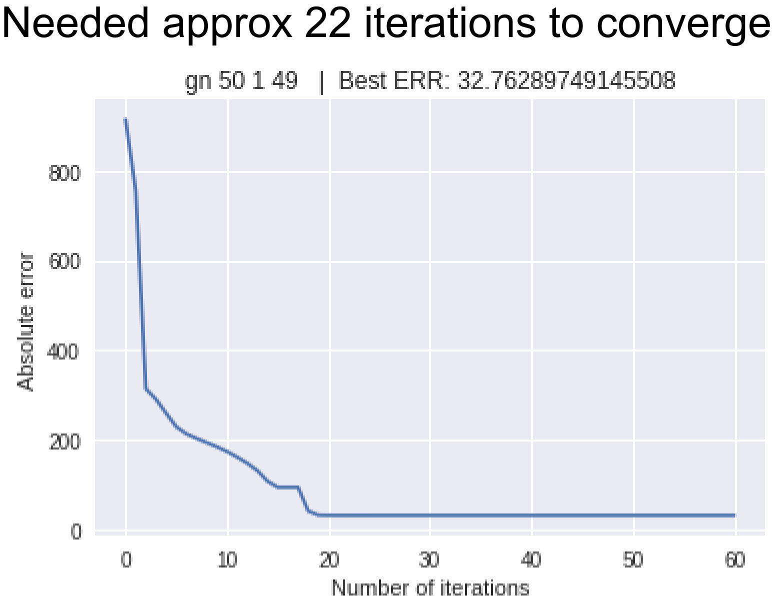


(Odom: 50, LOOP: 1, Anchor: 49) Gaussian Newton

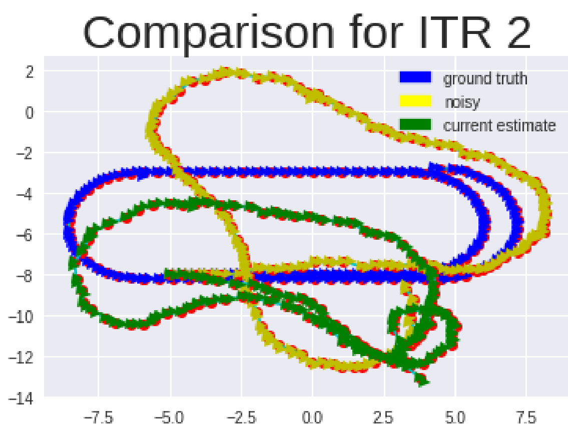
Reason for choosing:
I wanted to see the effect of increasing the odometry constraints edge weights.

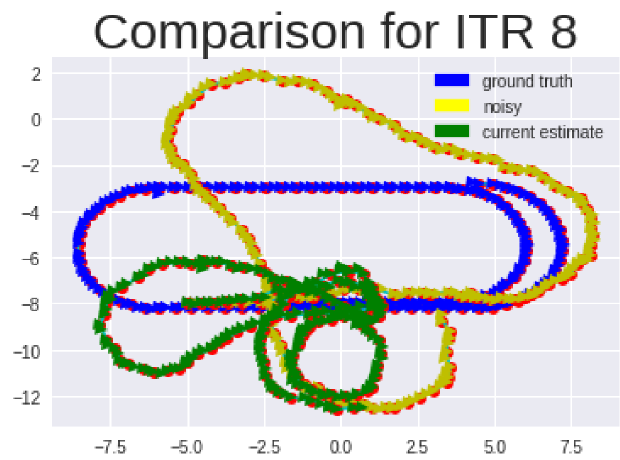
Best Error: 32

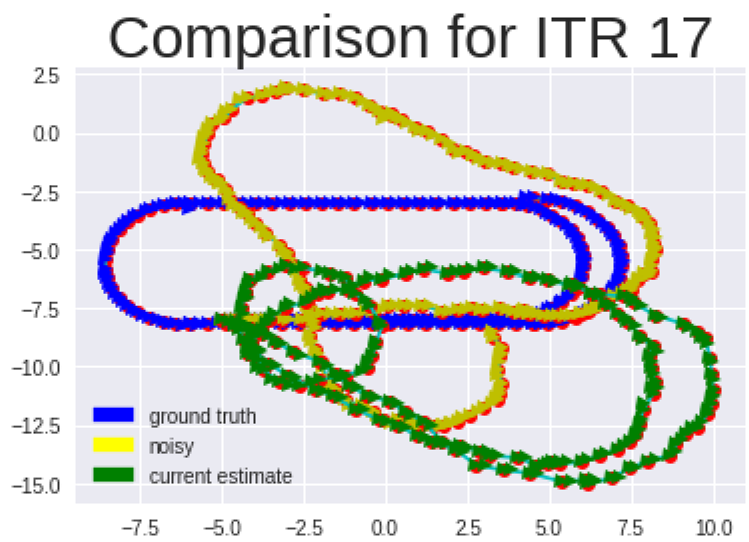
Comments on convergence:
Needed approx 22 iterations to converge.

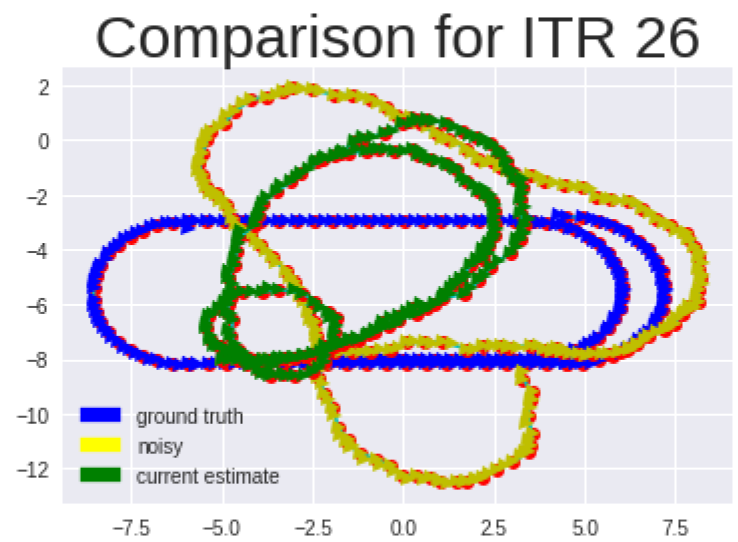


Comments on how trajectory changed over time:
We see that while earlier, the loop closure edges dominate pretty easily, here, we see that it takes the loop closure edges some time to reflect in the trajectory.

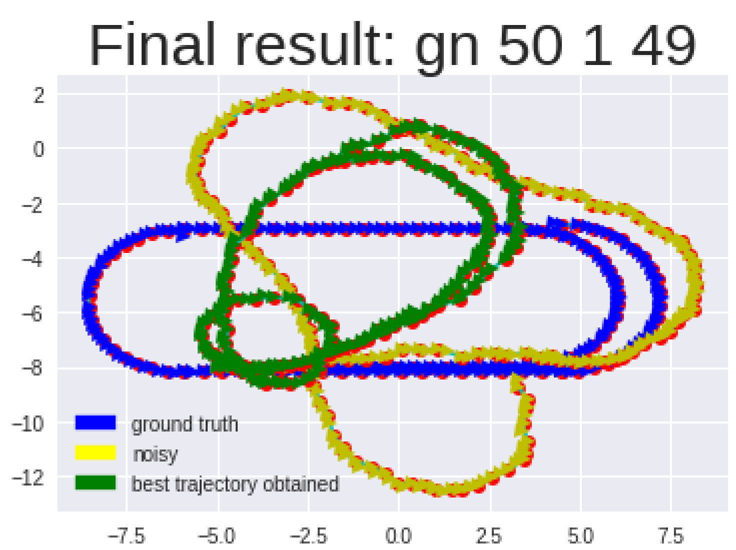








Picture:

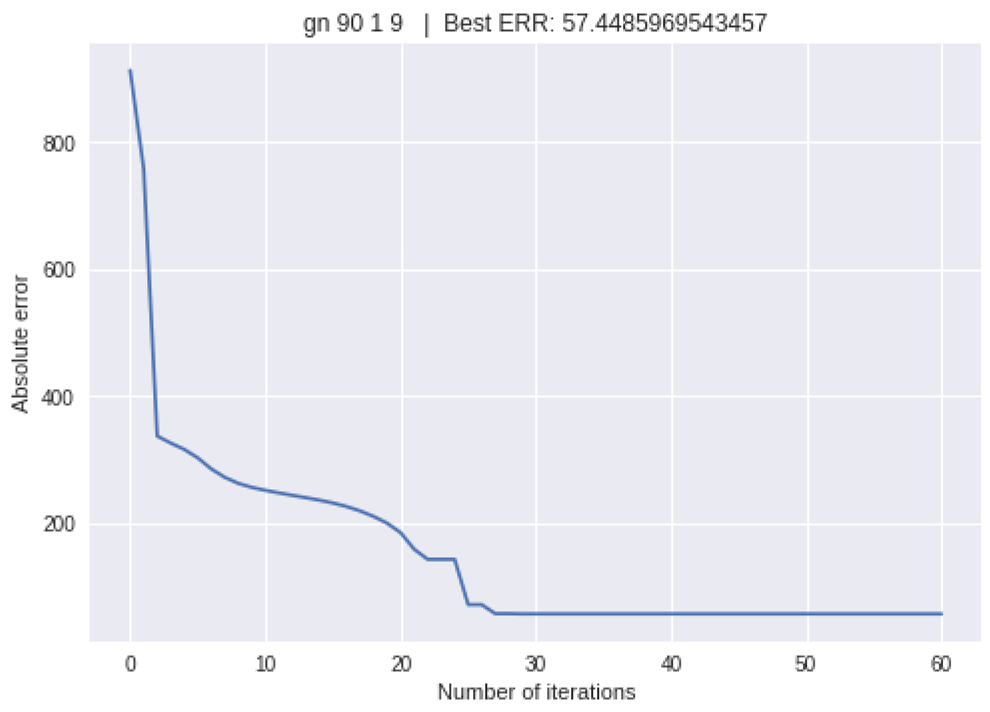


(Odom: 90, LOOP: 1, Anchor: 9) Gaussian Newton

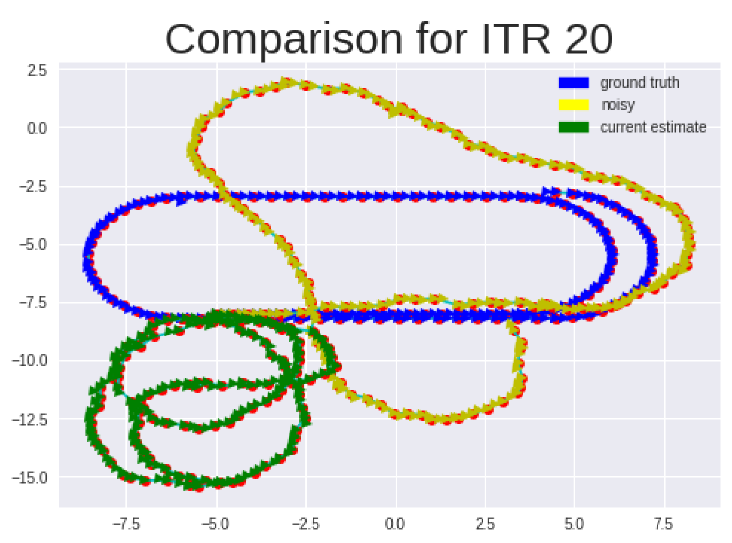
Reason for choosing:
I wanted to see how much I should decrease the loop closure weightage so that the trajectory is similar to the NOISE trajectory we had calculated only on the basis of odometry.

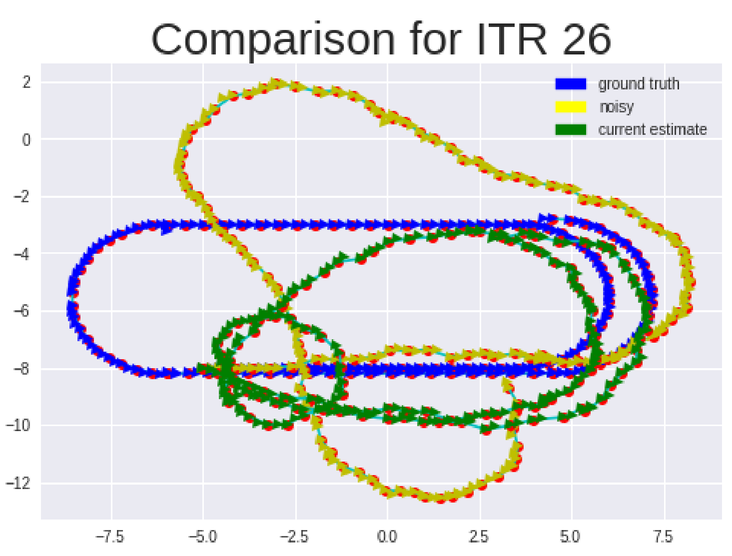
Best Error: 57

Comments on convergence:
Error converges even slower.

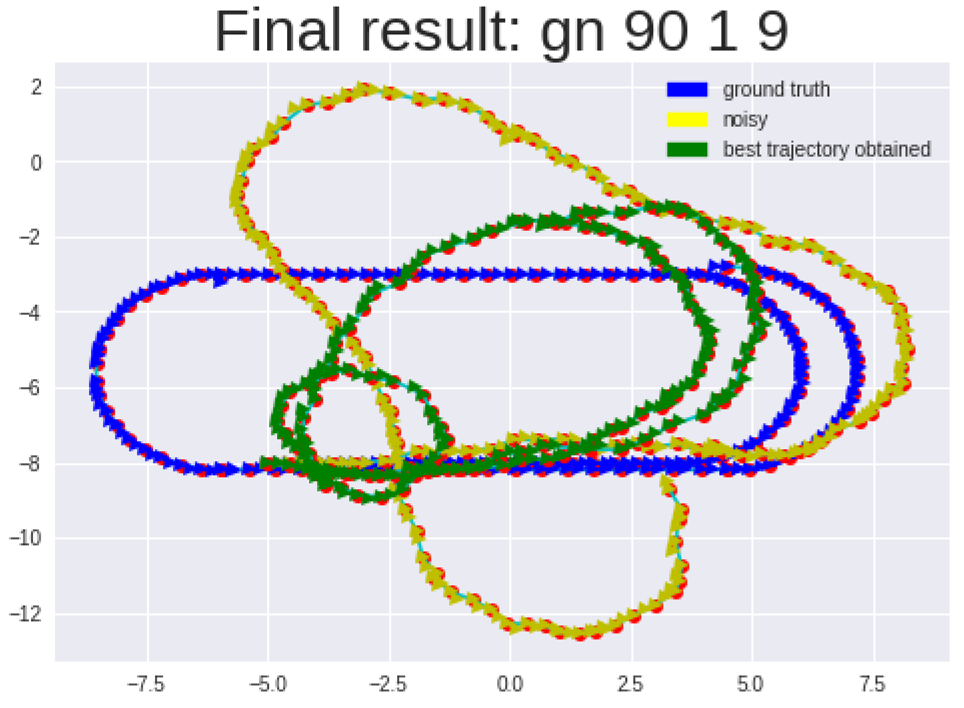


Comments on how trajectory changed over time:





Picture:



(Odom: 1, LOOP: 90, Anchor: 9) Gaussian Newton

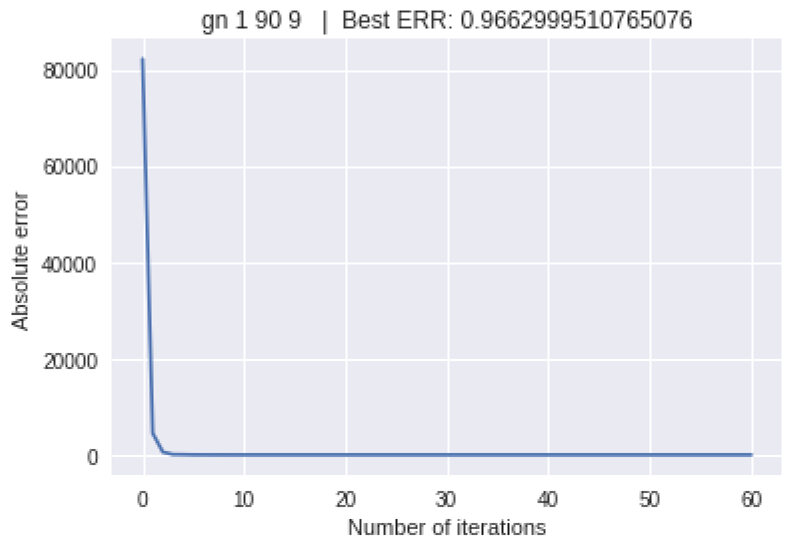
Reason for choosing:

I wanted to see what happens when we negligible weightage to odometry edges. I was interested to know if the initialization due to odometry still helps maintain the same structure.

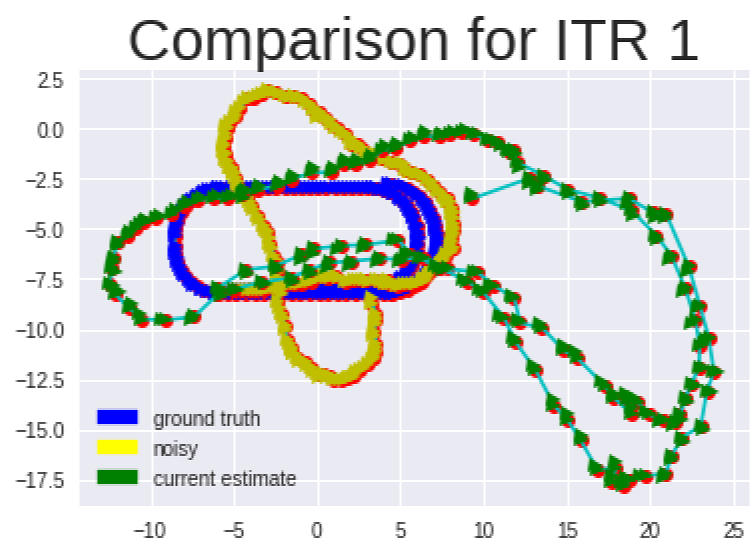
Best Error: 0.9

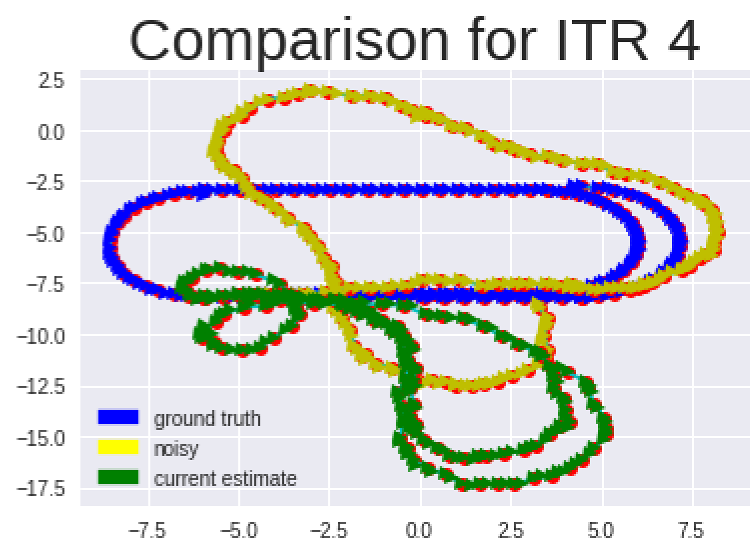
Comments on convergence:

Converges really quickly.

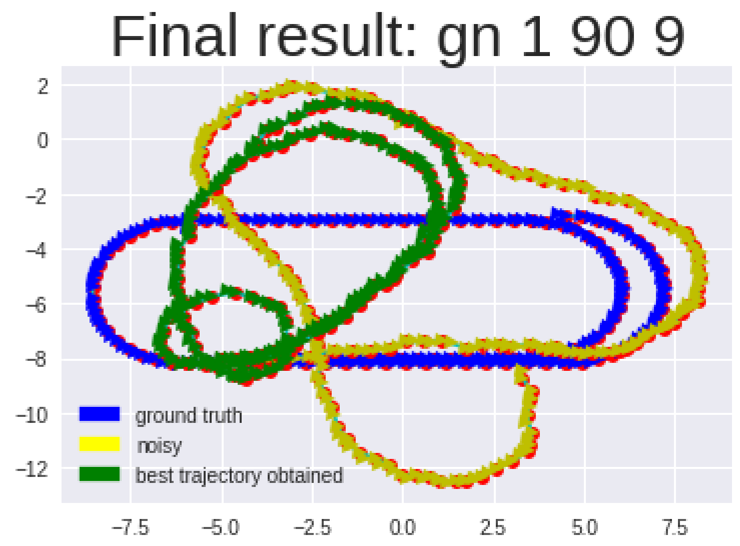


Comments on how trajectory changed over time:





Picture:



Odom: 99, LOOP: 0.009, Anchor: 1) Gaussian Newton

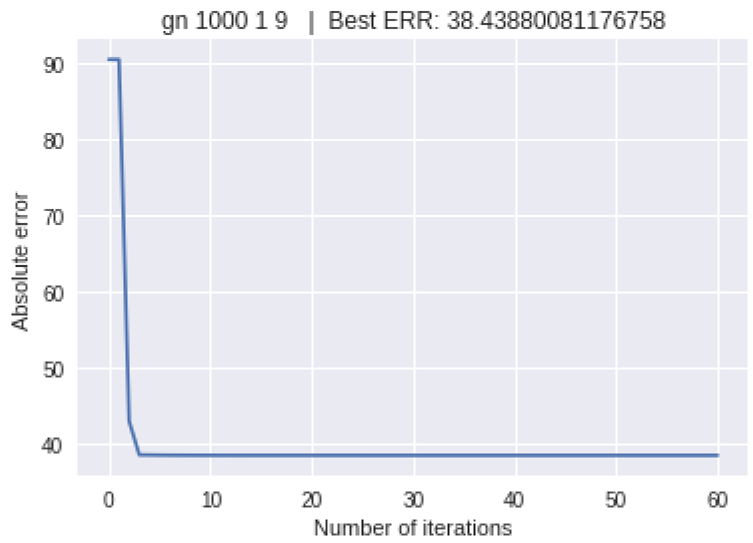
Reason for choosing:

I wanted to see what happens when we negligible weightage to loop closure edges and high weight to odom edges. I wanted to know slow/fast does the error converge (since our initialization itself was due to odom edges)/

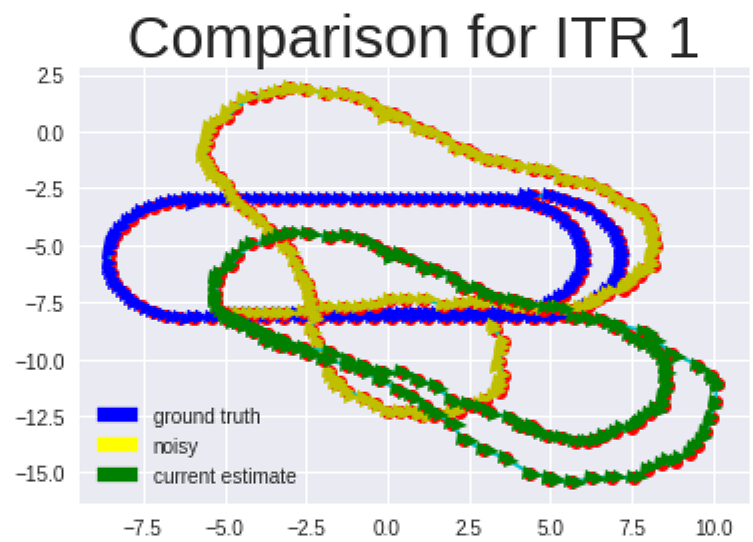
Best Error: 38

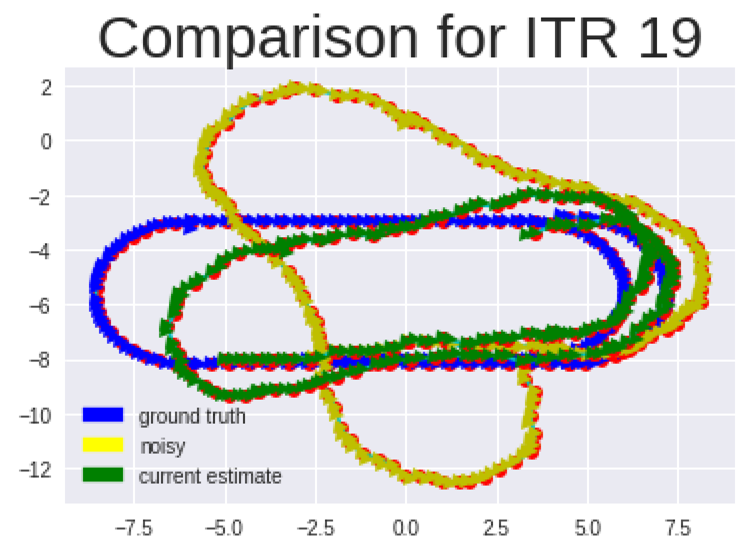
Comments on convergence:

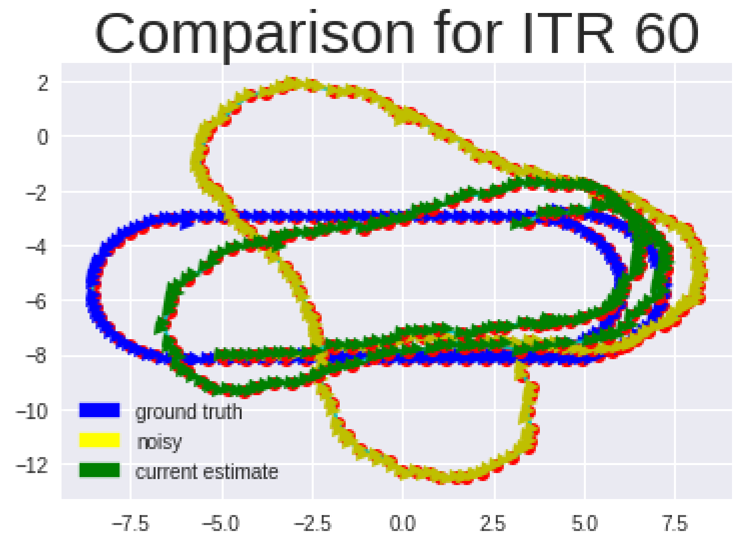
Converges really quickly. This is because the initialization is due to the ODOMETRY EDGES and again, finally, the highest emphasis is placed on odometry edges only.



Comments on how trajectory changed over time:







Picture:

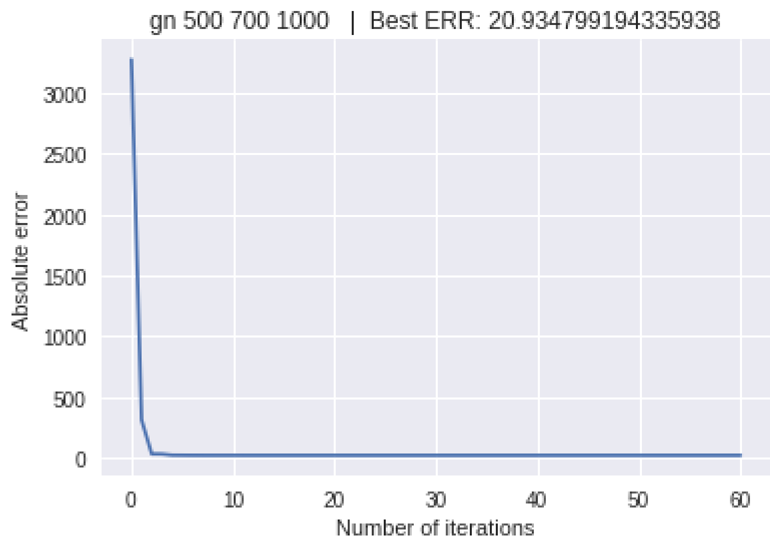
- **Provide table/plot for multiple initial estimate vs convergence.**

Earlier, for almost all weights, we had seen fast convergence.
Also, whatever change we did see was due to loop closure edges mostly (since the initialization itself was due to odometry edges). As a result, seeing how the trajectory reacts when all vertices are initialized with ORIGIN is interesting.

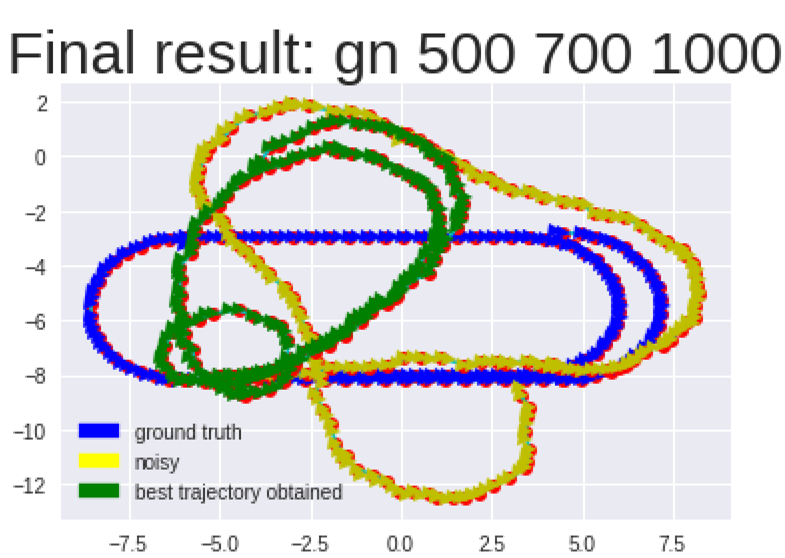
22, 31, 45 Gaussian Newton (Corresponds to 500, 700, 1000) : ORIGIN INITIALIZATION

Best Error: 20

Comments on convergence:
Error converged after iteration 6. almost same as the case with the ODOM initialization.

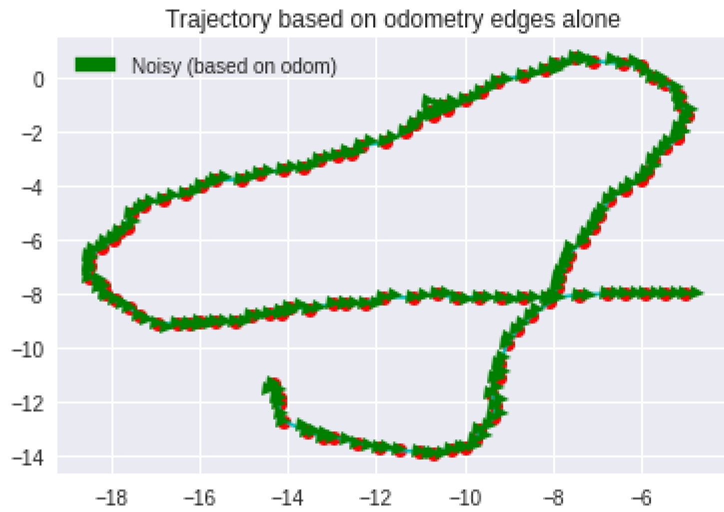


Picture:

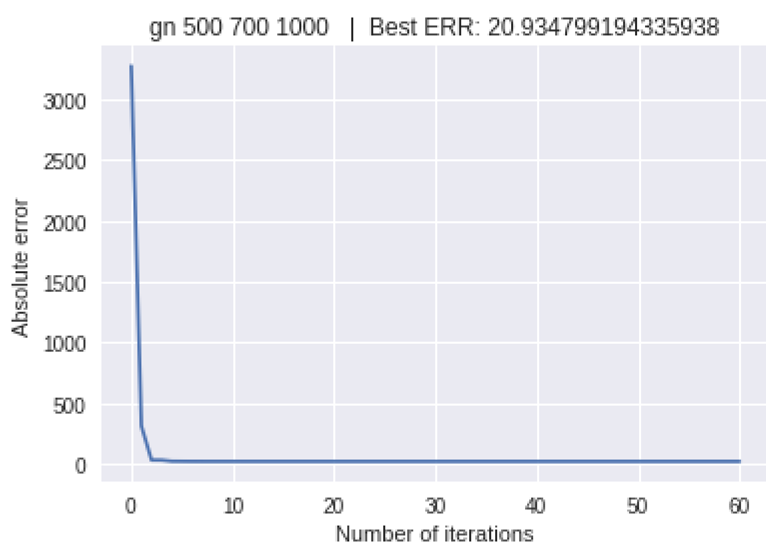
ODOM initial	ORIGIN INITIAL
	

22, 31, 45 Gaussian Newton (Corresponds to 500, 700, 1000) : INITIALIATION EXACTLY REVERSE as conveyed by the odometry edges

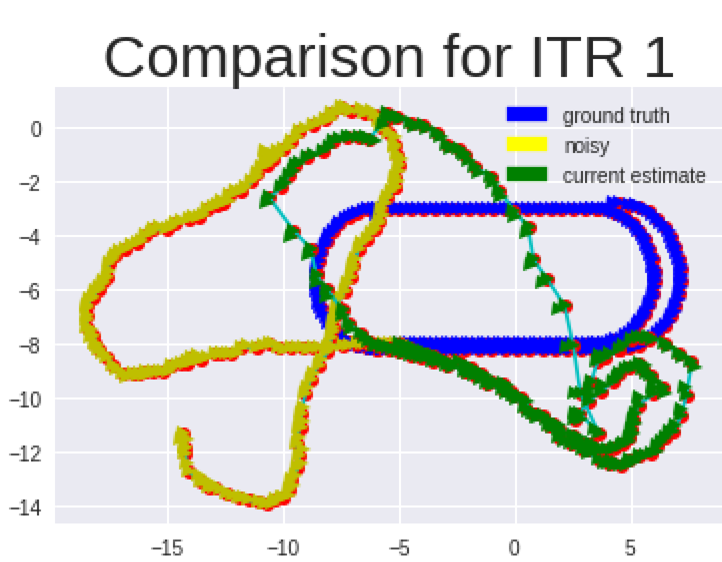
Initial: (Initialization was DONE as per the INVERSE OF odometry edges)

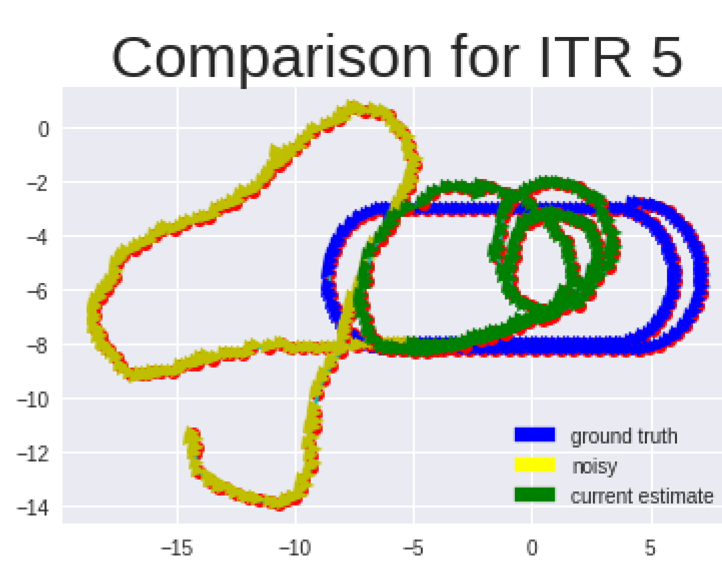


Best Error: 20.9

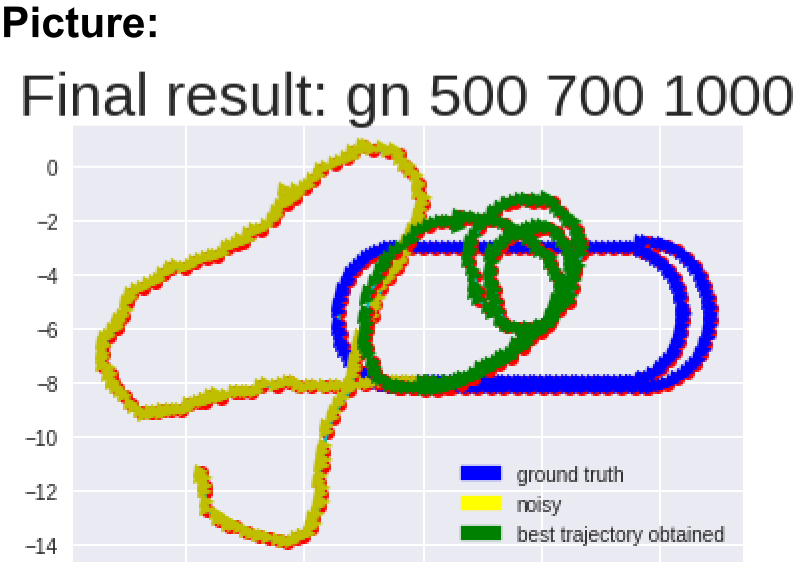


The convergence happened very quickly. In fact, the reverse odometry based initialization was overturned in the very first iteration itself.





Picture:

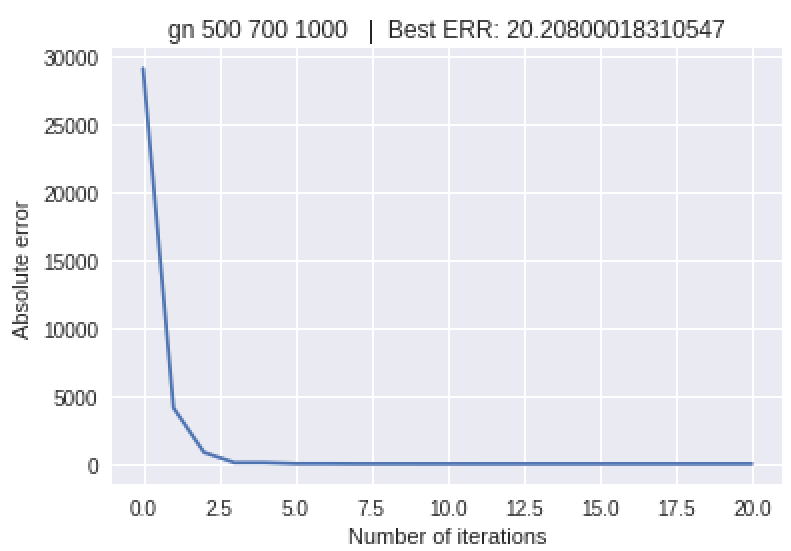
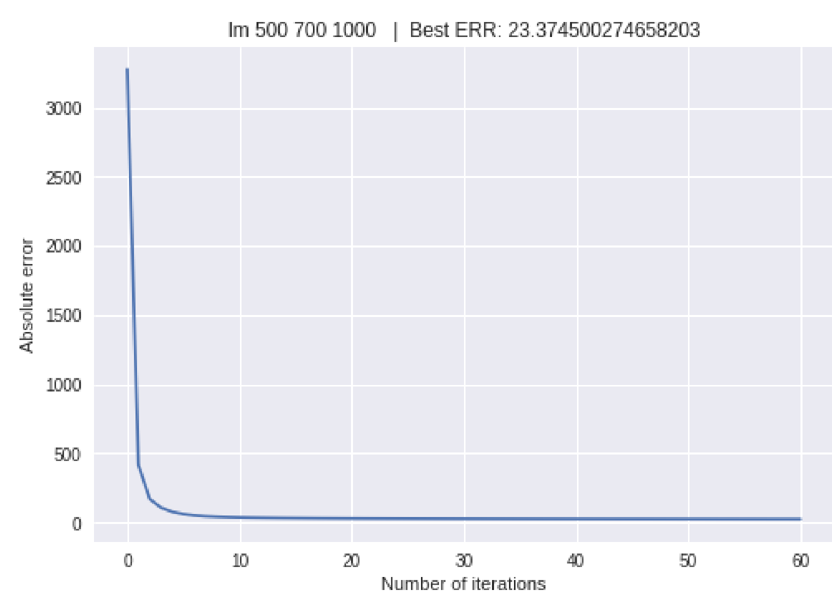


- <u>**Replace Gauss-Newton with Gradient-Descent and Levenberg Marquardt algorithms.**</u>

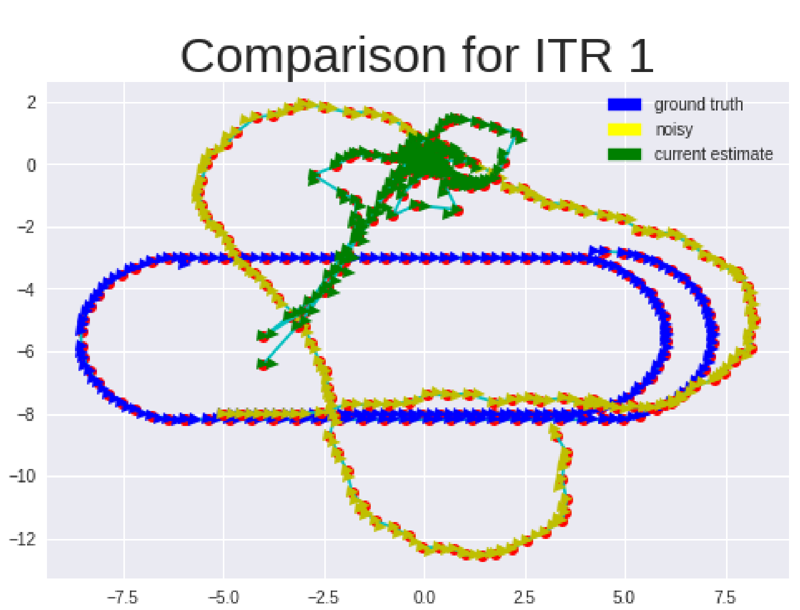
<u>**22, 31, 45 LM (Corresponds to 500, 700, 1000)**</u>

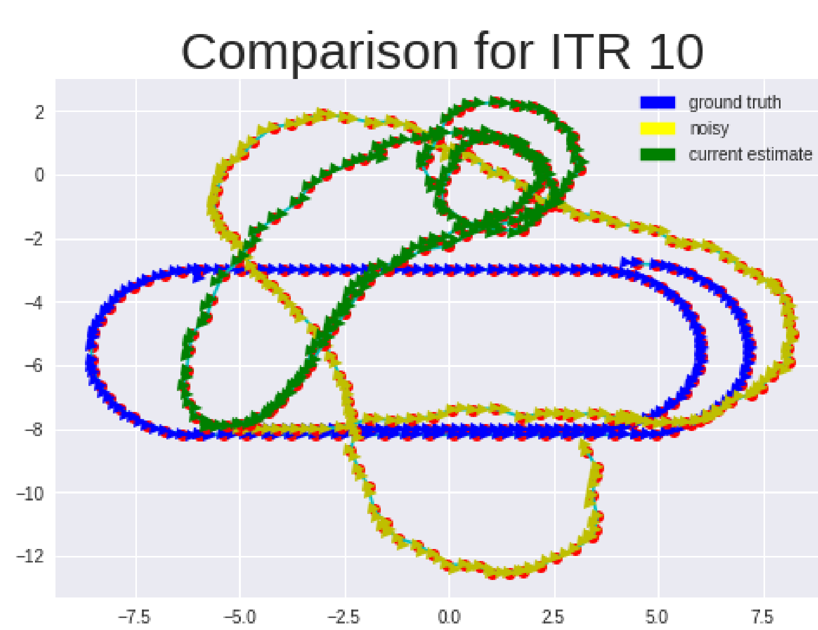
Best Error: 23 (GN was 20)

Comments on convergence:
Error converged after iteration 16.
It is possible that due to the penalization of large delta, the LOCAL MINIMA approached by LM and GN are different.

Via Gaussian Newton	**Via LM**
	

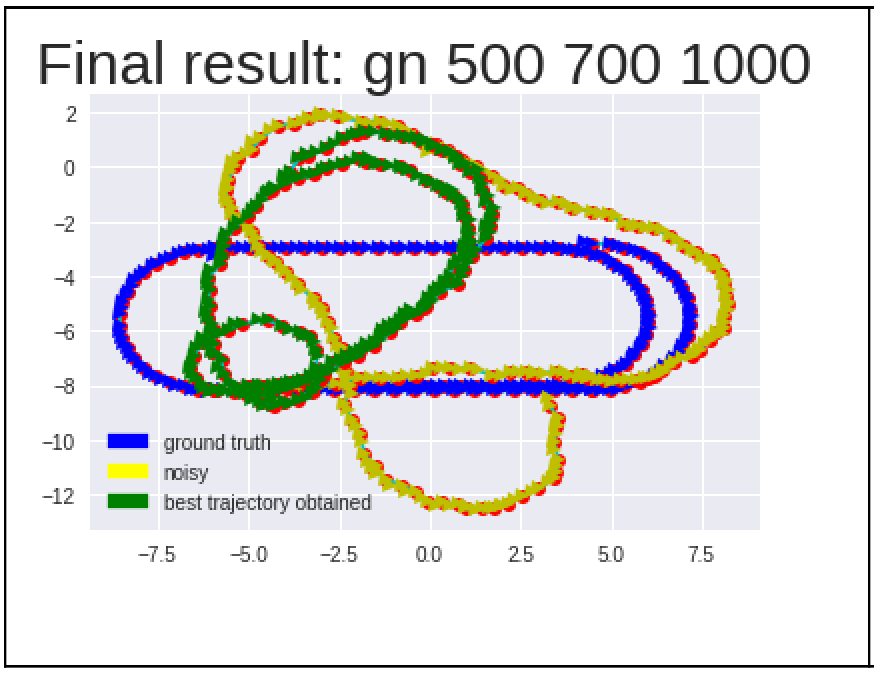
Comments on how trajectory changed over time:
The anchor edge was ignored for the first iteration it seems.

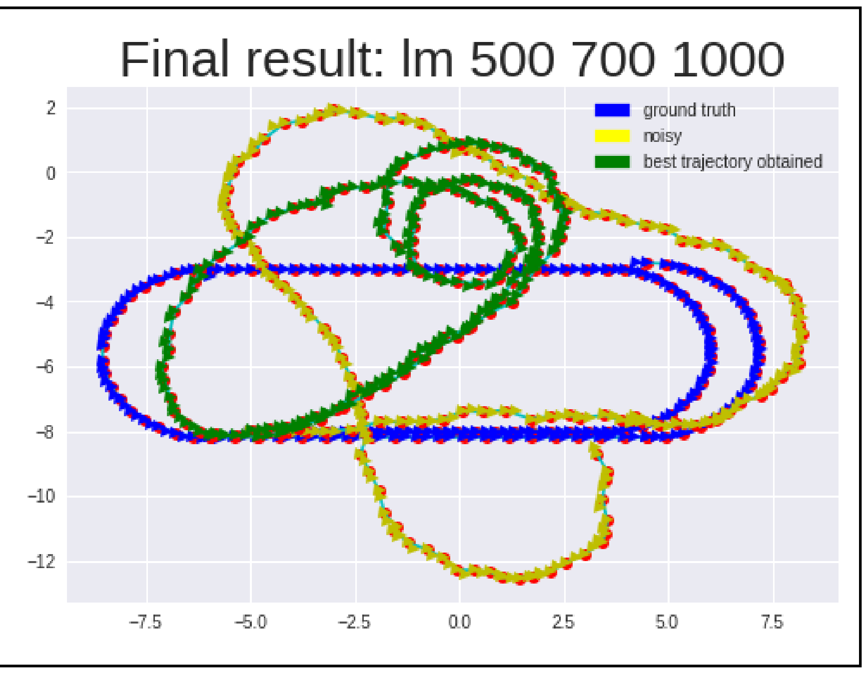




Picture:

Via Gaussian Newton	**Via LM**





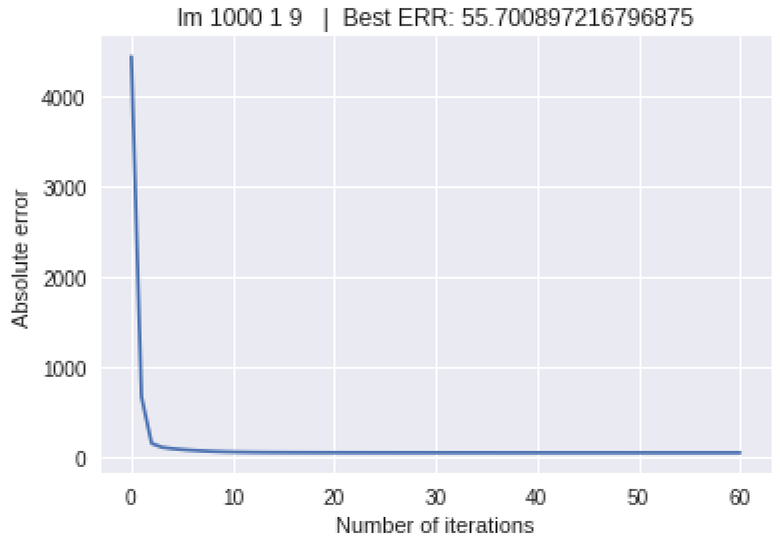
Odom: 99, LOOP: 0.009, Anchor: 1) LM

Reason for choosing:

I wanted to see what happens when we negligible weightage to loop closure edges and high weight to odom edges. I wanted to know slow/fast does the error converge (since our initialization itself was due to odom edges)/

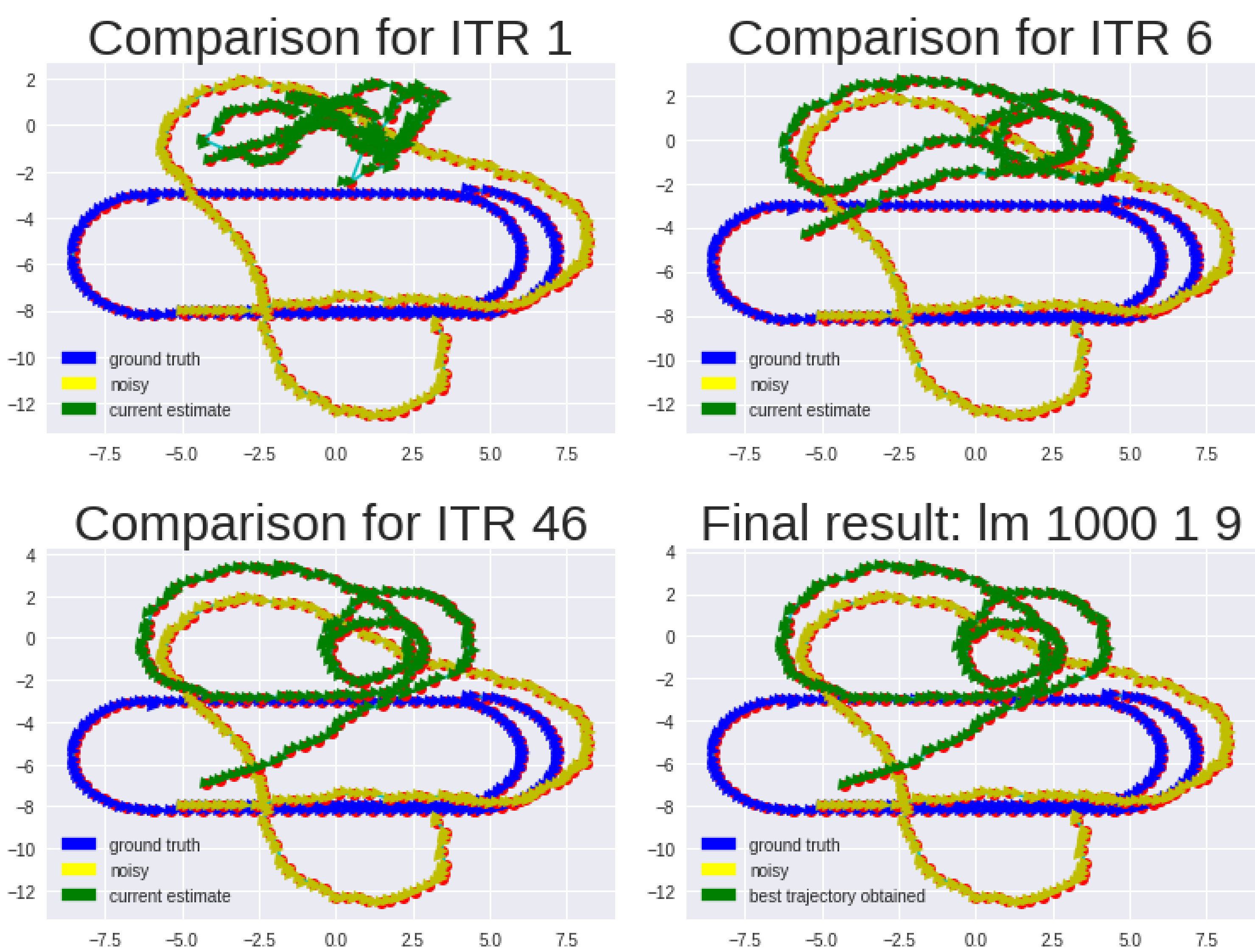
Best Error: 55 (GN led to 38)

Comments on convergence:

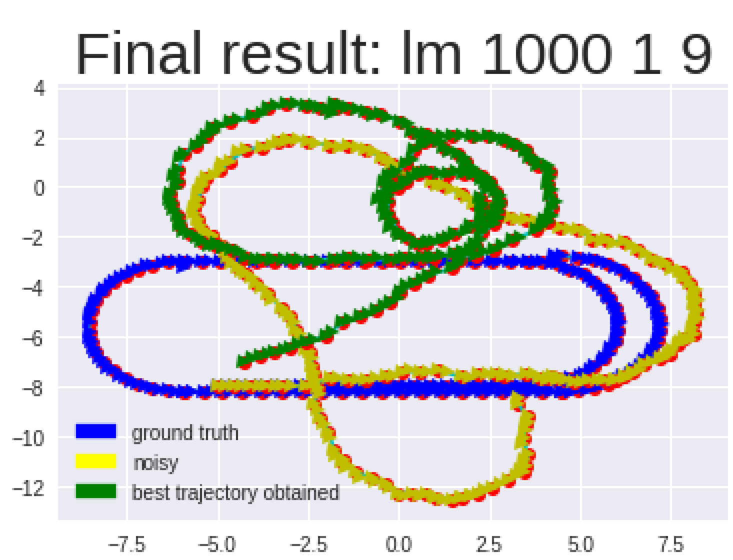


Converges with medium pace but to a pose where the ANCHOR EDGE IS NOT respected. This may be because of the low weight assigned to the anchor edge. As a result, the model would have favored some other constraint for the 0th edge. With time, it started to respect the anchor edge BUT was not able to make immediate changes due to the penalization for a large DX.

Comments on how trajectory changed over time:



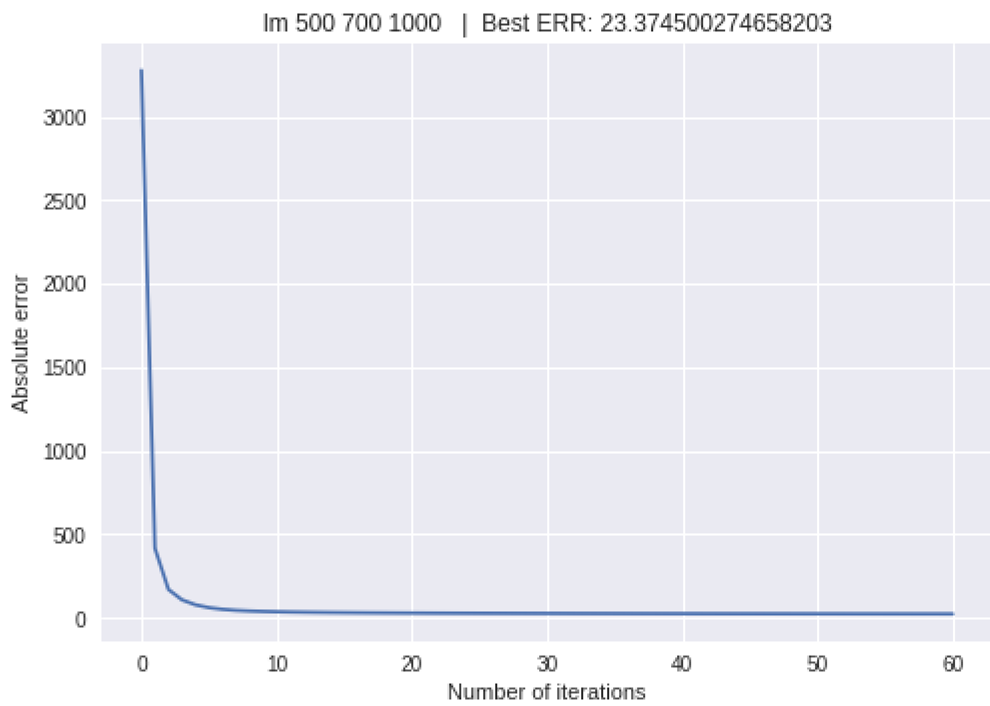
Picture:

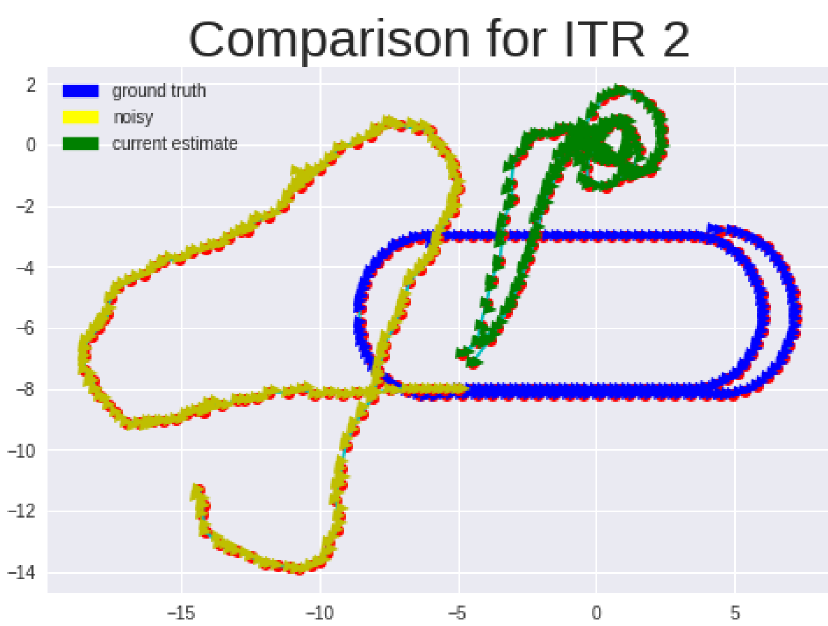
Via GN	Via LM
	

22, 31, 45 LM (Corresponds to 500, 700, 1000) : ORIGIN INITIALIZATION

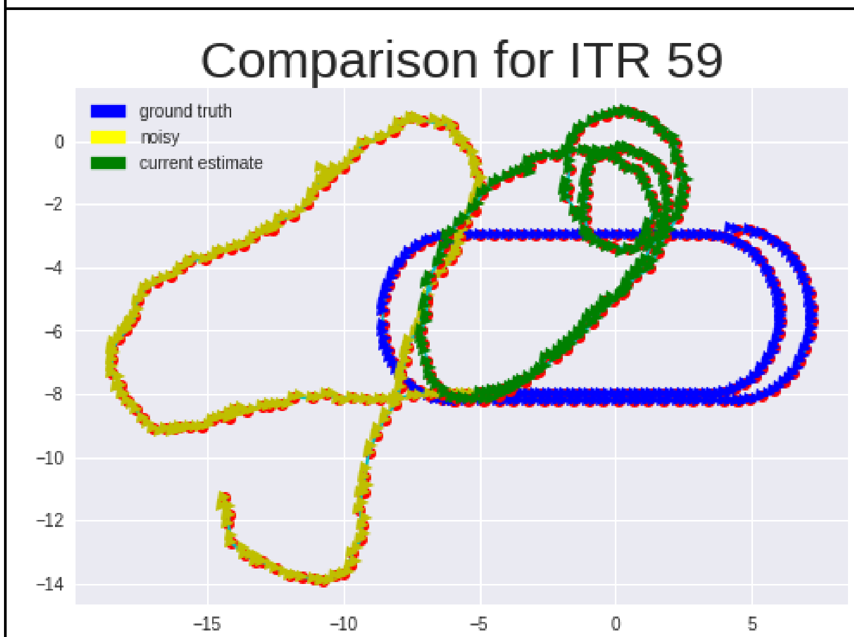
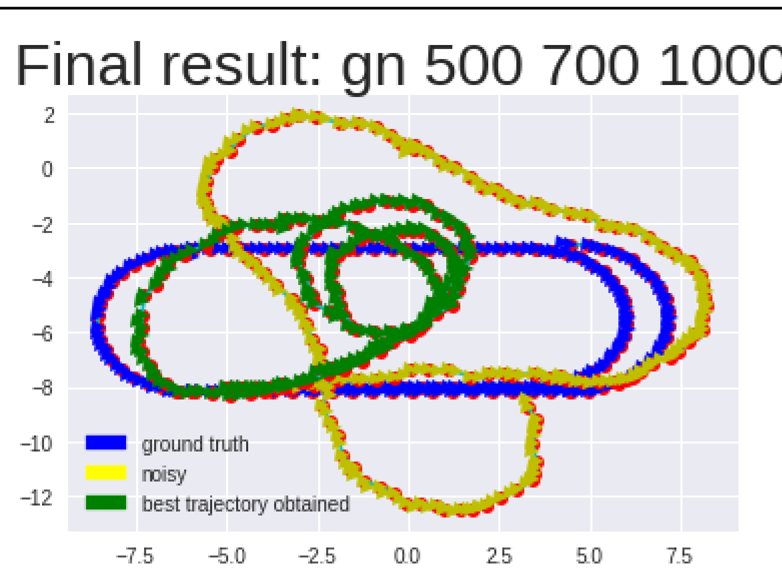
Best Error: 23 (GN had led to 20)

Comments on convergence:
Error converged after iteration 9.





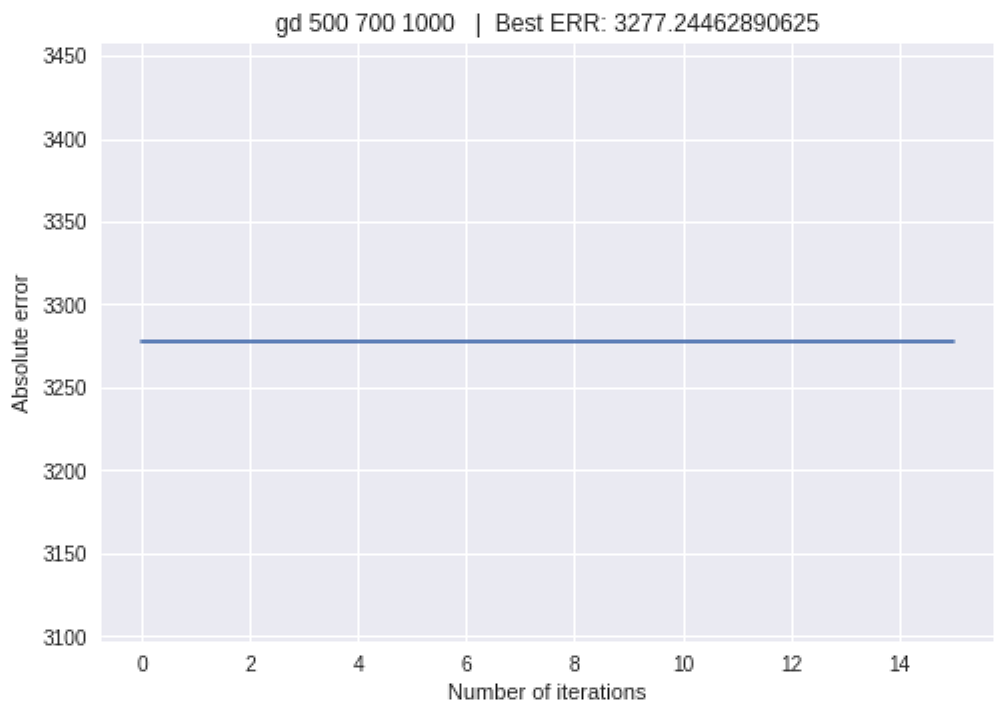
Picture:
We see that the final location reached for LM is same as that of GN (WHICH ARE DIFFERENT FROM GM with odom initialization). .

ORIGIN INITIAL (LM)	ORIGIN INITIAL (GN)
	

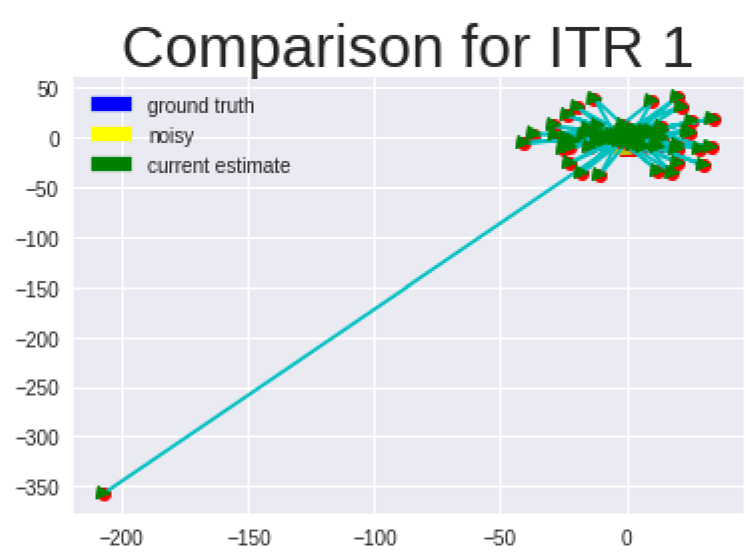
22, 31, 45 Gradient descent (Corresponds to 500, 700, 1000)

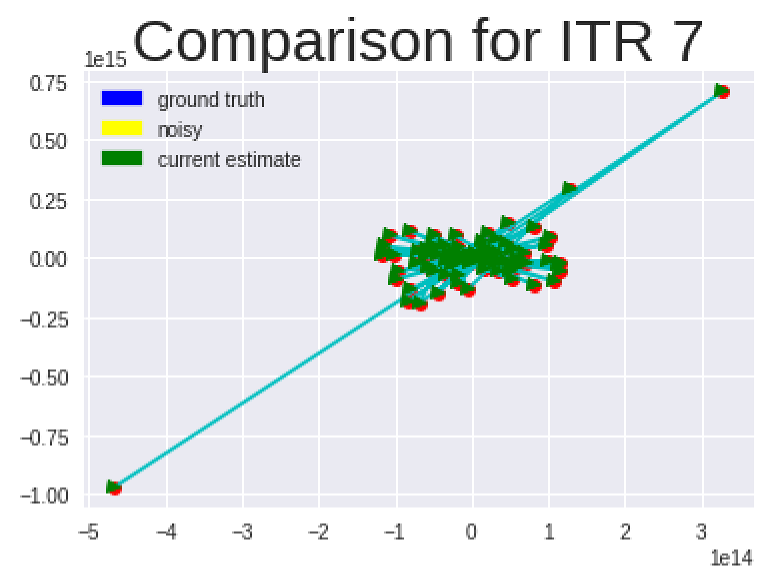
For me, the trajectory obtained via GD was worsening with time and the error started to approach inf. I introduced a learning rate and the same problem persisted.

Comments on convergence:
The y-axis is in order of 1e36.



Comments on how trajectory changed over time:





Picture: