Perception, localization and mapping for mobile robots

First project

• Marco Cella	10578855
• Giacomo Delcaro	10560602
• Alessandro Colombo	10573335

POLITECNICO
MILANO 1863

FILES

- bags:
    - bag1.bag
    - bag2.bag
    - bag3.bag

- cfg:
    - parameters.cfg: file for the dynamic reconfigure

- launch:
    - scout_launch.launch : launch file, launches the velpubsub, odompubsub and residual nodes. It is also possible to change the inital position of the robot.

    - scout_launch_rviz.launch: launch file, just like the other one but it also plays the bag and runs rviz with the right configuration.

    - scout_launch_bag.launch: launch file just for the bag. It can be modified to play bag1, bag2 or bag3.

    - scout_launch2.launch: just like scout_launch but with the initial conditions of the second bag.

    - scout_launch2.launch: just like scout_launch but with the initial conditions of the third bag

POLITECNICO
MILANO 1863

FILES

- msg:
  - OdomInt.msg: contains the custom message to publish odometry and integration method
  - er_array.msg: custom message to publish the residuals between the ground truth and the estimated odometry
- srv:
  - resetOdom.srv: empty, it does not need any input from the user since it simply resets the pose
  - seOdom.srv: defines the types of the requested input from the user to set the pose to x, y, theta
- src:
  - velpubsub.cpp: the node that subscribes to the rpm topics and publishes a twist message containing all of the robot's speeds
  - odompubsub.cpp: the node that subscribes to the topic advertised by velpubsub and publishes the complete odometry and the custom message
  - residuals.cpp: the node publishes the residuals between /scout_odom and our computed odometry /our_odom and also the residuals between /gt_pose (in odom reference frame) and /our_odom

FILES

- config:
    - plot_layout.xml: layout for plotjuggler.

    - rototranslation.m: MATLAB file to compute the “odom” to “world” rototranslation parameters.

    - scout_rviz.rviz: rviz configuration to visualize the computed odometry vs the ground truth

ROS PARAMETERS

x_init: initial x position in the odom frame
y_init: initial y position in the odom frame
theta_init: initial yaw angle wrt the odom frame
CHI : ratio between apparent and real base_line
Inv_RATIO: 1/ratio of the gearbox

TF TREE

- odom
    - base_link
    - world (static transform)

CUSTOM MESSAGES

er_array.msg:

```
std_msgs/Float64 dx
std_msgs/Float64 dy
std_msgs/Float64 dtheta
std_msgs/Float64 cumulateError
```

OdomInt.msg:

```
nav_msgs/Odometry odo
std_msgs/String   int_method
```

HOW TO RUN EVERYTHING

I - catkin_make in the catkin workspace root
II- source catkin_ws/devel/setup.bash
1 - `roslaunch project1 scout_launch.launch`
2 - `rosbag play -l bag1.bag`
NB: for the second step you must cd to the folder where the bag files are.
You can also use `roslaunch project1 scout_launch_bag.launch`
It is now possible to echo the data from the different topics.

If you want to visualize the movement on rviz:
1 - Launch as before
2 - Open rviz with the scout_rviz.rviz configuration
3 - Play the bag

You can also simply launch the scout_launch_rviz.launch file, it will take care of everything.
For the dynamic reconfigure:
while the nodes are running, type in the bash:

```
rosrun dynamic_reconfigure dynparam set /odompubsub intmethod value
```

"Value" can be either 0 for euler integration or 1 for runge - kutta

Reset service: `rosservice call /reset_odom`
Set service: `rosservice call /set_odom x y theta` NB: theta [deg]

APPARENT BASELINE AND TRANSMISSION RATIO

In order to find the parameters to match /scout_odom, we compared the estimated velocities values with the ground truth ones. For the transmission ratio we've used the values of the longitudinal velocity vx, whereas for the apparent baseline we've used the angular velocity wz.
The best values we've found are the following:
- Transmission ratio: 1:38.7
- Apparent baseline: 1.75 * B

To have a quantitative assessment of how we were improving when changing the parameters, we used plotJuggler on the residuals node mentioned above: the cumulateError is a filtered error, which means that it is the average of the last 100 quadratic errors. Each quadratic error is the square root of the sum of the square of deltaX, deltaY, deltaTheta (each multiplied by a particular quadratic weight).
In particular for the errors wrt /scout_odom an error of 1 cm is weighted as 60 degrees, while for the errors wrt /gt_pose 1 cm weights as 120 deg (since the orientation measurement was not smooth and had an offset problem mentioned below).

To find the parameters to match /gt_pose we also had to find the rototranslation from world to odom and find an orientation offset of /gt_pose with respect to its trajectory (see below). Since we did not have any velocities we used the following heuristic function: we started from the previous values and decreased the transmission ratio to match /gt_pose velocity, and then we decreased the apparent baseline so that the curves had a smaller radius (greater curvature)
We obtained the following results:
- Transmission ratio: 1:40.0
- Apparent baseline: 1.64 * B

"ODOM" TO "WORLD" ROTOTRANSLATION

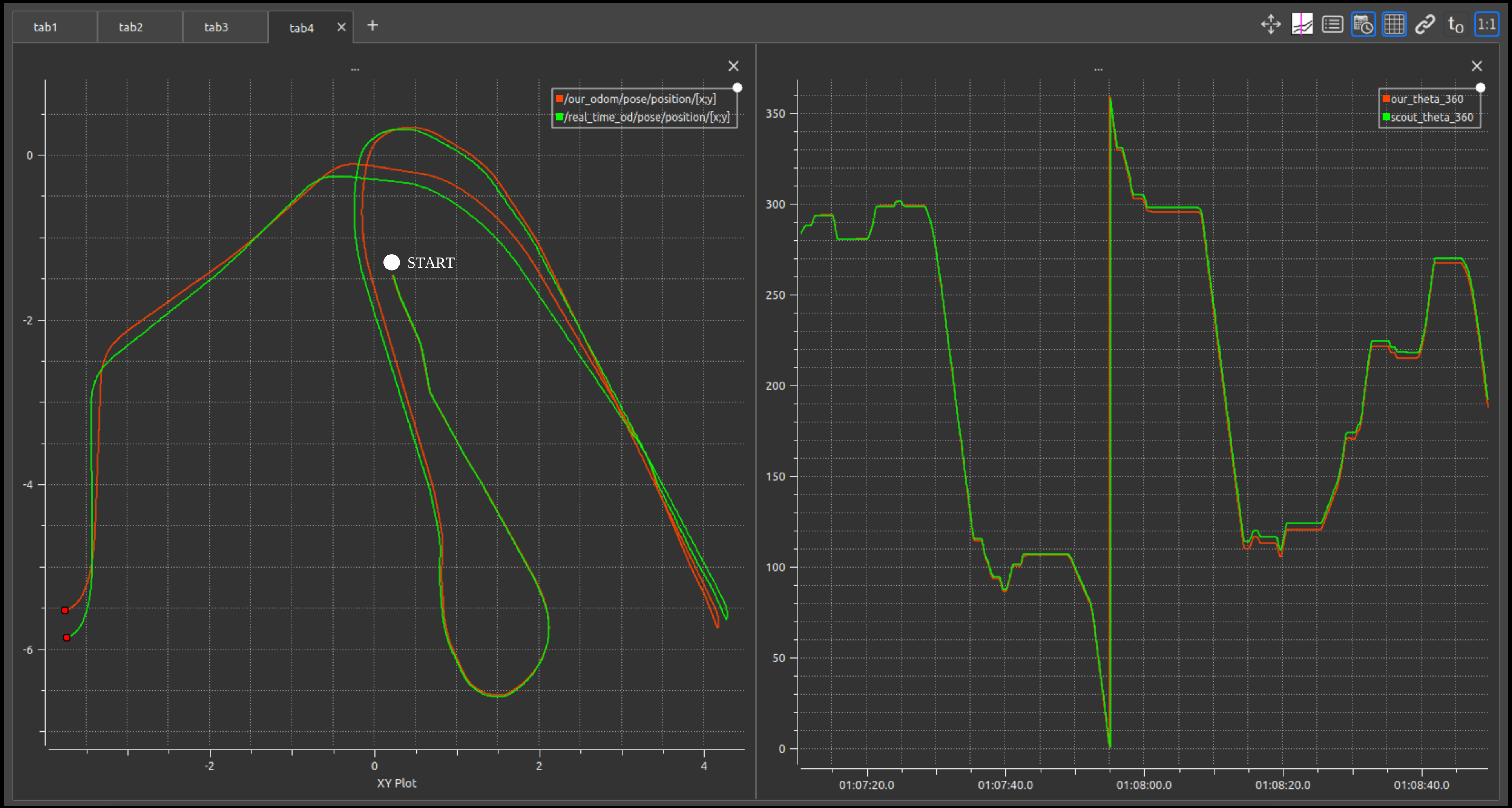
To compute this rototranslation we copied 2 messages with rostopic echo from /scout_odom and /gt_pose.
/gt_pose is not accurate in the first seconds, so we waited for some seconds (about 16s) and as soon as we saw a coherent /gt_pose we copied its pose and the corresponding /scout_odom pose.
We assume that at the beginning /scout_odom is correct, so we can use its pose as reference to find the rototranslation from "odom" (frame_id of /scout_odom) and "world" (frame id of /gt_pose). This has been computed in MATLAB (in the script rototranslation.m).
Doing so we noticed that there was an inconsistency between the /gt_pose theta orientation and the direction of motion of /gt_pose, as if the orientation vector were not indicating the motion direction (so it was not tangent to the trajectory). As a matter of fact, the orientation was very accurate but the trajectory seemed to have a different rotation.
We made the hypothesis that the orientation field in /gt_pose could have an offset wrt the motion of the robot.
We tuned CHI, Inv_RATIO, the rotation of the trajectory (so the rotation from "world" to "odom") and the constant offset of the orientation of /gt_pose.
We noticed that this offset was very consistent (in the first 60 seconds of simulation it is almost constant), and was about 13.5 degrees.
Therefore we fixed the reference system accordingly (and we removed this offset in the residuals node when computing the difference between /gt_pose orientation and our computed orientation).
In the end the world reference frame has a yaw rotation of about 67 deg with respect to odom frame, and
/gt_pose has an orientation offset of about 13.5 degrees wrt its trajectory.

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



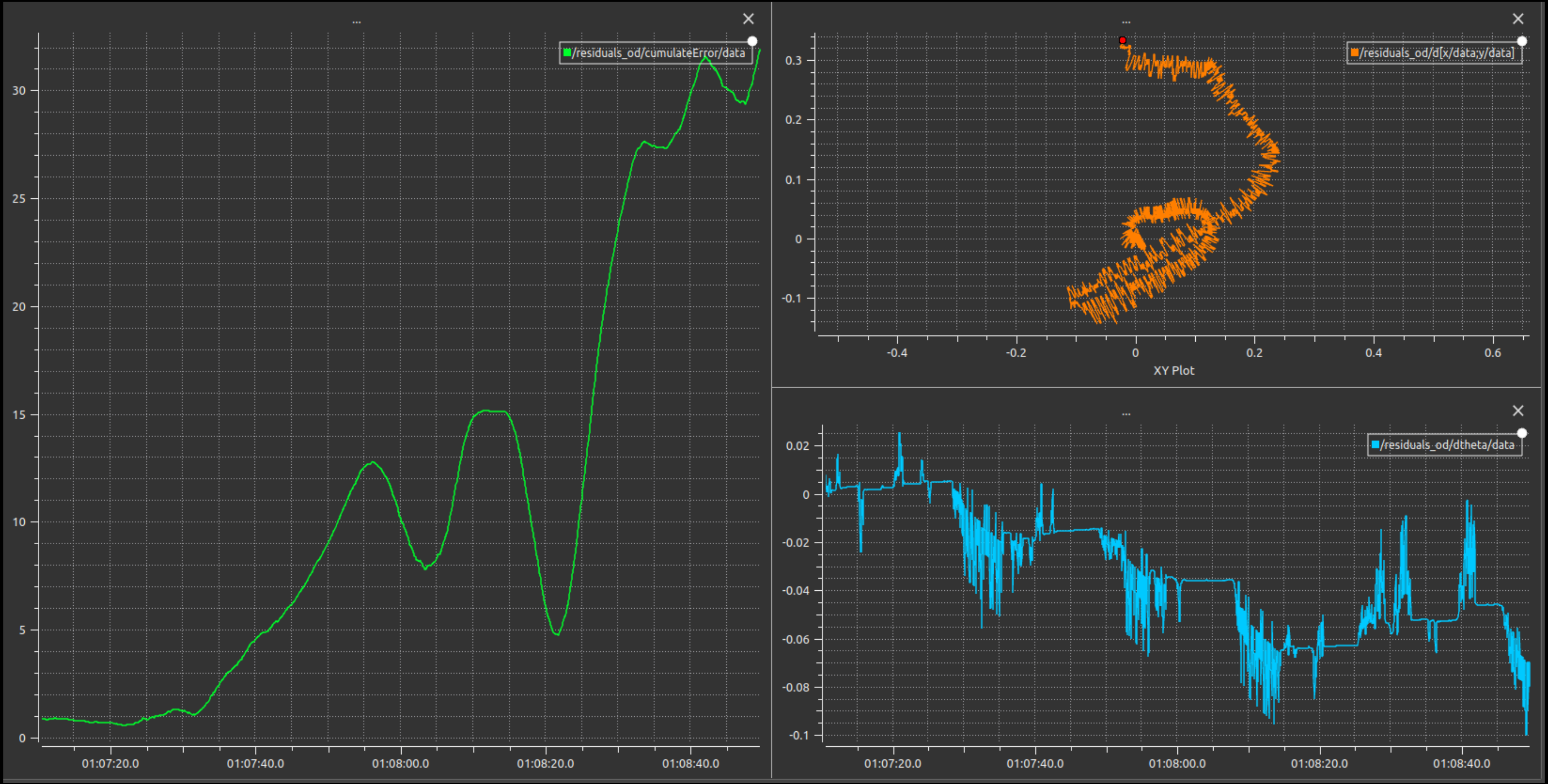
CHI: 1.76
RATIO: 1:38.3

Estimated trajectory compared to the ground truth from scout_odom using not optimal values for the transmission ratio and for the apparent baseline. Nevertheless, the tracking is acceptable.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



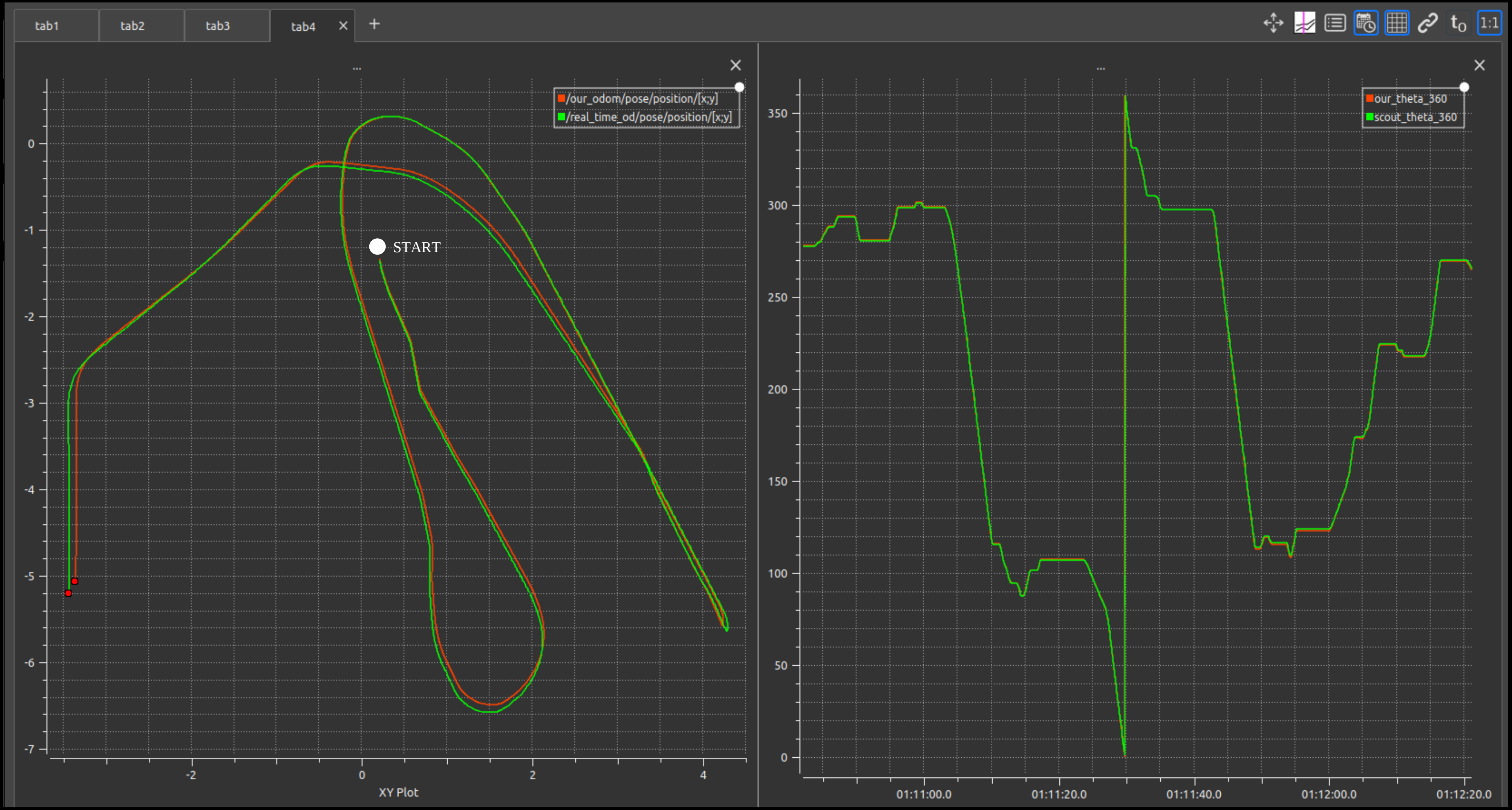
CHI: 1.76
RATIO: 1:38.3

Cost function of the computation with not optimal parameters using scout_odom as ground truth.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



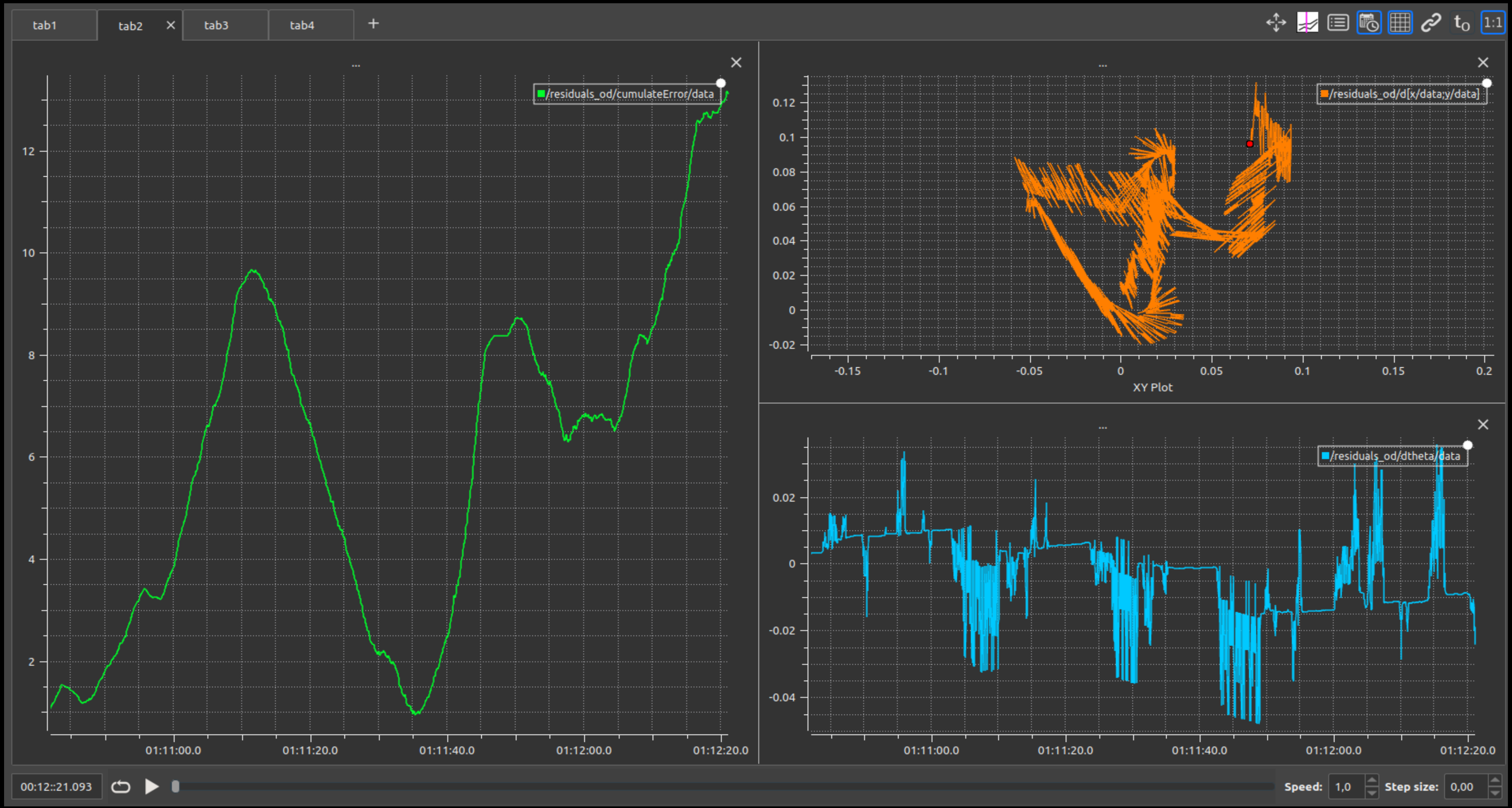
CHI: 1.75
RATIO: 1:38.7

Estimated trajectory compared to the ground truth from scout_odom using the optimal values for the transmission ratio and for the apparent baseline. Now the tracking is nearly perfect.

POLITECNICO
MILANO 1863

RESULTS

Simulation time = 1’30”



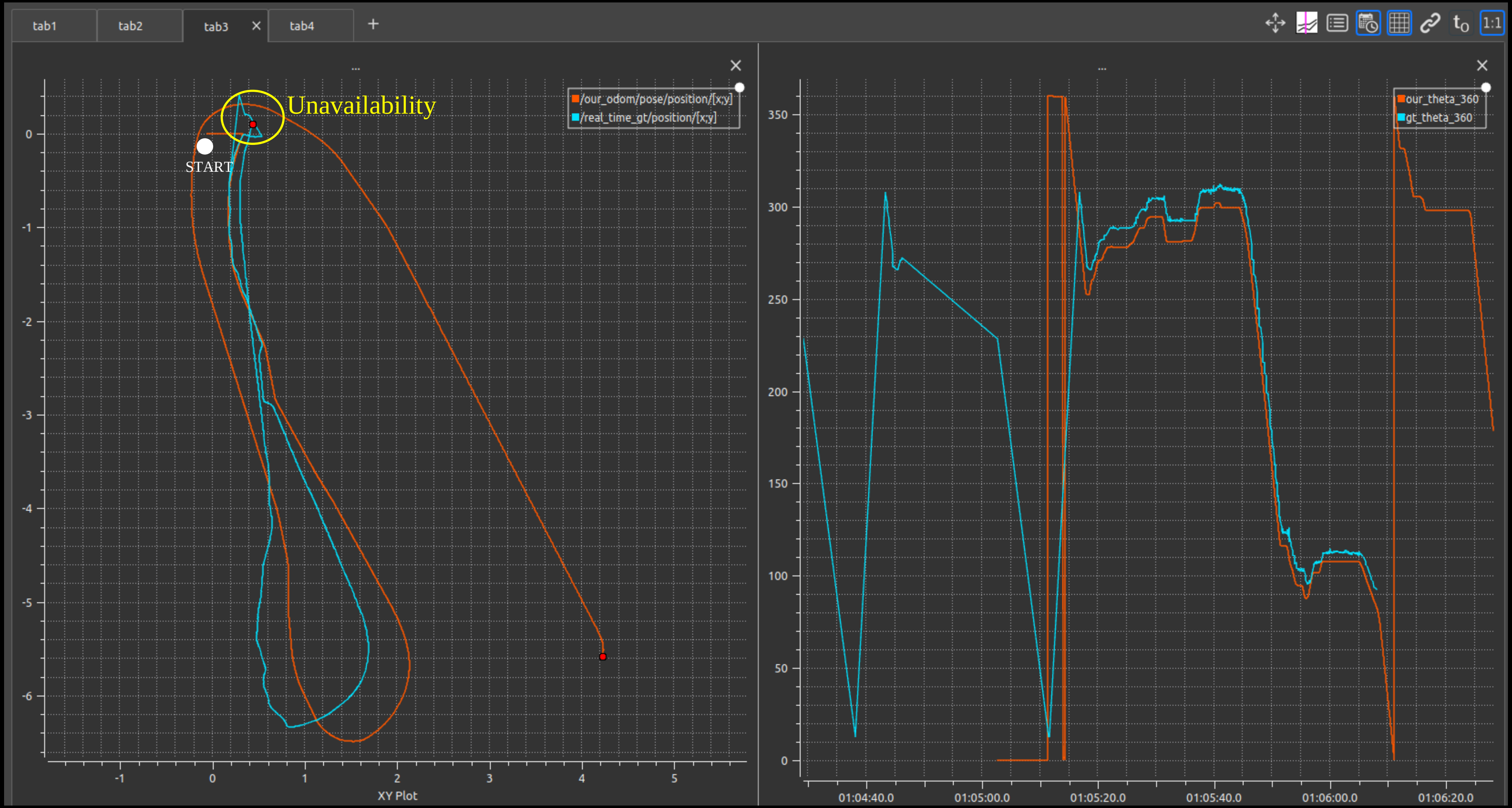
CHI: 1.75
RATIO: 1:38.7

Cost function of the computation with optimal parameters using scout_odom as ground truth.
Note that it’s lower than the previous one.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



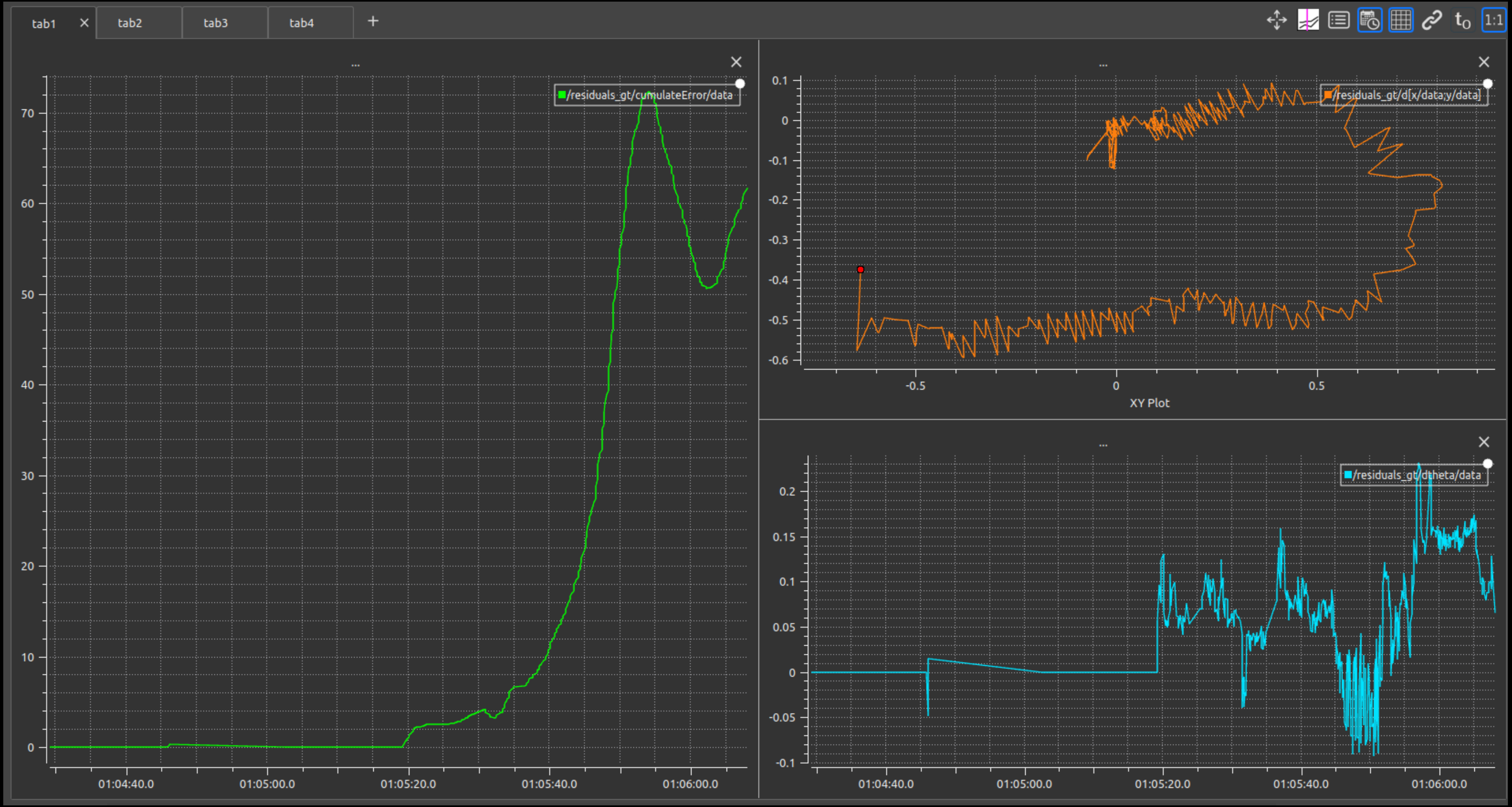
CHI: 1.75
RATIO: 1:38.7

Estimated trajectory using the optimal parameters compared to the ground truth from gt_pose.
Note that, given the unavailability of the OptiTrack system from the higlighted point, the calibration of the parameters using gt_pose has been done using only the first ~ 40” of simulation.

POLITECNICO
MILANO 1863

RESULTS

Simulation time = 1’30”

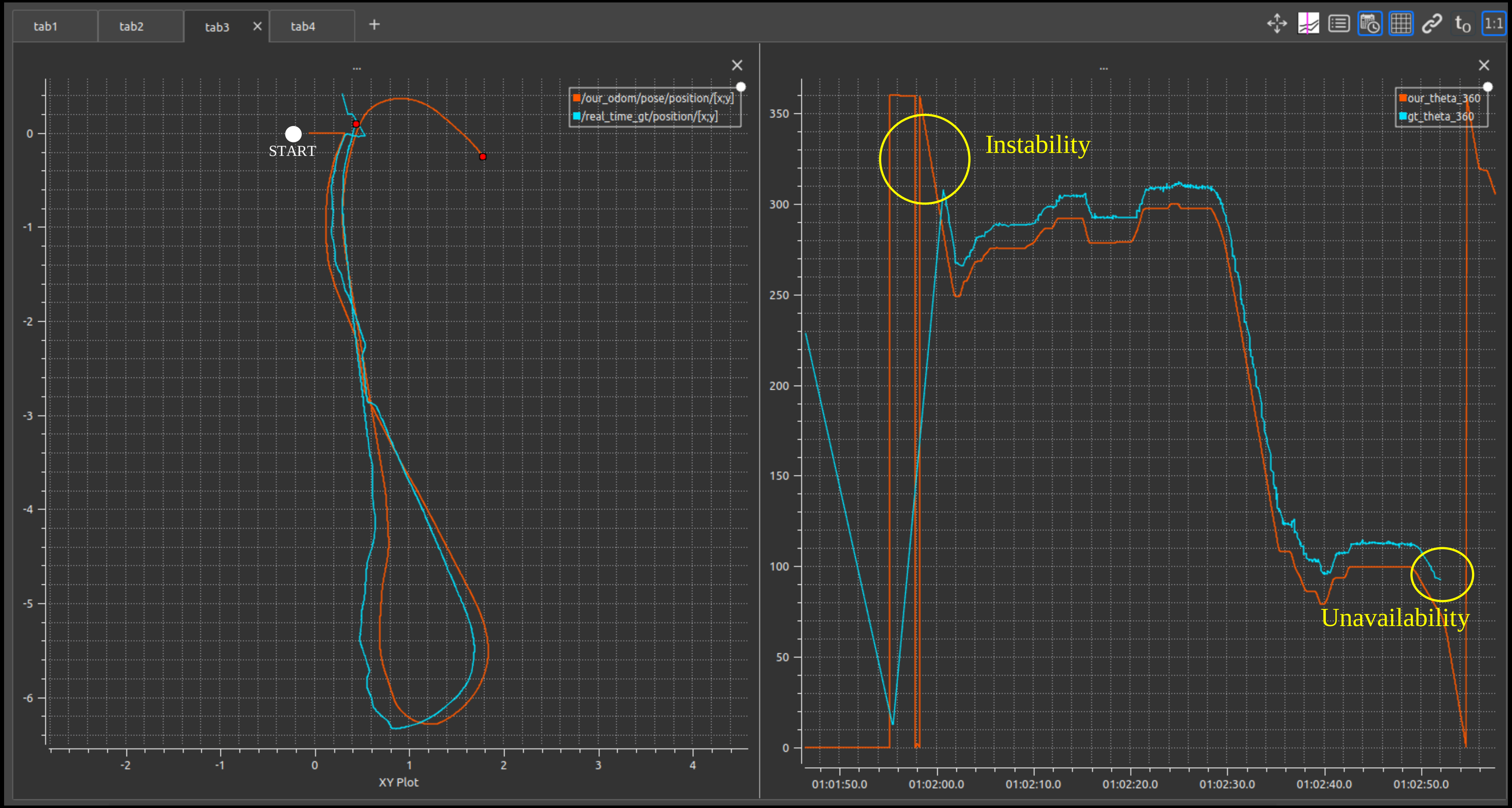


CHI: 1.75
RATIO: 1:38.7

Cost function of the computation with the optimal parameters found with scout_odom and using gt_pose as ground truth.

RESULTS

Simulation time = 1’30”



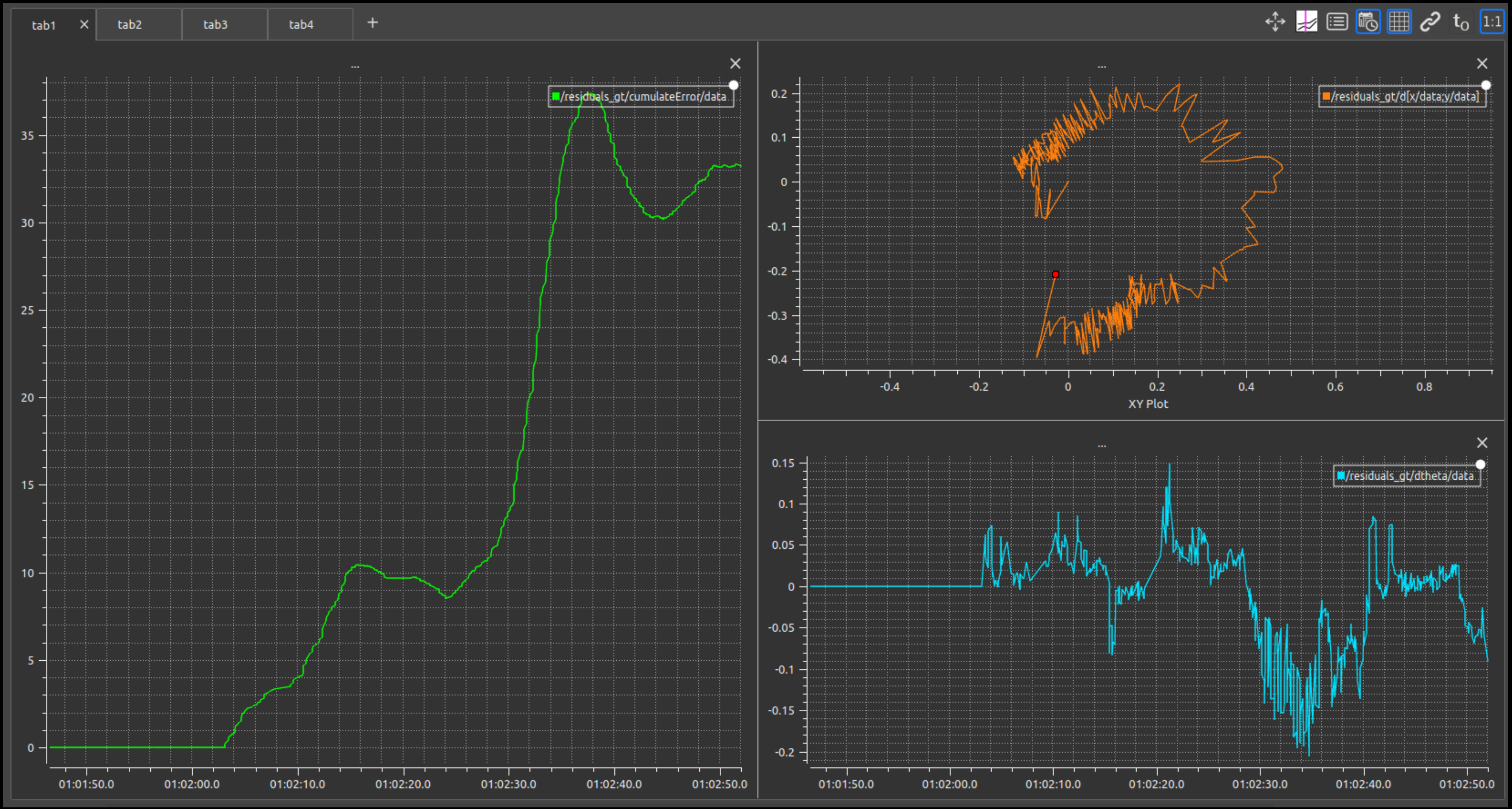
CHI: 1.64
RATIO: 1:40

Estimated trajectory using the new optimal parameters compared to the ground truth from gt_pose.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



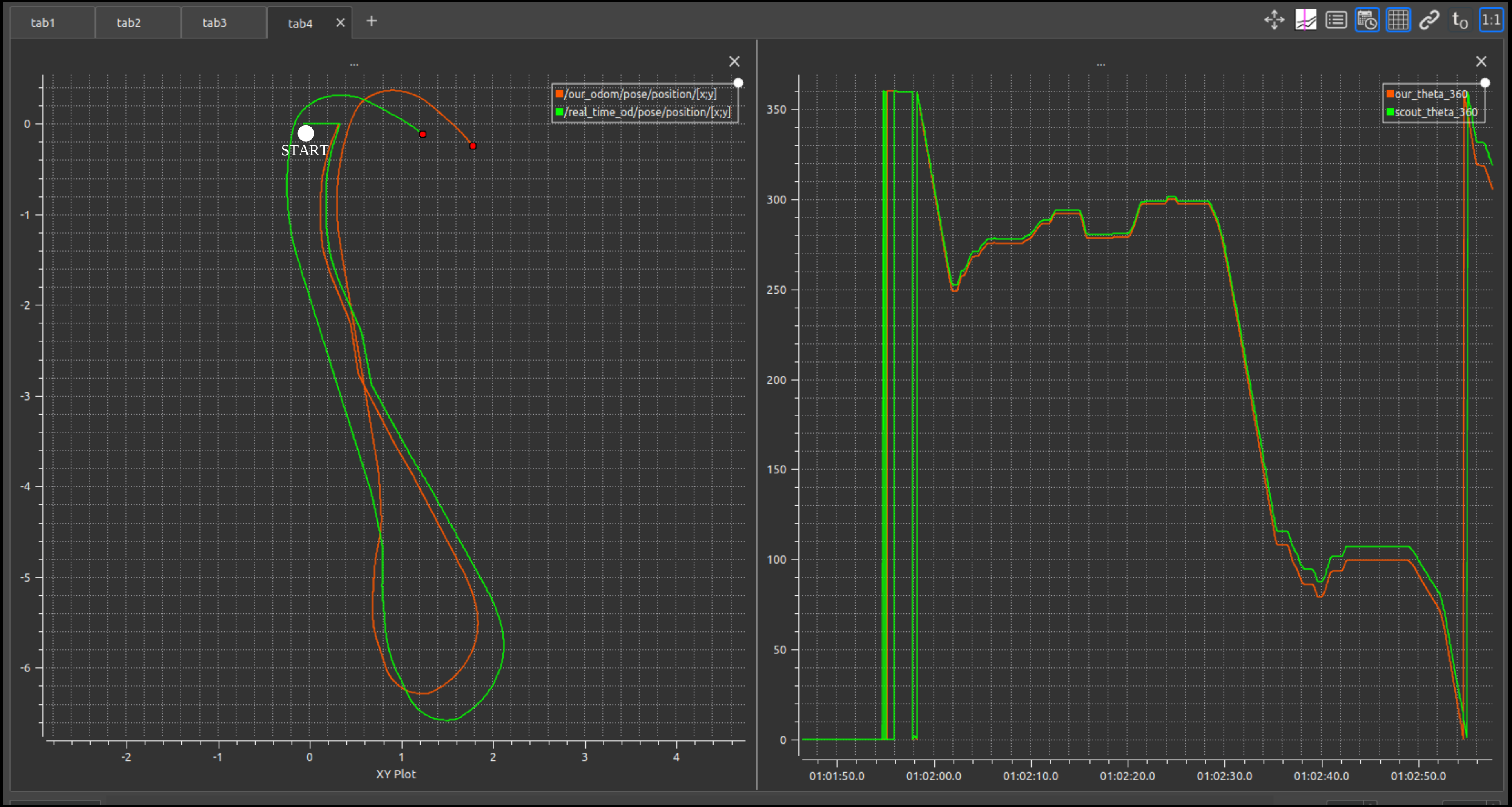
CHI: 1.64
RATIO: 1:40

Cost function of the computation with the new optimal parameters and using gt_pose as ground truth.
Note that it’s lower than the previous one.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”



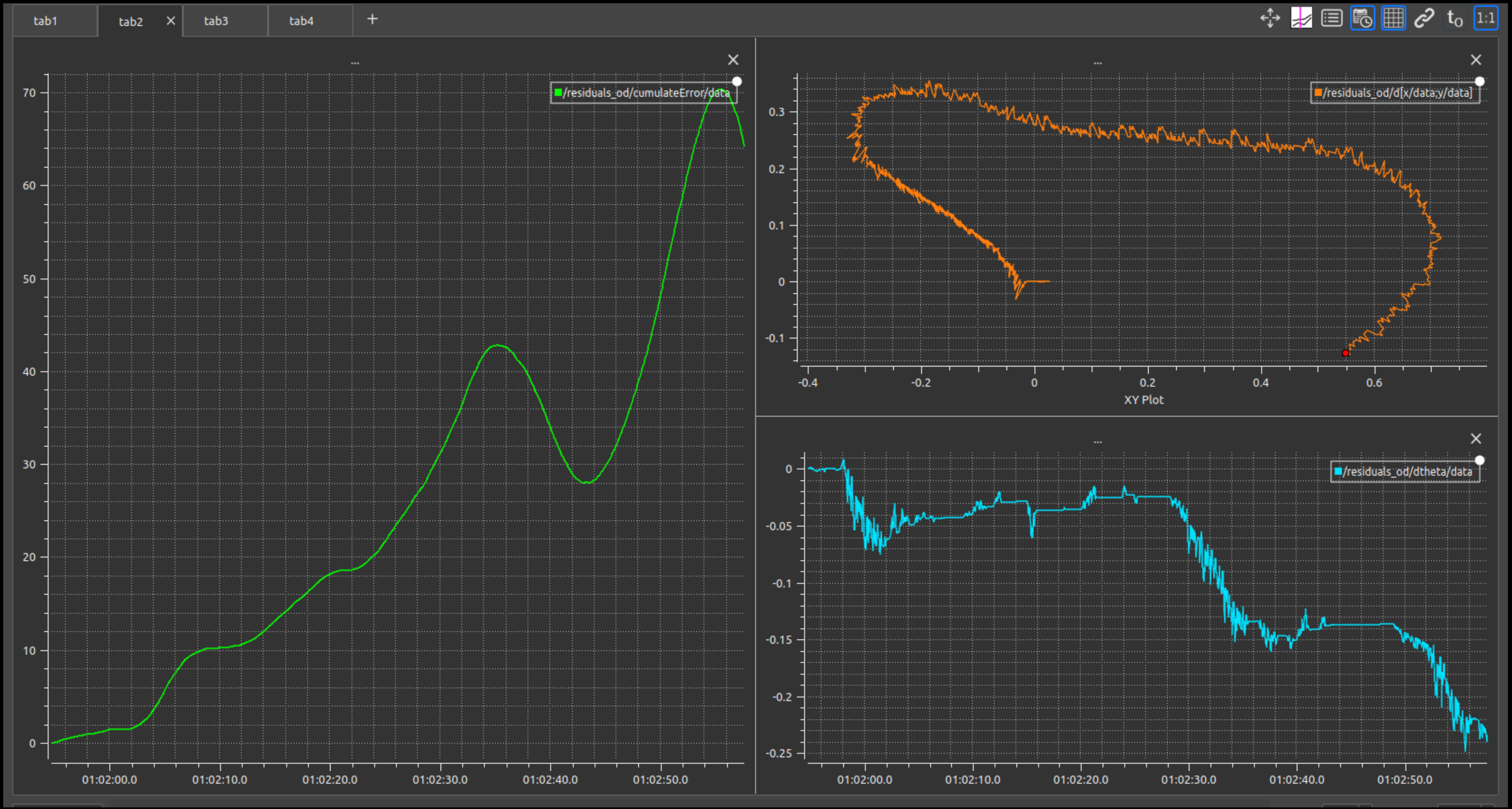
CHI: 1.64
RATIO: 1:40

Estimated trajectory using the optimal parameters found with gt_pose compared to the ground truth from scout_odom.
The result is still acceptable but worse than the one given by the optimal parameters evaluated using scout_odom.
In conclusion, we will use the first computed values both for the apparent baseline and for the transmission ratio.

POLITECNICO MILANO 1863

RESULTS

Simulation time = 1’30”

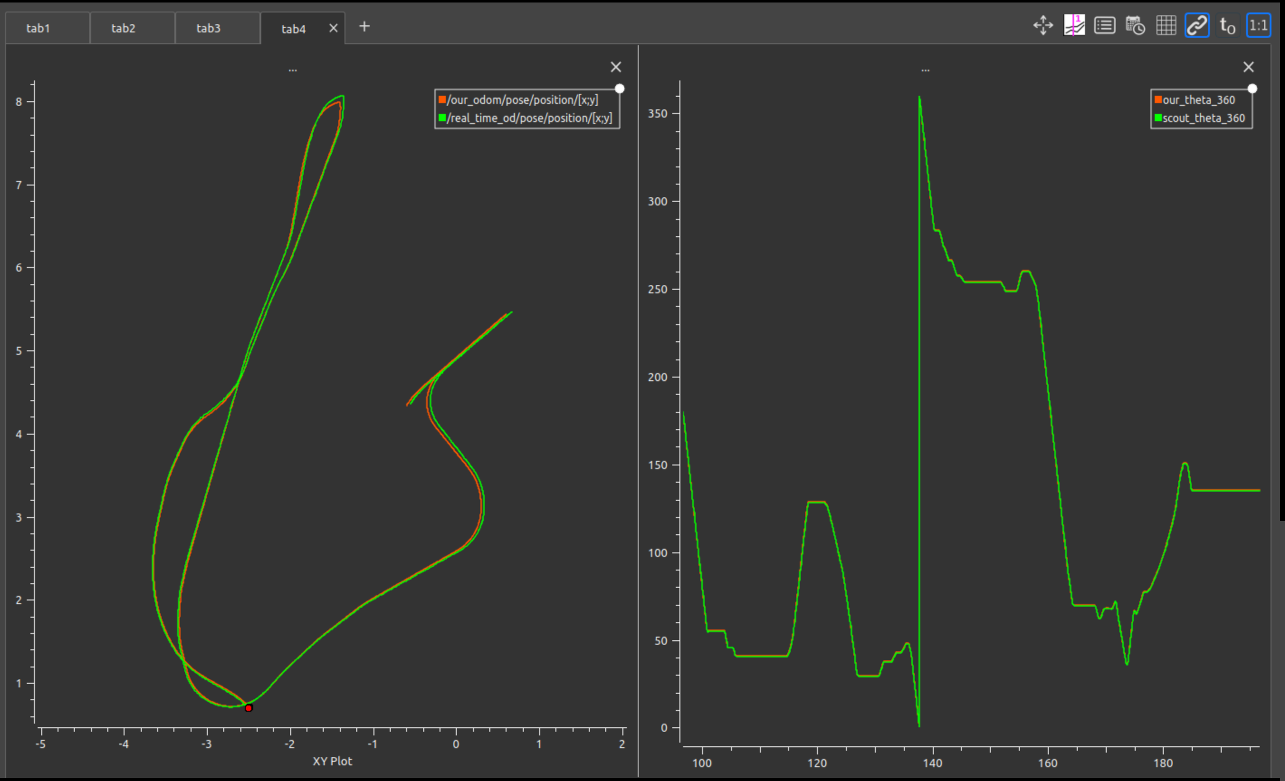

tab1
tab2
tab3
tab4
/residuals_od/cumulateError/data
/residuals_od/d[x/data;y/data]
XY Plot
/residuals_od/dtheta/data
01:02:00.0
01:02:10.0
01:02:20.0
01:02:30.0
01:02:40.0
01:02:50.0

CHI: 1.64
RATIO: 1:40

Cost function of the computation with the gt_pose optimal parameters and using scout_odom as ground truth. Comparing it with the second cost function – the one computed using the scout_odom optimal parameters and using scout_odom as ground truth – it’s clear that the first values of the parameters perform better.

VALIDATION – BAG 2

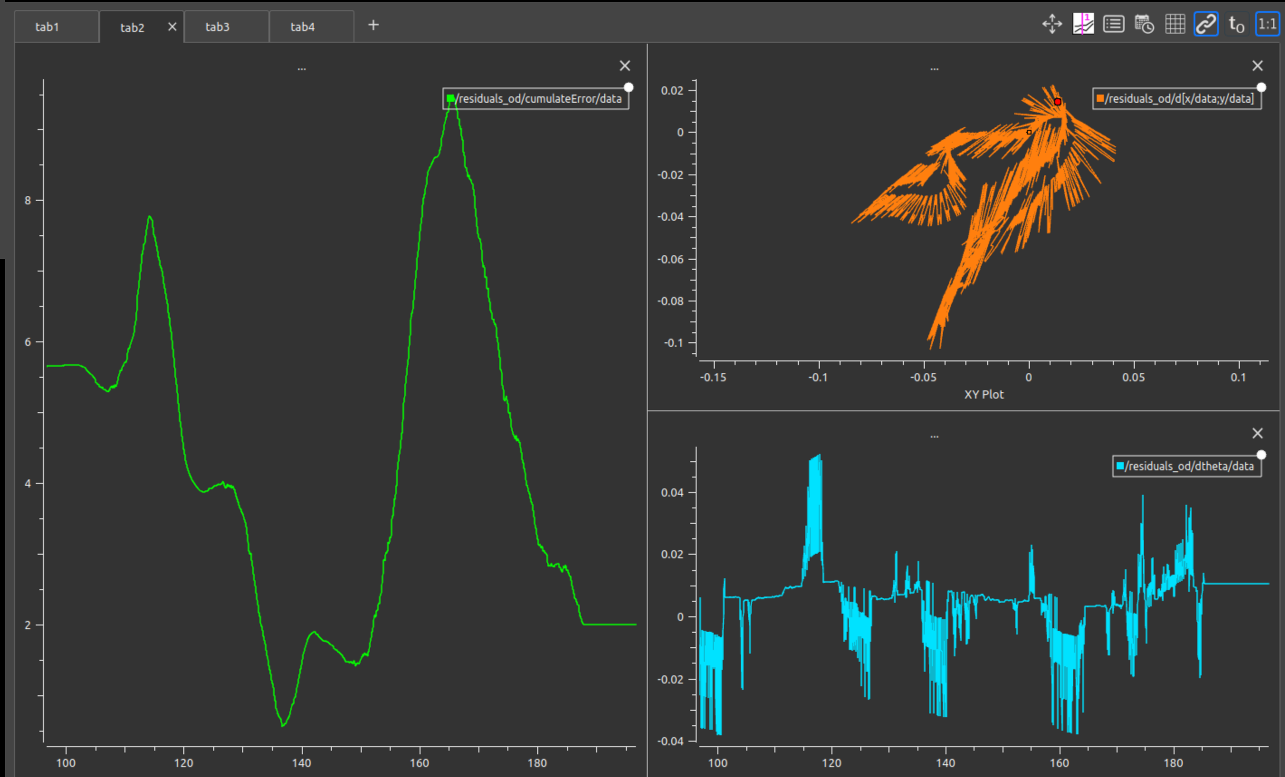
POLITECNICO MILANO 1863



tab1
tab2
tab3
tab4
🟥/our_odom/pose/position/[x;y]
🟩/real_time_od/pose/position/[x;y]
XY Plot
🟥our_theta_360
🟩scout_theta_360

Odometry

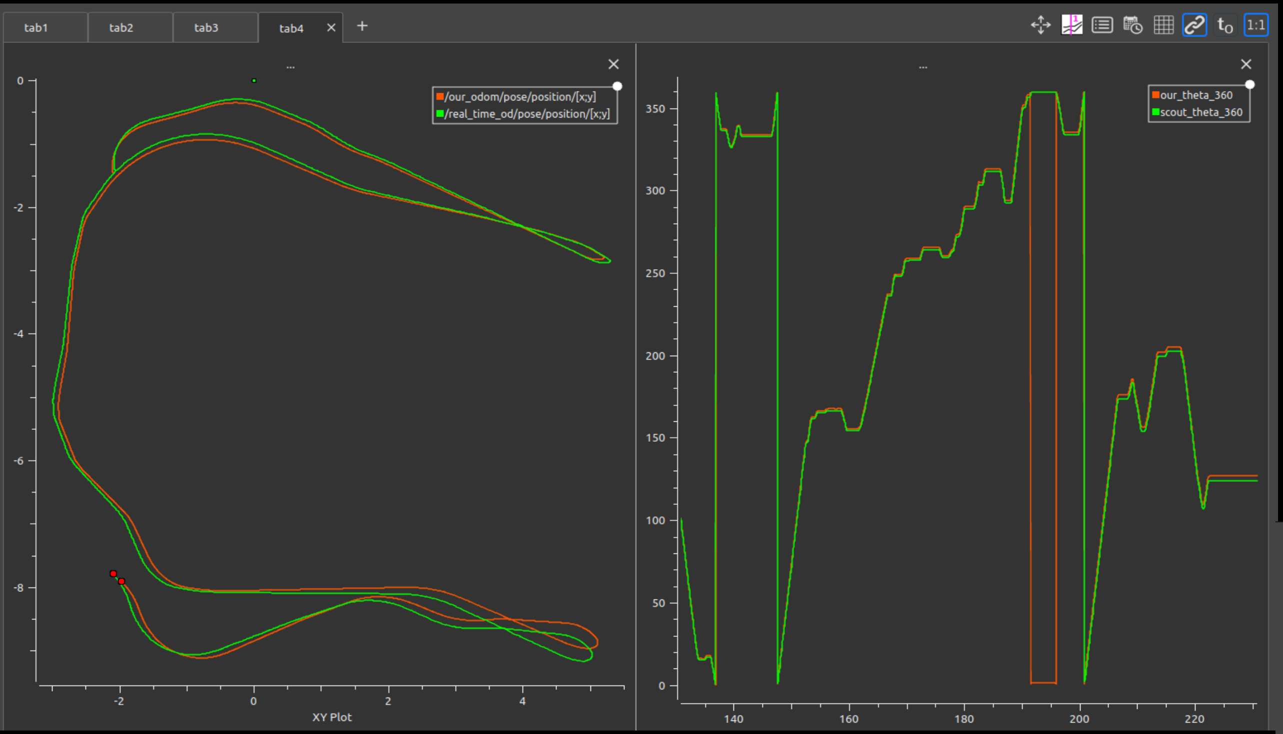
Cost function



tab1
tab2
tab3
tab4
🟩/residuals_od/cumulateError/data
🟧/residuals_od/d[x/data;y/data]
XY Plot
🟦/residuals_od/dtheta/data

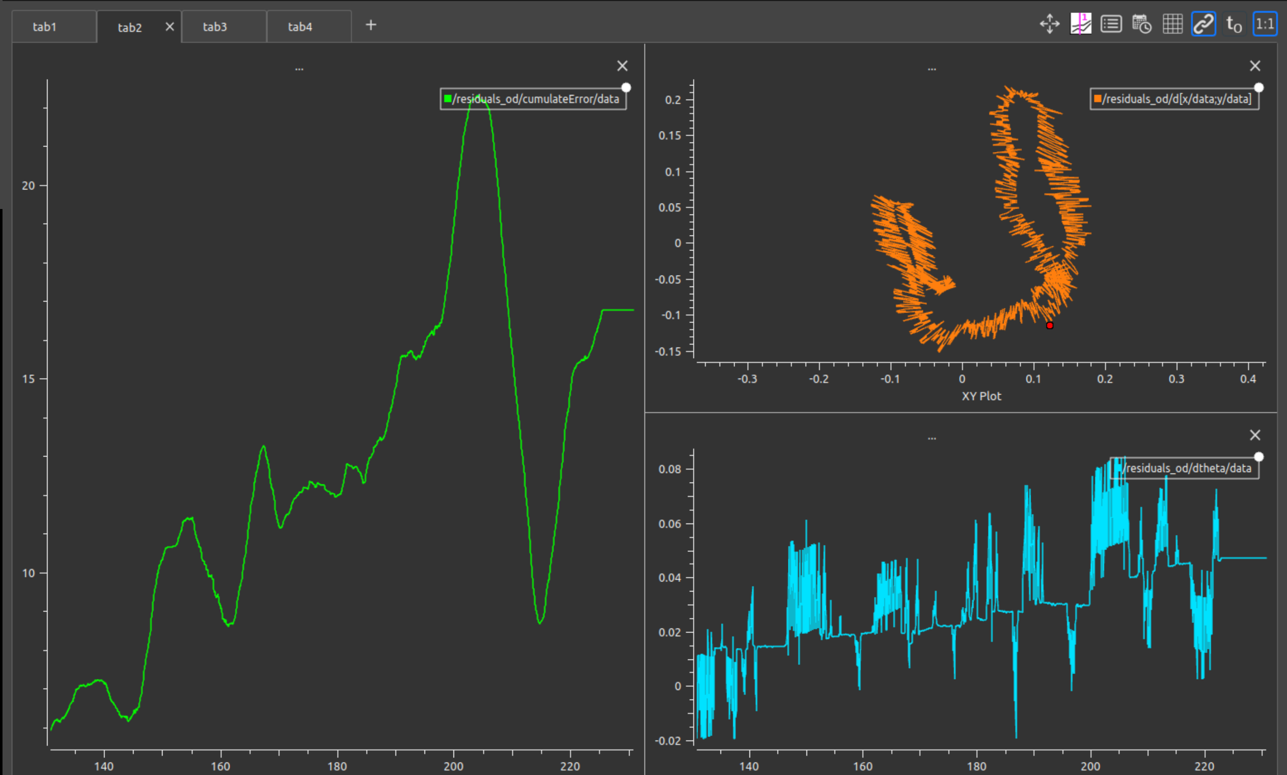
CHI: 1.75
RATIO: 1:38.7

VALIDATION – BAG 3


tab1
tab2
tab3
tab4
🟥/our_odom/pose/position/[x;y]
🟩/real_time_od/pose/position/[x;y]
XY Plot
🟥our_theta_360
🟩scout_theta_360

Odometry

Cost function


tab1
tab2
tab3
tab4
🟩/residuals_od/cumulateError/data
🟧/residuals_od/d/[x/data;y/data]
XY Plot
/residuals_od/dtheta/data

CHI: 1.75
RATIO: 1:38.7