5 Implementation

5.1 Calculation of process error and measurement error

- Measure the process covariance **Q** through observation: Let the robot move forward 2m under the velocity of 0.5 m/s [1] and record its position afterwards, the ground truth coordinate should be exactly 2m forward compared to the starting point. Measure the actual stop position. Repeat e.g. 100 times, which makes it reliable, although it took quite a lot of effort, through the following method, *Root Mean Square Error* $rmse = 0.001090280296483436$. Here in order to demonstrate the original calculation result, no round operation was conducted.

- As for the measurement covariance **R**, let the robot stay at specific positions for a short time period and gather some coordinate information, its ground truth position can be measured by *BOSCH Laser measure* and the measurement result comes from the *UWB tag*. Through this we can calculate the mean error, the variance of the error and Standard Deviation. *Root Mean Square Error* is the standard deviation of the residuals, which is the measurement noise in our case.

Through *describe()* method we can get details about X-coordinates ($expected\_x = 4.045$):

Listing 5.1: describe output

```
count    100.000000
mean       4.045630
std        0.022511
min        3.985000
25%        4.033750
50%        4.051500
75%        4.062000
max        4.086000
Name: x, dtype: float64
```

We can also plot its *Histograms with different binwidths*, as shown in Figure 5.1:
Its density plot is in Figure 5.2, which is essentially a smooth version of Histogram:
Standard deviation can be calculated by:

[1] 0.5 m/s is a proper velocity, lower velocity will make experiments less efficient and higher velocity will increase the process error because the robot move a specific distance in an accumulative way, the faster it would be harder to move exactly the specified distance.

```
def variance(data, ddof=0):
    n = len(data)
    mean = sum(data) / n
    return sum((x - mean) ** 2 for x in data) / (n - ddof)
def stdev(data):
    var = variance(data)
    std_dev = math.sqrt(var)
    return std_dev
```

The standard deviation for the process error is: 0.02241880295075522.

According to [Nil18], the spatial RMSE can be calculated separately for X- and Y-axis through:

$$RMSE_i = \sqrt{\frac{1}{n}\Sigma_{m=1}^{n}\Big(Est_i - Actual_i\Big)^2} \tag{5.1}$$

where i is the coordinate axis. And a net RMSE can be calculated through:

$$RMSE_{Net} = \sqrt{RMSE_X^2 + RMSE_Y^2} \tag{5.2}$$

The ground truth of the first three points we choose are:

1. pose1: (x = 3.916, y = 2.465)
2. pose2: (x = 5.143, y = 1.947)
3. pose3: (x = 6.641, y = 4.788)

We gathered their measurements from *UWB tag* and write the coordinate information into csv files and calculate the MSE using *mean_squared_error()* method *from sklearn.metrics*. And the calculation result is: $rmse$ = 0.01422606083083504. Their separate and concatenated density plot are in Figure 5.3.

In order to get representative measurement error, 25 poses are chosen across the lab. Their density plot and histogram is in Figure 5.4.

The standard deviation for the measurement error is: 0.07915516240539047.

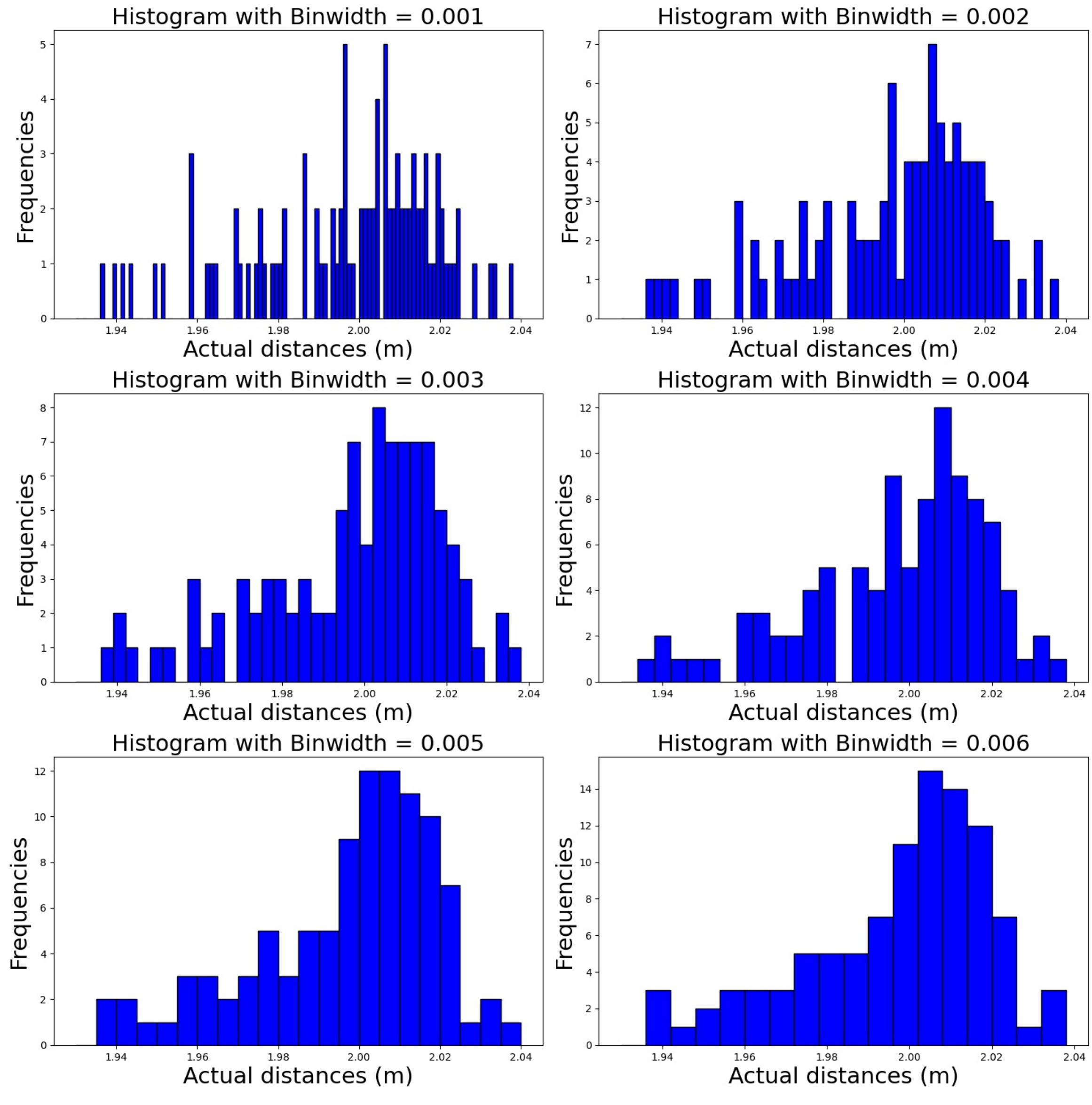
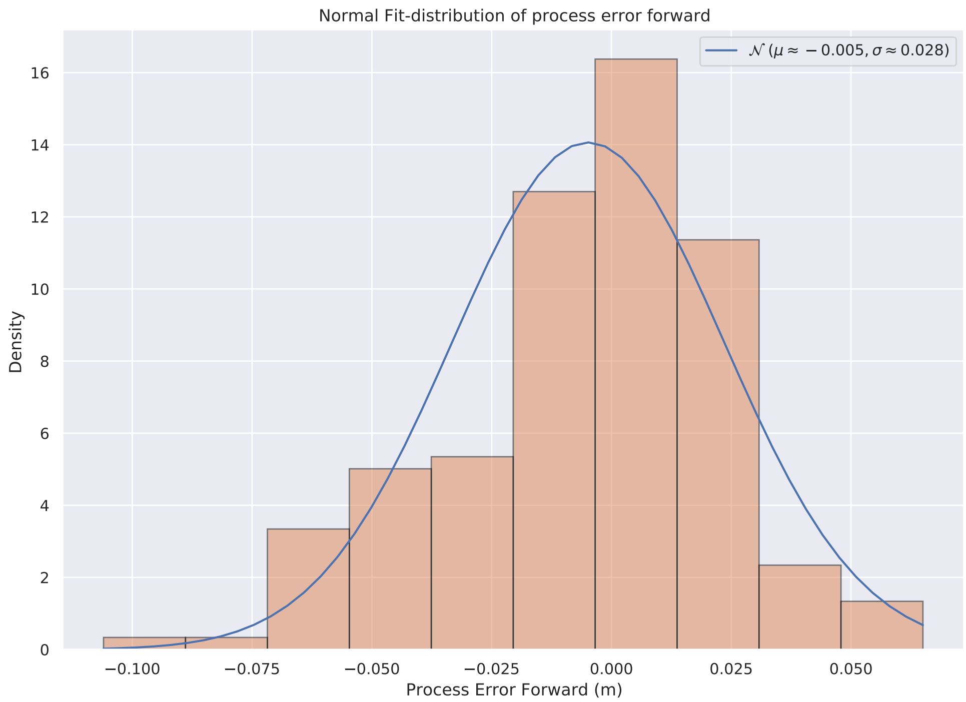


Figure 5.1: Histograms with different bin-widths for 2m movement



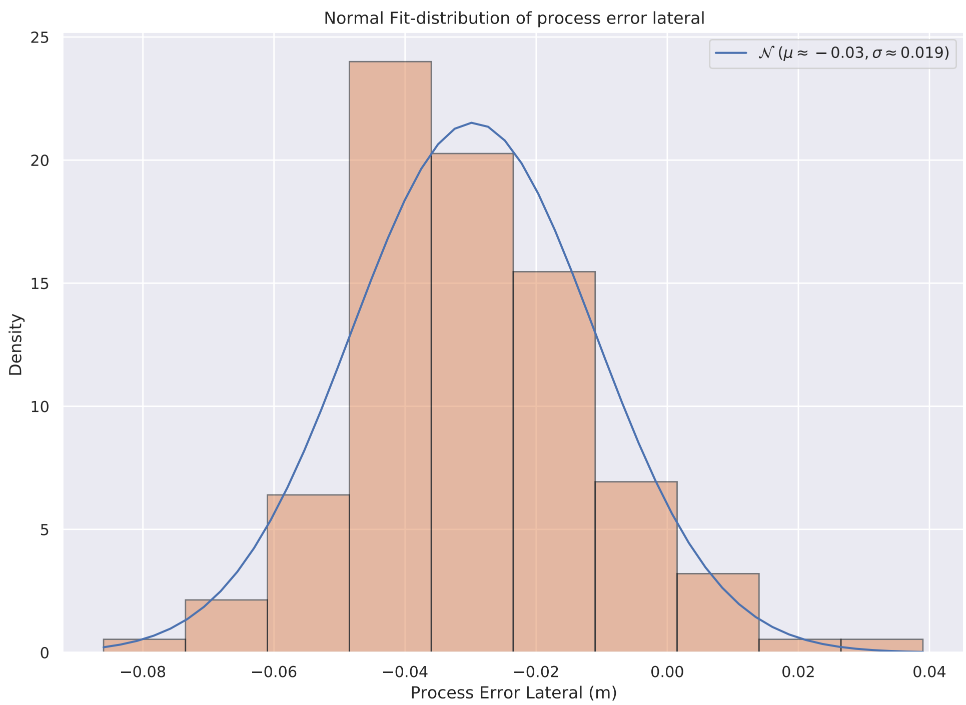
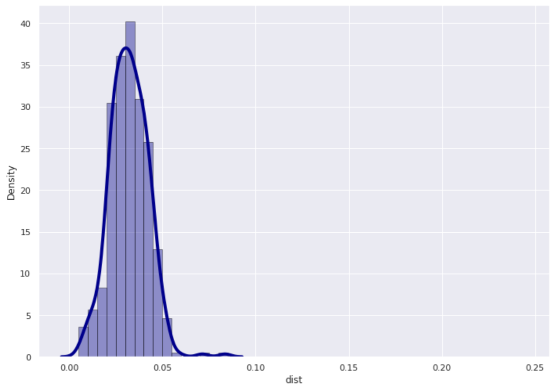
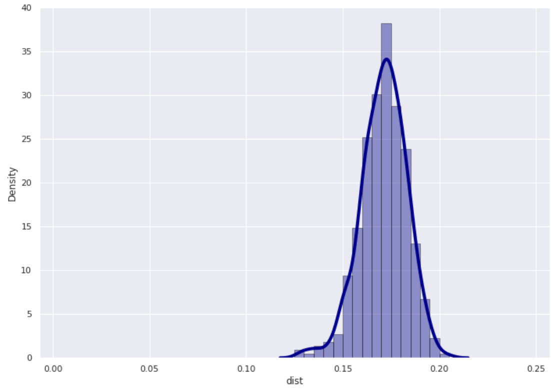


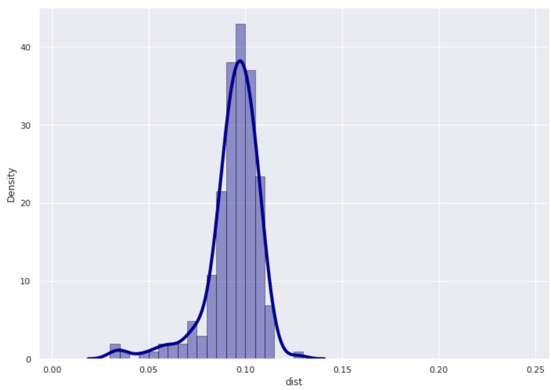
Figure 5.2: Density plot and histogram for process error forward and lateral based on 175 experiments with velocity from 0.5 m/s to 1.1 m/s



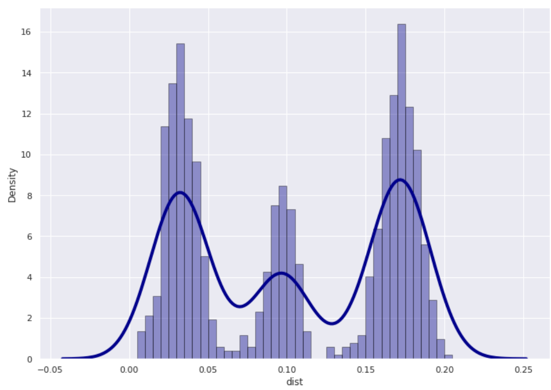
(a) Density plot for pose1



(b) Density plot for pose2

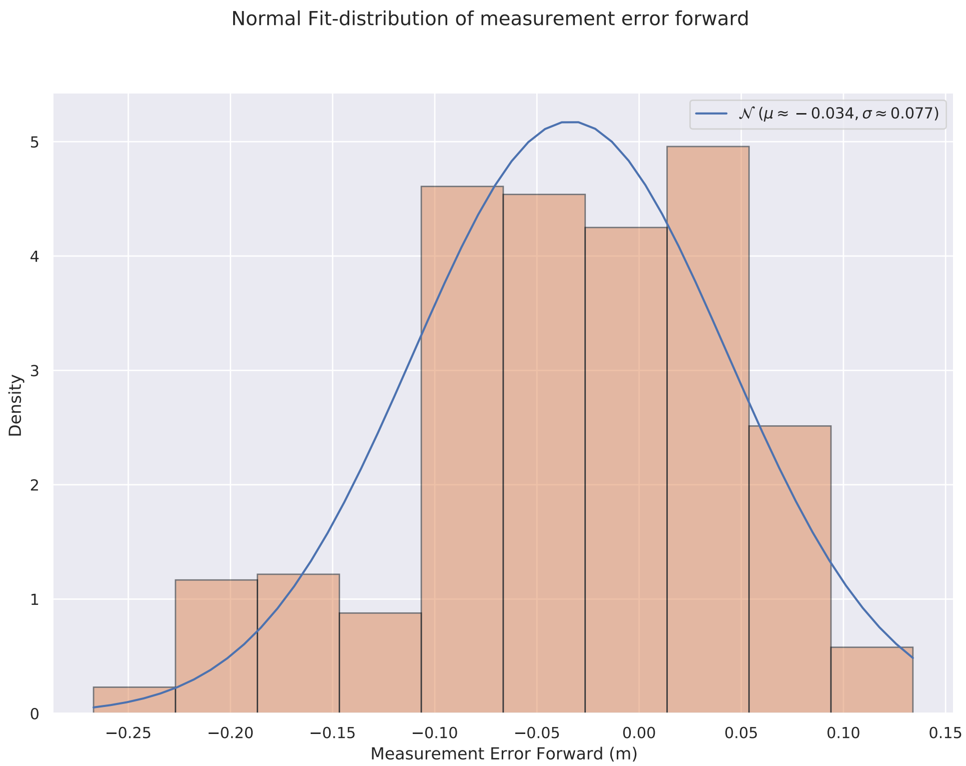


(c) Density plot for pose3



(d) Density plot for all these three poses

Figure 5.3: Density plots for measurement error for UWB tag at three randomly chosen positions



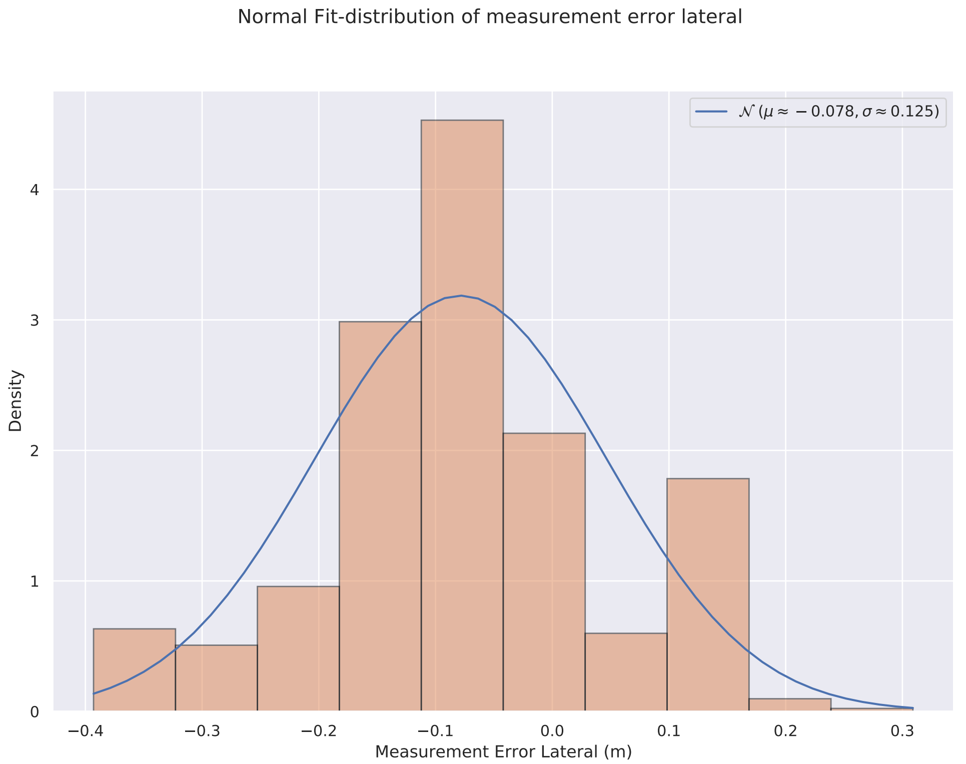


Figure 5.4: Density plot and histogram for measurement error forward and lateral based on 2500 data across 25 poses in the energy lab