Ex 2 Odometry, Dead Reckoning and Error predictions

Gao Sun, Yang Gao

Abstract

This exercise is mainly about the influence of uncertainties on the measurement of the robot's position. We have two varieties of robots. One is the two-wheel vehicle. At first, we let the encoder read the data of khepera_circle.txt. We set the error (variance) of the encoder is 0.5/12 to calculate the forward directions and headings of the robot. We obtain the co-variance of them is $\begin{bmatrix} 0.0000 & 0.0003 \\ 0.0003 & 0.0249 \end{bmatrix}$. Then we compare the variables of the position statements under different condition, which is whether we use the compensation term $\frac{\sin(\theta_n/2)}{\theta_n/2}$ in our model. In this process, we get the uncertainty of the robot position by using the Jacobian matrix. After doing this, we get the path plot and the state variables. Additionally, we take the uncertainty of wheelbase in consideration. By changing the wheelbase from 53mm to 45mm, and the wheel diameter from 15.3mm to 14mm, we also find that the results of the position are different. At last, we let the encoder read the data of 'khepera.txt' and do the same thing again. Another model is a three-wheel robot. The encoder reads the data of 'snowhite.txt'. In this model, uncertainty of sample time, speed of the robot and the steering angel can also produce errors on the state variables. We get the co-variance of position parameters of a 4050×9 matrix. Besides, we set the steering angle error of 0.01, speed error of 5 and sample time error of 0.0001. We plot the errors between true and estimated values, and the estimated standard deviation of the position parameters. Through this, we find that these to stay very close to each other.

Key words: robot, uncertainty, error, Jacobian, state variables, co-variance.

1. Introduction

Determining the location of a robot is an important problem in navigating an autonomous vehicle in an unstructured environment [1]. In this experiment, what we are discussing is a two-wheel and a three-wheel drive system. For analyzing the path of the robot, we use three parameters (x, y, θ) to describe the position and orientation of it. On the base of geometry, we assume that the entire vehicle's traveling trajectory looks like a circle. We take the data coming from the encoders to do calculation.

Actually, the measurements from the optical encoders are not error-free[1]. There exists uncertainty in the movements of the robot. In order to decrease the effects of that, we take the Jacobian matrix as the uncertainty measure for the location estimation.

2. Theory and method

2.1 Geometry model of the two-wheel driving system (Odometry)

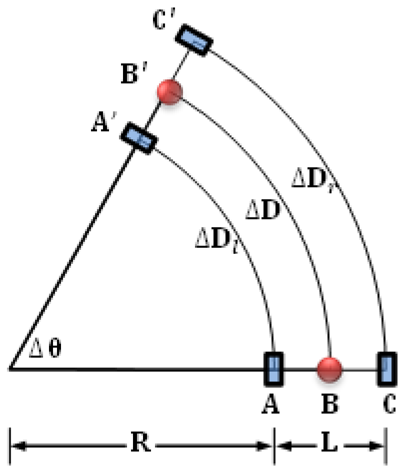

Figure 2.1 Two-wheel driving geometry model

According to the paper of 'Location estimation and uncertainty analysis for mobile robots' written by Wang, we have relation as follows:

$$\Delta D_r = (L+R)\Delta\theta;$$
$$\Delta D_l = R\Delta\theta;$$
$$\Delta D = (\Delta D_r + \Delta D_l)/2;$$
$$\Delta\theta = (\Delta D_r - \Delta D_l)/L;$$

In the equations, L is the distance between two wheels; R is the distance from the left wheel to the center of the circle which means the locus; $\Delta\theta$ is the changed angle of one period in the robot's movement; ΔD_r and ΔD_l denote the covered distance of the right and left wheels respectively.

To get the three parameters (x, y, θ), roughly, under the approximation of each moving segment of the robot to a straight line we can get the following relation:

$$\Delta x_n = x_{n-1} + \Delta D_n \cos(\theta_{n-1} + \frac{\Delta\theta_n}{2})$$
$$\Delta y_n = y_{n-1} + \Delta D_n \sin(\theta_{n-1} + \frac{\Delta\theta_n}{2})$$
$$\Delta\theta_n = \theta_{n-1} + \Delta\theta_n$$

To get a more accurate result, we can approximate the each moving segment into an arc, then we need to add some compensate factor:

$$x_n = x_{n-1} + \frac{\sin(\Delta\theta_n/2)}{(\Delta\theta_n/2)}\Delta D_n \cos(\theta_{n-1} + \frac{\Delta\theta_n}{2})$$
$$y_n = y_{n-1} + \frac{\sin(\Delta\theta_n/2)}{(\Delta\theta_n/2)}\Delta D_n \sin(\theta_{n-1} + \frac{\Delta\theta_n}{2})$$
$$\theta_n = \theta_{n-1} + \Delta\theta_n$$

2.2 Geometry model of the three-wheel driving system (Dead Reckoning)

The three-wheel driving system we use in this exercise can measure the speed of the driving wheel and the angel of the steering wheel. The geometry model of this system is shown in Figure 2.2.

We can calculate the change in the robot's forward and rotational displacements like this:

$$\Delta D = v(t)T\cos[\alpha(t)] \qquad \Delta\theta = \frac{v(t)T\sin[\alpha(t)]}{L}$$

In the equation, $v(t)$ and $\alpha(t)$ refers to the data of driving wheel and the angel of the steering wheel respectively. So we can update the robot's position data through these equations:

$$x_k = x_{k-1} + v\cos(\alpha)T\cos(\theta_{k-1} + \frac{v\sin(\alpha)T}{2L})$$
$$y_k = y_{k-1} + v\cos(\alpha)T\sin(\theta_{k-1} + \frac{v\sin(\alpha)T}{2L})$$
$$\theta_k = \theta_{k-1} + \frac{v\sin(\partial)T}{2}$$

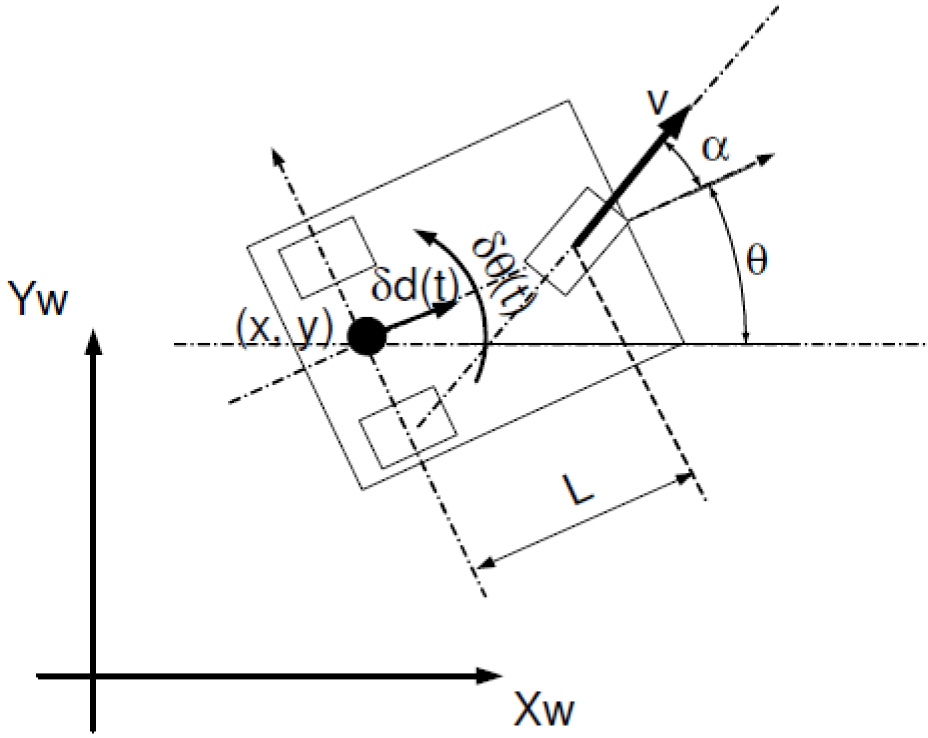

Figure 2.2 Three-wheel driving geometry model

2.3 Error Propagation

In mobile robotics we need a way of estimating robot's position uncertainty. The uncertainty is represented by the covariance matrix Σ_k of the pose $X_k = (x, y, \theta)$. The uncertainty of the new position Σ_k is based on previous state uncertainty Σ_{k-1}, and the additive uncertainty depending on current movement[2]. To describe it in the mathematical equation:

$$\Sigma_{X_k} = J_{x_{k-1}} \Sigma_{X_{k-1}} J_{x_{k-1}}{}^T + J_u \Sigma_u J_u{}^T$$

In the equation, $J_{X_{k-1}}$ refers to the Jacobian Matrix which is about the previous system state. We can calculate this by calculating the partial derivative of each propagation function to $x_{k-1}, y_{k-1}, \theta_{k-1}$ respectively. In this way, we can get a 3×3 Matrix. Similarly, J_u refers to the Jacobian Matrix which is about all the sources of uncertainty in the system. We can calculate this by calculating the partial derivative of each propagation function to each variable which will cause uncertainty respectively. If we have n variables which can cause uncertainty, then we can get a $3 \times n$ Matrix. Σ_u is the $n \times n$ co-variance matrix of the variables which can cause uncertainty to the system. Normally, the different variables which can cause uncertainty are all independent from each other, so only the diagonal of co-variance matrix Σ_u has some value, and the value in other places in this matrix should be all 0.

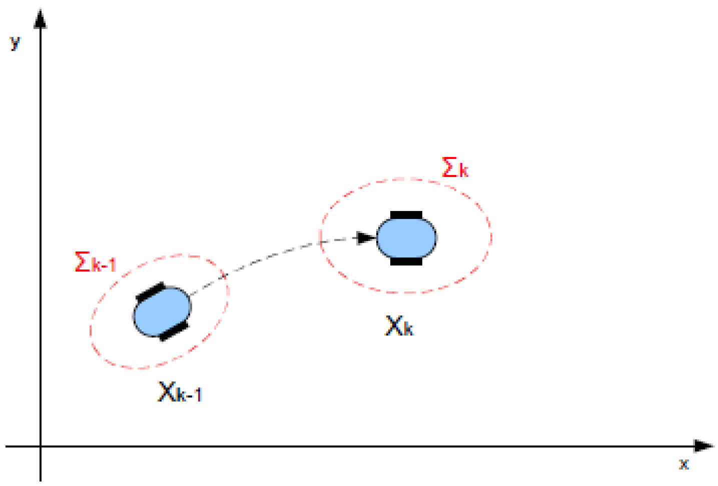

Figure 2.3 Prediction of robot uncertainty based on previous state uncertainty and current movement

To make the uncertainty analysis more concrete, we usually plot an ellipse to describe the uncertainty like in Figure 2.3. The practical meaning of the ellipse is that there will be a possibility of 68.3% that the position of the robot is inside of the ellipse. So we can say that the bigger the ellipse is, the greater uncertainty that state is.

3. Experiments and results

3.1 Differential drive (Khepera mini robot)

Part 1: Read the data of Khepera_circle.txt.

(1). As we have discussed in figure 2.1, we have the following relation:

$$\Delta D_r = (L + R)\Delta\theta;$$
$$\Delta D_l = R\Delta\theta;$$
$$\Delta D = (\Delta D_r + \Delta D_l)/2;$$
$$\Delta\theta = (\Delta D_r - \Delta D_l)/L;$$

Then we can get the forward direction and heading, and also the co-variance of these two.

After the measurement, at the last movement, both the value of the forward direction and heading are 0. And the covariance of these two is $\begin{pmatrix} 0.0000 & 0.0003 \\ 0.0003 & 0.0249 \end{pmatrix}$. About the uncertainties we assume, we assume the uncertainties of both right and left wheel are the encoder error of 0.5/12.

(2). In the question of this part, the assumption of the robot's movement is a circular, which is the same condition as in Figure 2.1. For calculating the state variables, we just need to use the formulations as follows:

$$x_k = x_{k-1} + \begin{pmatrix} \Delta d_k \cos(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ \Delta d_k \sin(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ \Delta\theta_k \end{pmatrix} = \begin{pmatrix} x_{k-1} + \Delta d_k \cos(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ x_{k-1} + \Delta d_k \sin(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ \theta_{k-1} + \Delta\theta_k \end{pmatrix}$$

which stands for the three parameters used to describe the robot's position in the world co-ordinate. After we calculate the state variables recursively, we plot them into 3 figures which display how the three parameters changes, viewed by Figure 3.1 below.

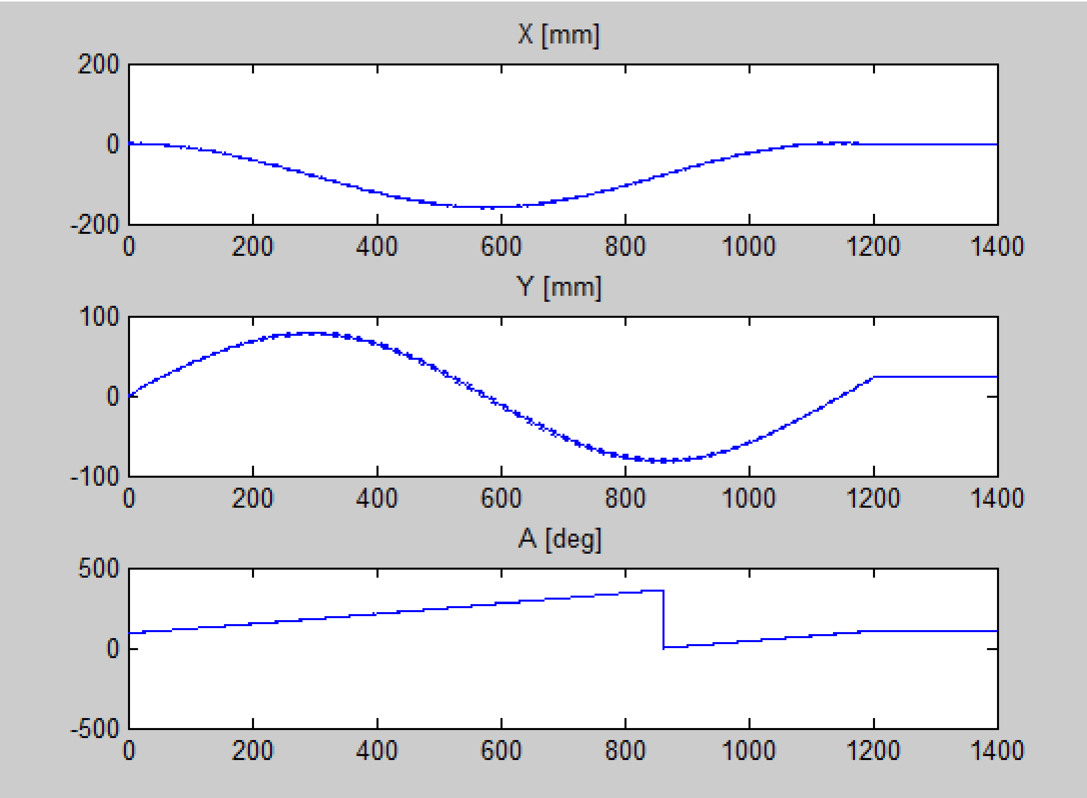

Figure 3.1 : The solid lines are predicted state variables x,y,θ in world coordinate respectively, and the dashed lines show the standard deviation

Taking errors into consideration, we use the Jacobian matrix to obtain the precision result. To calculate the uncertainty of the position update, the law of error propagation is used:

$$\sum x_k = J_{x_{k-1}} \sum x_{k-1} J_{x_{k-1}}{}^T + J_{\Delta d \Delta\theta} \sum_{\Delta d \Delta\theta} J_{\Delta d \Delta\theta}{}^T$$

The Jacobian matrix of $x_{k-1}, y_{k-1}, \theta_{k-1}$ is shown below:

$$J_{X_{k-1}} = \begin{pmatrix} 1 & 0 & -\Delta d_k \cos(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ 0 & 1 & \Delta d_k \sin(\theta_{k-1} + \frac{\Delta\theta}{2}) \\ 0 & 0 & 1 \end{pmatrix}$$

The Jacobian matrix of the uncertainty in passing distance and angle is shown below:

$$J_{\Delta d \Delta\theta} = \begin{pmatrix} \cos(\theta_{k-1} + \frac{\Delta\theta_k}{2}) & -\frac{\Delta d_k}{2} \sin(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ \sin(\theta_{k-1} + \frac{\Delta\theta_k}{2}) & \frac{\Delta d_k}{2} \cos(\theta_{k-1} + \frac{\Delta\theta_k}{2}) \\ 0 & 1 \end{pmatrix}$$

In this exercise, we assumed the original position of the robot is (0, 0, 90*pi/180), and the initial uncertainty

$$\sum x_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & (\frac{\pi}{180})^2 \end{bmatrix}$$

With the geometry model of the two-wheel model, we can get the co-variance matrix of the Δd and $\Delta\theta$ should be:

$$\sum_{\Delta d \Delta\theta} = \begin{bmatrix} \frac{\sigma_r^2 + \sigma_l^2}{4} & 0 \\ 0 & \frac{\sigma_r^2 - \sigma_l^2}{L^2} \end{bmatrix} \approx \begin{bmatrix} 0.000868 & 0 \\ 0 & 0 \end{bmatrix}$$

In the next step, through the calculation, we can get the co-variance of these new positions variables is a 1400×9 matrix. Then we plot each co-variance matrix into the ellipse, which is shown in Figure 3.2 below.

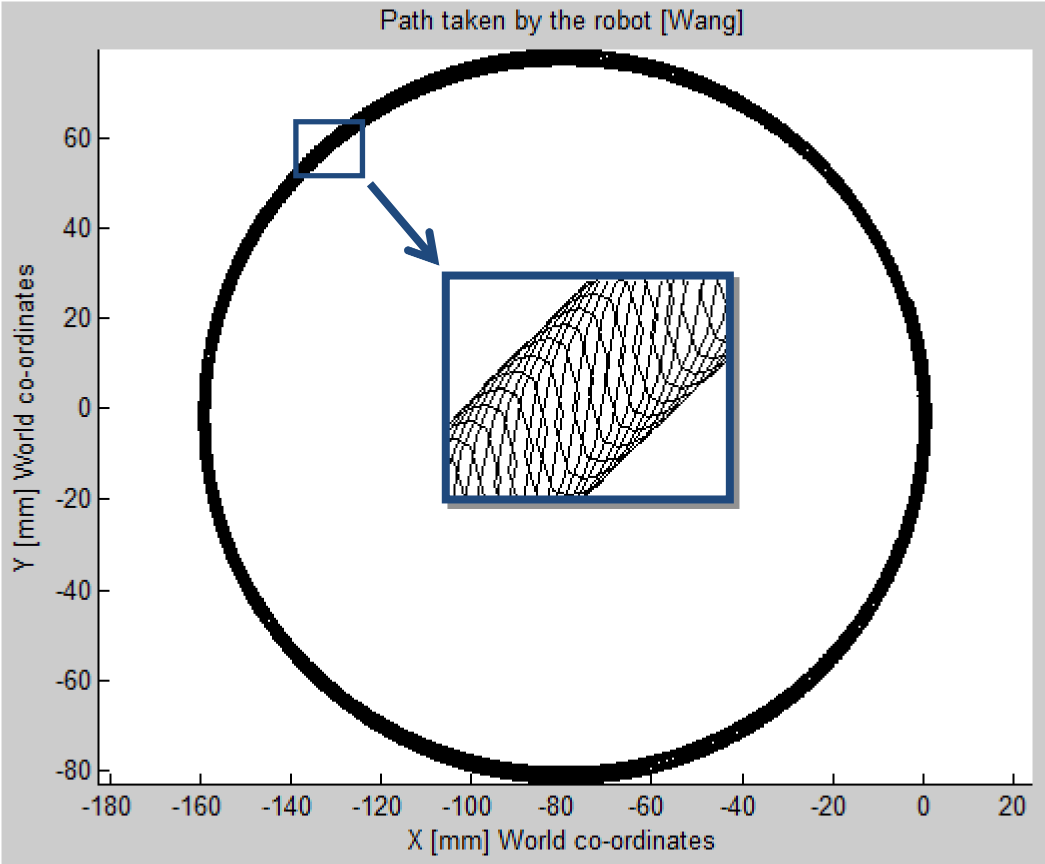

Figure 3.2 : Uncertainty of each sample point describe by uncertainty ellipse

(3). In this part, we should add the compensation term in Wang paper, which means we should use the factor of $\dfrac{\sin(\theta_n/2)}{\theta_n/2}$ in our formulations for the statement below:

$$x_k = x_{k-1} + \begin{pmatrix} \dfrac{\sin(\triangle\theta_k/2)}{\triangle\theta_k/2}\triangle d_k \cos(\theta_{k-1} + \dfrac{\triangle\theta_k}{2}) \\ \dfrac{\sin(\triangle\theta_k/2)}{\triangle\theta_k/2}\triangle d_k \sin(\theta_{k-1} + \dfrac{\triangle\theta_k}{2}) \\ \triangle\theta_k \end{pmatrix} = \begin{pmatrix} x_{k-1} + \dfrac{\sin(\triangle\theta_k/2)}{\triangle\theta_k/2}\triangle d_k \cos(\theta_{k-1} + \dfrac{\triangle\theta_k}{2}) \\ x_{k-1} + \dfrac{\sin(\triangle\theta_k/2)}{\triangle\theta_k/2}\triangle d_k \sin(\theta_{k-1} + \dfrac{\triangle\theta_k}{2}) \\ \theta_{k-1} + \Delta\theta_k \end{pmatrix}$$

At the same time, the Jacobian matrix of $x_{k-1}, y_{k-1,}\theta_{k-1}$ also has changed. As the new Jacobian matrix is too complicated, we don't show it here. Then we plot the new co-variance matrix and the new state variable. We find the difference between the result we get without compensation is really small. It's very difficult to tell the difference just by looking at the figures of the co-variance matrix and the state variable. Instead, we calculate the difference between the data with and without compensation term and plot it into figure 3.3:

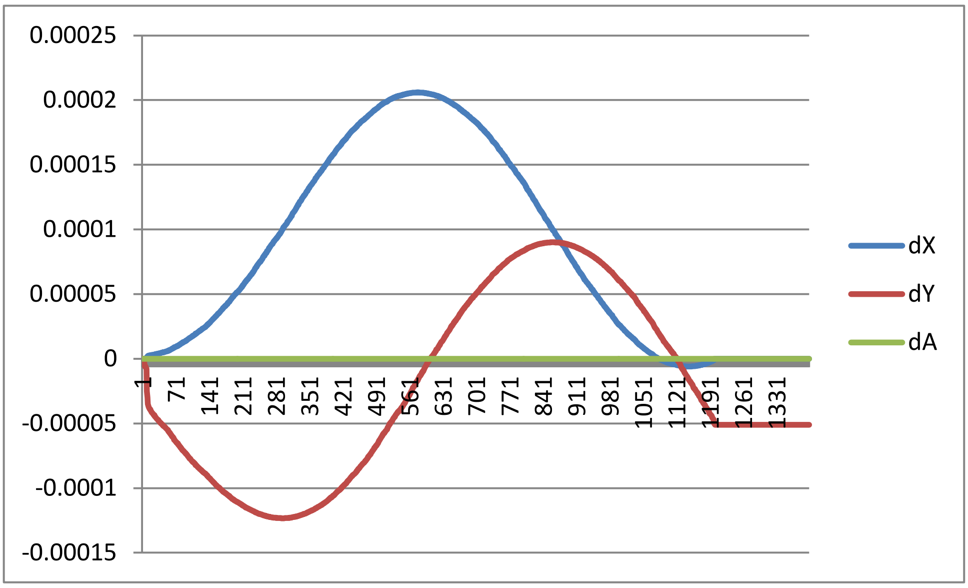

Figure 3.3 : The difference between the data with and without compensation, blue refers to the X-coordinate, red refers to the Y-coordinate and green refers to the heading angle

We can see from the Figure 3.3, the difference between them is really small, even the largest difference is only 0.000206mm. It is just because the sample rate we use is high enough so that the moving distance for the robot between two sampling points is rather short. So the distance which the robot moves for during one sampling period is really short. So the distance of the straight line between the two points where the robot is at the two adjacent sampling time points is quite close to the arc between those two points. For example, see the Figure 3.3 on the right side, $\angle ABC = \Delta\theta_k/2$, if the sampling rate is high enough, $\Delta\theta_k/2$ will be very small, so $sin(\Delta\theta_k/2)\,/(\Delta\theta_k/2)$ will be very close to 1, so the length of $arc\ OP$ will be very close to the length of straight line $|OP|$.

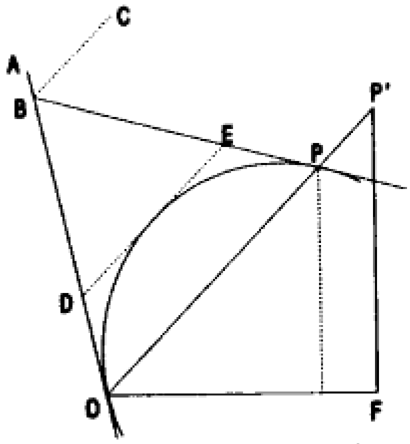

Figure 3.4: Geometry model of the two-wheel robot

As it is mentioned in the question, if the encoder values were not read as often as they are, we need to analyze the changes about the state variables and the co-variance matrix. For solving this, in our code, we just need to set a sample period, turning it from 1 to 2,5 or 10, then observe what happens to the variables we want to discuss. Again, it is still difficult to see the difference from the state variable figure. But if we calculate the co-variance matrix for the Δd and $\Delta\theta$, we can find pattern from it.

When sample period is 1 time unit, the co-variance matrix is: $\begin{bmatrix} 0.0249 & 0.0003 \\ 0.0003 & 0 \end{bmatrix}$.

When sample period is 2 time units, the co-variance matrix is: $\begin{bmatrix} 0.0955 & 0.0012 \\ 0.0012 & 0 \end{bmatrix}$.

When sample period is 5 time units, the co-variance matrix is: $\begin{bmatrix} 0.5729 & 0.0072 \\ 0.0072 & 0.0001 \end{bmatrix}$.

When sample period is 10 time units, the co-variance matrix is: $\begin{bmatrix} 2.2225 & 0.0282 \\ 0.0282 & 0.0004 \end{bmatrix}$.

So it's easy to see that the variance and the co-variance are becoming greater when the sampling period grows. This is because when the sampling points become seldom, the difference between each Δd and $\Delta\theta$ becomes greater. After viewing those co-variance matrixes, we can say it starts to differ greatly when the sampling period changes from 5 time units to 10 time units.

(4). In this question, we are required to get variation trend of the covariance matrix. As when the sampling rate is too high, the uncertainty figure we get from Matlab will be too dense, which is difficult for us to recognize how the ellipse changes. So we use the sampling period of 10 time units to do the analysis. It is shown in Figure 3.5 below.

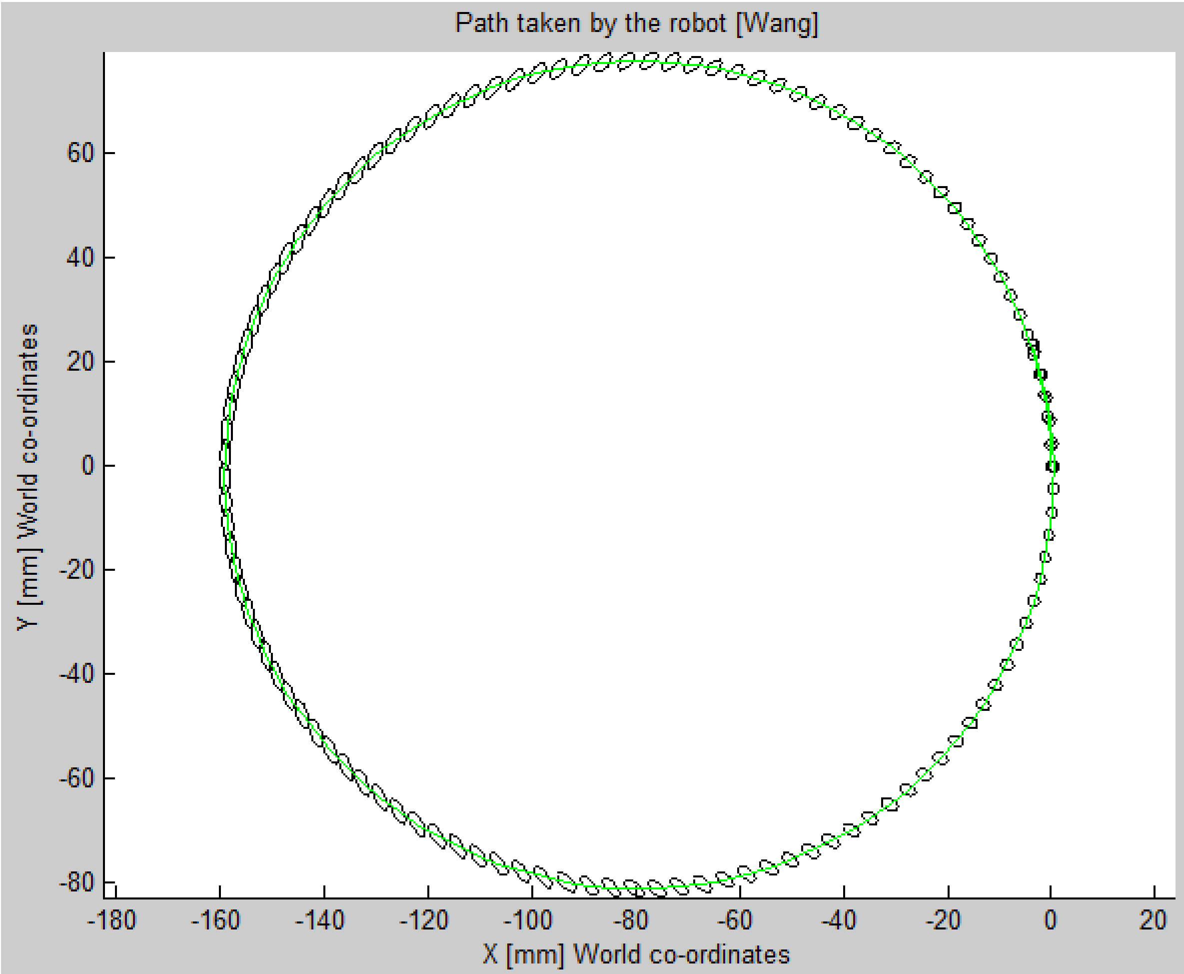

Figure 3.5: Variation trend of the covariance matrix

From the figure, we can find the uncertainty ellipse appears to be smallest in the beginning, and at that time, it is nearly a round, which means it has almost the same uncertainty both in X-direction and Y-direction. After the robot moves further and further to the starting point, the uncertainty ellipse becomes bigger, and looks less round as before. The ellipse always becomes longer in the direction the robot is moving. That means we have greater uncertainty in the direction which the robot is moving. When the robot finishes the first half circle path, the uncertainty at that time appears to be the greatest, and the ellipse shows to be vertical in its direction. Then on the way where the robot moves on the remaining half circle, the uncertainty ellipse becomes smaller and still becomes longer in the direction the robot is moving. It becomes almost round again when it goes back to its previous starting point. But anyway, the ellipse on the second half circle is a little bigger than the ellipse on the first half circle because the path the robot moves is longer, so that we have greater uncertainty. So the uncertainty we get is realistic.

(5). In this part, what we are discussing is about effects to the uncertainty when we use the new variables of the wheel base and wheel diameter. First we just simply change the wheel base into 45mm and wheel diameter into 14mm. Also take the sampling period of 10 time units to get an easy recognized uncertainty figure which is shown in Figure 3.6.

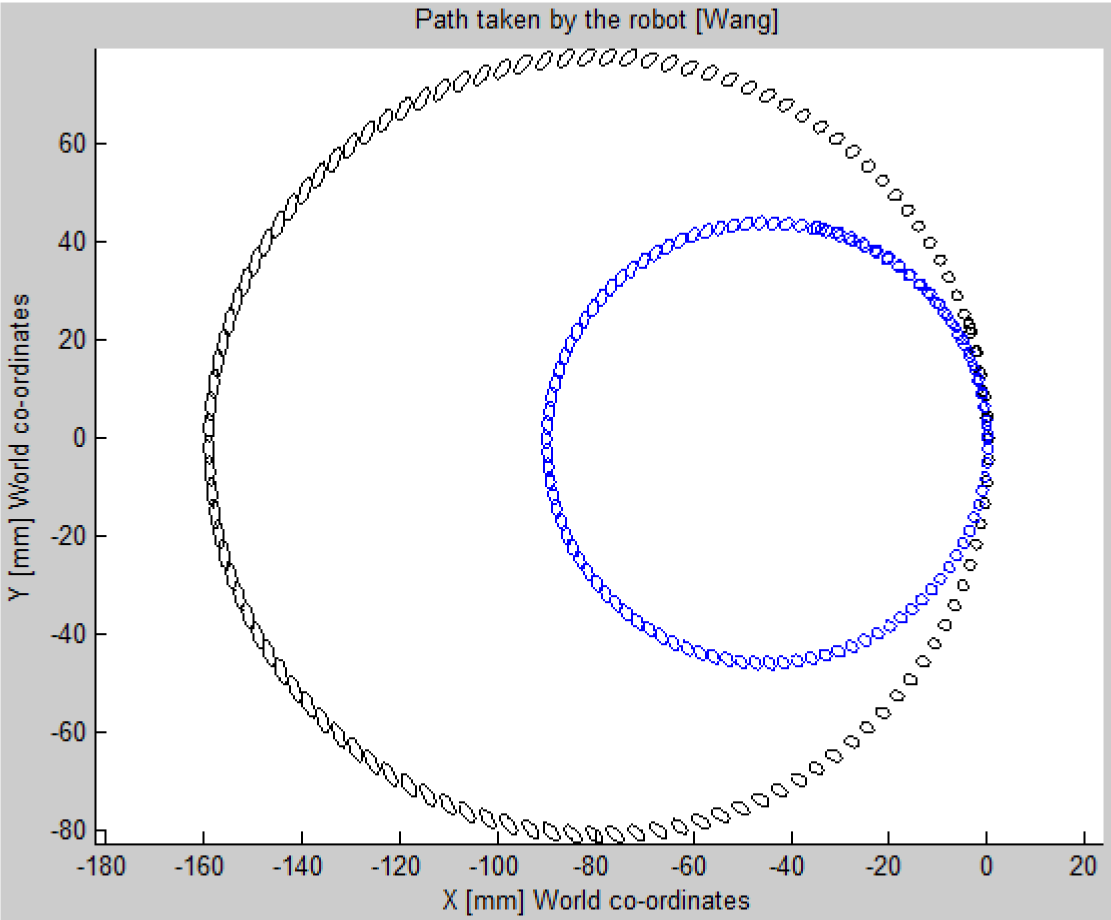

Figure 3.6: Comparing the uncertainty of original robot and of a smaller size robot, blue one stands for the smaller size robot, black one stands for the original size robot

In this figure, we can see that the uncertainty of the small size robot(with the wheels base and diameter is shorter) is smaller than the original size robot. This is easy to understand, because the size of the robot is smaller, the error it produce will also be smaller in some certain proportion.

Next, we want to further analyze how the new uncertain parameter affects the system. For easy calculation, we just add the uncertain of the wheel base into the model of the system.

So with the uncertain of wheel base, the transition function of the position of the robot should be:

$$X_k = X_{k-1} + \begin{pmatrix} \Delta d \cos(\theta_{k-1} + \frac{\Delta\theta}{2}) \\ \Delta d \sin(\theta_{k-1} + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{pmatrix} = \begin{pmatrix} x_{k-1} + \frac{\Delta R + \Delta L}{2} \Delta d \cos(\theta_{k-1} + \frac{\Delta R - \Delta L}{2b}) \\ y_{k-1} + \frac{\Delta R + \Delta L}{2} \Delta d \sin(\theta_{k-1} + \frac{\Delta R - \Delta L}{2b}) \\ \theta_{k-1} + \frac{\Delta R - \Delta L}{b} \end{pmatrix}$$

In the equation, △R stands for the distance driven by the right wheel; △L stands for the distance driven by the left wheel; b stands for the wheel base.

To calculate the variance of the position update the law of error propagation is used. As it is shown below:

$$\sum x_k = J_{x_{k-1}} \sum x_{k-1} J_{x_{k-1}}{}^T + J_{\Delta R\Delta L} \sum_{\Delta R\Delta L} J_{\Delta R\Delta L}{}^T + J_b \sum_b J_b{}^T$$

Then we should calculate the new Jacobian matrix. The Jacobian matrix $J_{X_{k-1}}$ is about the parameters $x_{k-1}, y_{k-1,}\theta_{k-1}$. It has already been shown before, so we don't repeat it again here. The Jacobian matrix $J_{\Delta R\Delta L}$ is about the parameters of ΔR and ΔL.

$J_{\Delta R\Delta L} =$

$$\begin{bmatrix} \frac{cos\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2} + \frac{\frac{\Delta L + \Delta R}{2b} sin\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b} & \frac{cos\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2} - \frac{\frac{\Delta L + \Delta R}{2b} sin\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b} \\ \frac{sin(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2} - \frac{\frac{\Delta L + \Delta R}{2b} cos\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b} & \frac{sin\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2} + \frac{\frac{\Delta L + \Delta R}{2b} cos\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b} \\ -\frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

The Jacobian matrix J_b is about the parameters of the wheel base b.

$$J_b = \begin{bmatrix} -\frac{\frac{\Delta L + \Delta R}{2} sin\,(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b^2} \\ \frac{\frac{\Delta L + \Delta R}{2} cos(\theta_{k-1} - \frac{\Delta L - \Delta R}{2b})}{2b^2} \\ \frac{\Delta L - \Delta R}{b^2} \end{bmatrix}$$

As for the co-variance matrix of △R and △L, we set the variance of them the value of the error of the encoder, and assume the left and right encoder working independently.

$$\sum_{\Delta R\Delta L} = \begin{bmatrix} (0.5/12)^2 & 0 \\ 0 & (0.5/12)^2 \end{bmatrix}$$

$\sum_{\mathrm{b}}$ is the variance of the wheel base. According to the exact value of the wheel base of $53\ mm$, we set $\sum_{\mathrm{b}} = 5$.

Using all these states above, we plot the uncertainty which is shown in Figure 3.7. In the figure, we can see that after adding the wheel base as the new uncertainty, the uncertainty of the robot is increasing all the time instead of first increasing then decreasing. So if we have uncertainty in the wheel base, it will have really bad effect to the accuracy of the robot localization.

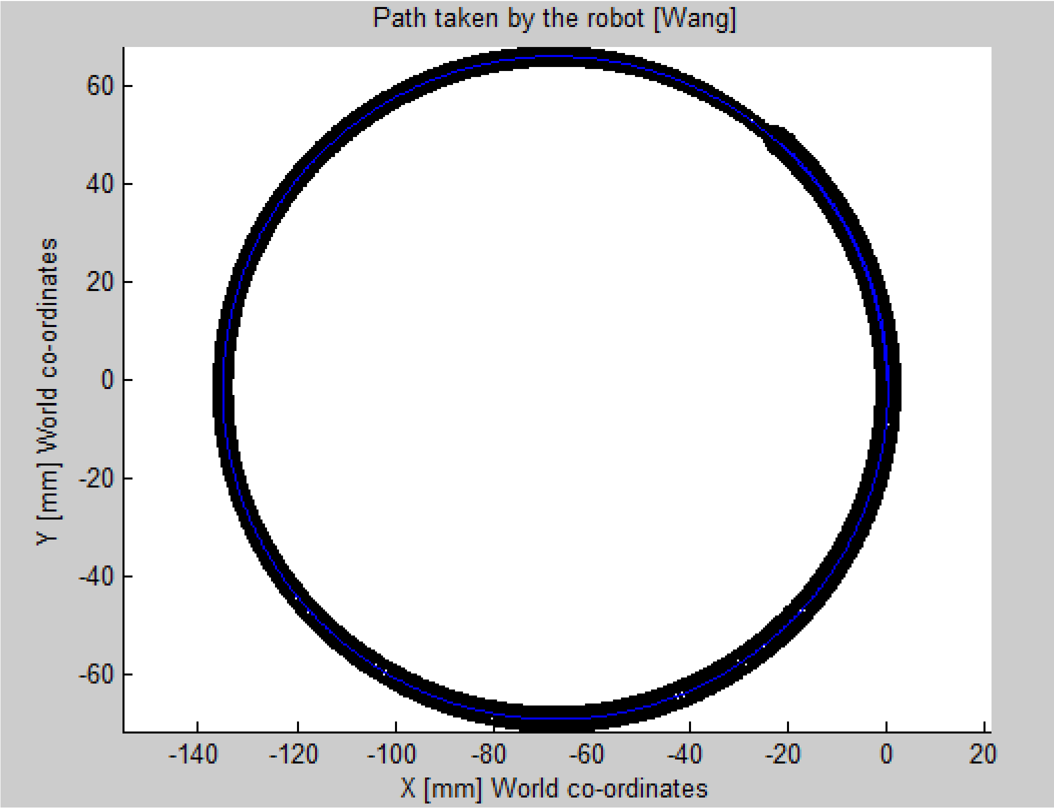

Figure 3.7: The blue circle is the predict locus of the robot, and the black part around it is the uncertainties.

Part 2. Read the data of ‘Khepera.txt’.

In the requirement of this question, we should use another data file, what we need to do is reading another file’s data. And the theory and method are the same as we talked about above. So we don’t repeat the theory again. We just list the result below.

In the code, we just need to change the sentence ENC = load(‘khepera_circle.txt’) to ENC = load(‘khepera.txt’), then run part of the code. We’ll get the change in the forward direction in the last movement is -0.0831; the change of the heading is 7.1699. And the co-variance of these two is $\begin{pmatrix} 7.9229 & -0.0644 \\ -0.0644 & 0.0048 \end{pmatrix}$.

The variables of position statements are shown in the figure 3.8 below. The co-variance matrix of these new positions is becoming an 890×9 matrix, which is plotted into ellipses showed in figure 3.9 below.

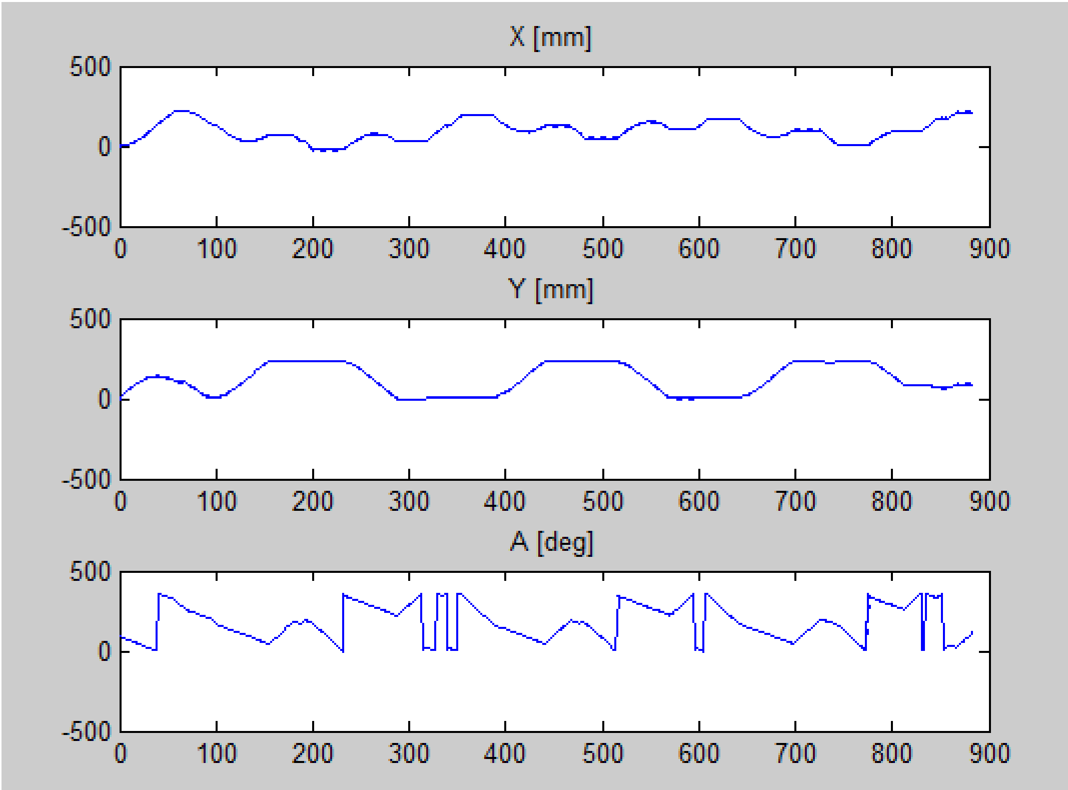

Figure 3.8: The plots are x,y,θ in world coordinate separately after reading the new data

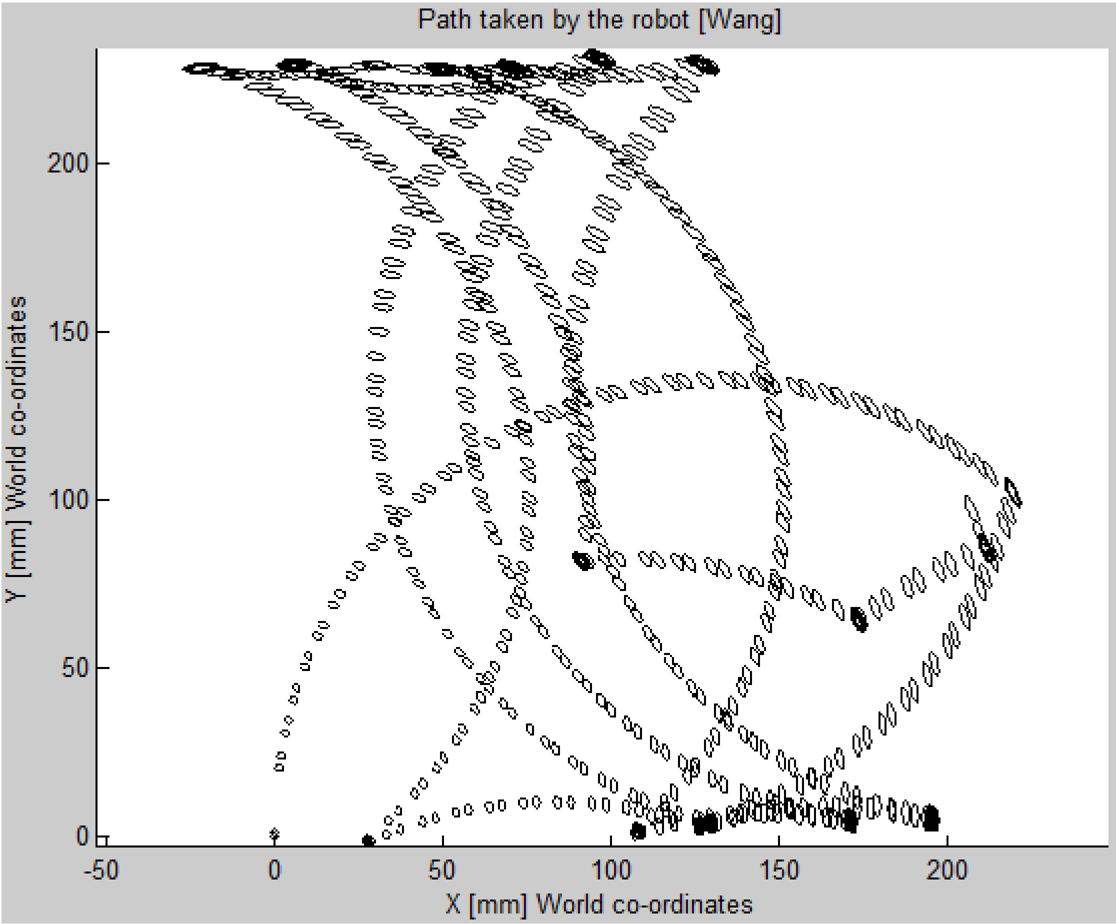

Figure 3.9: Uncertainty of each sample point sampling every 1 time unit

Then we compare the difference of the data with compensation and without compensation, showed in Figure 3.10:

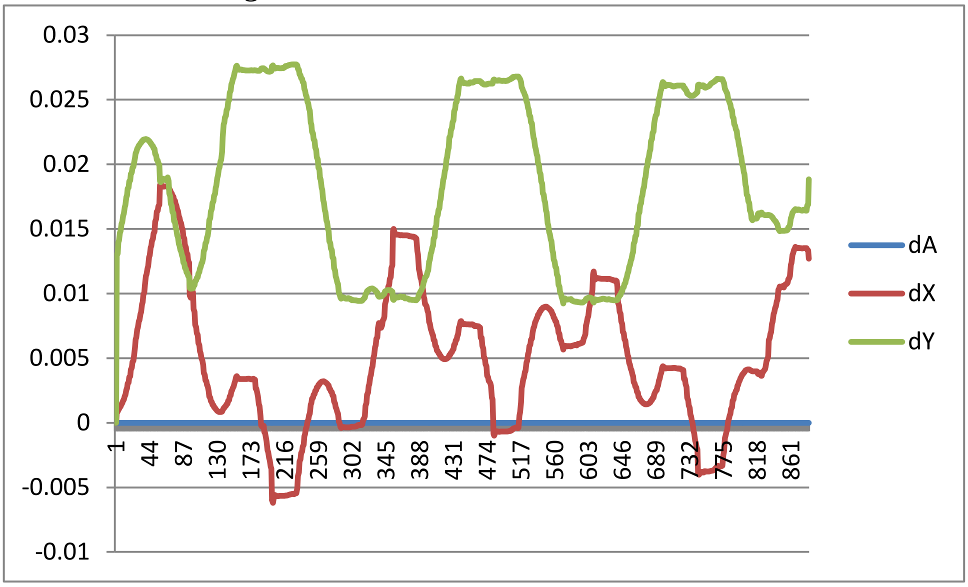

Figure 3.10 : The difference between the data with and without compensation, red refers to the X-coordinate, green refers to the Y-coordinate and green refers to the heading angle

The difference is still quite small, but is much bigger than the difference showed in Figure 3.3. This is because that the distance between each sampling point is much bigger now.

Again looking at the Figure 3.9, the variation trend of the uncertainty is similar with which we analyzed before. That is, the father we move away from the starting point, the greater uncertainty we have, and the uncertainty ellipse always becomes longer at the direction of the heading angle that the robot is moving.

Next we should analyze how the state variables change with different sampling periods. As we try, we still find that it starts to differ greatly at sampling period of 5, because when the sampling become seldom, the difference between each Δd and $\Delta\theta$ becomes greater. Figure3.11 compares the state variables with the sampling period of 1 and 5, and the state variables with the sampling period of 1 and 10:

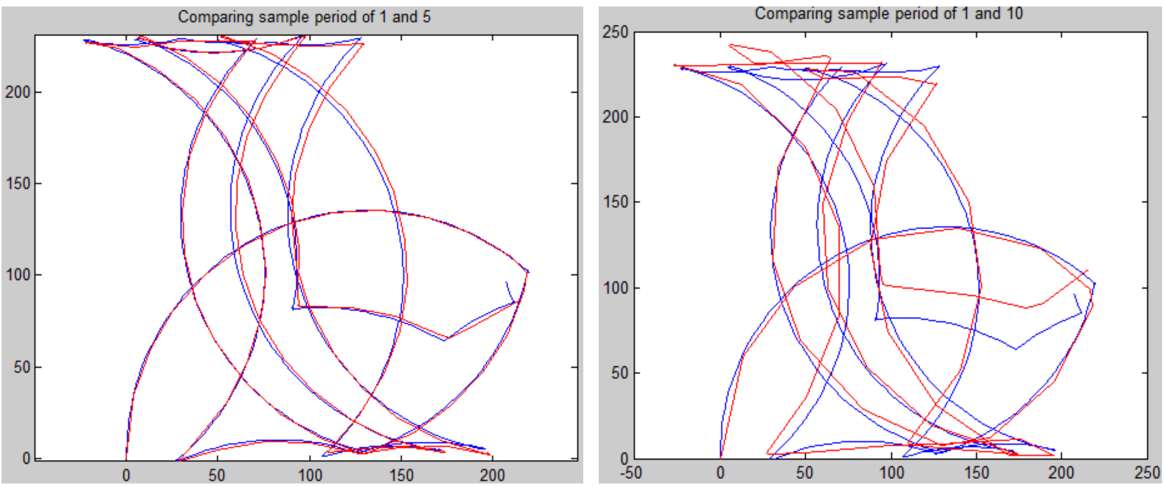

Figure 3.11 : The difference between the state variables with the sampling period of 1 (blue) and 5 (red) in

left figure and with the sampling period of 1 (blue) and 10 (red) in right figure

After turning the wheelbase to 45mm, and the wheel diameter to 14mm, the condition is not like the previous condition that the figure is proportionally scaled-down in Figure3.6. Now the new figure of the uncertainty is totally different from the original uncertainty figure, even without any similarity with the scaled-down version of the original uncertainty figure, which is shown in Figure 3.12. This is because that unlike the previous data we get, under this condition the robot is moving disorderly and unsystematic. So the parameters of the wheelbase and the wheel diameter affect a lot to the system.

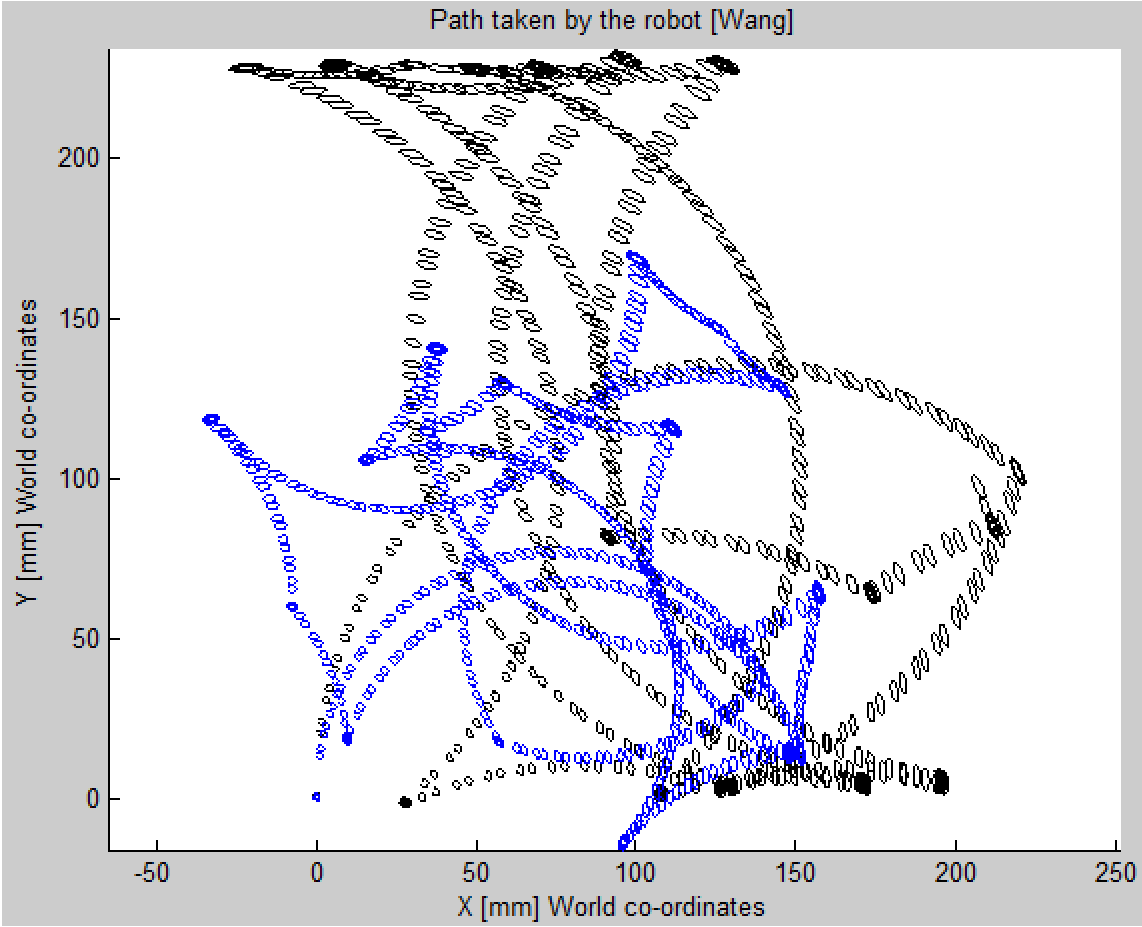

Figure 3.12: Comparing the uncertainty of original robot and of a smaller size robot, blue one stands for the smaller size robot, black one stands for the original size robot

Then we also make the wheel base as a new uncertain variable to analyze the uncertainty, which is shown in Figure 3.13. Similar with the Figure 3.7, the uncertainty of the robot is increasing all the time.

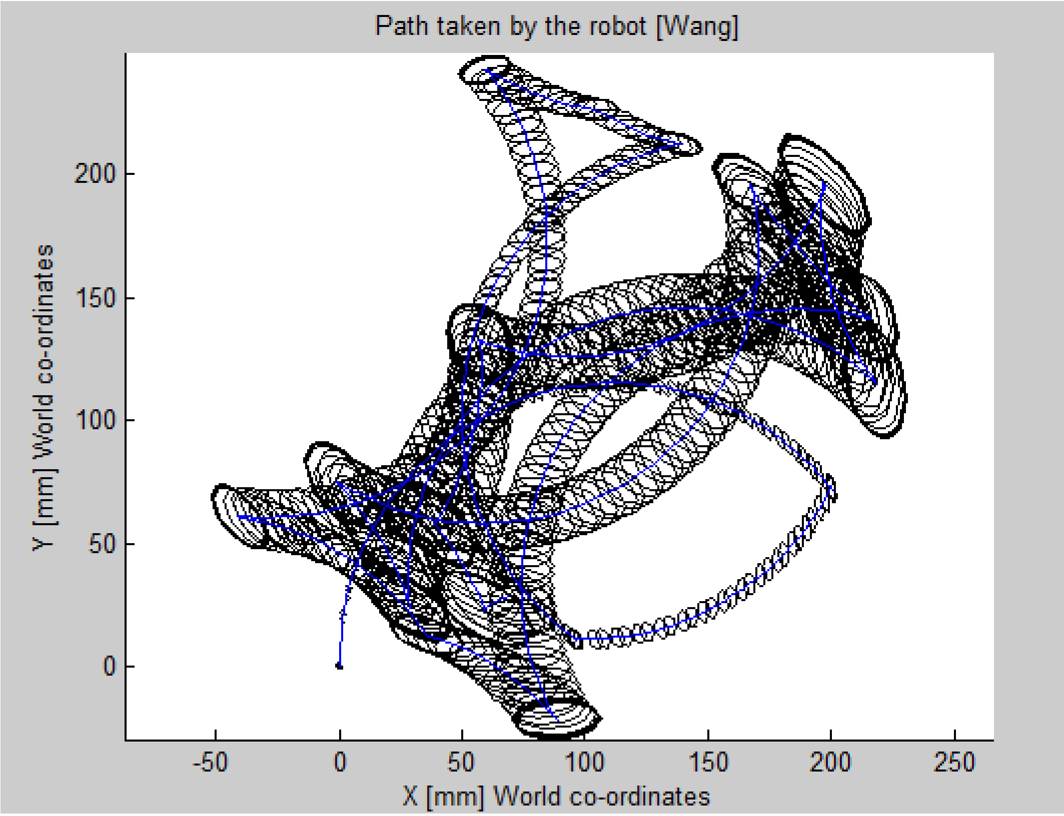

Figure 3.13: Uncertainty figure including the uncertainty of the wheel base with blue line refers to the predicted state variables

3.2 Three-wheeled vehicle (Snowhite)

(1). In this part, as we change the differential-driven vehicle to the three-wheel-driven one, we need to take new uncertainty parameters into consideration. The formulas we set up for this model is below:

$$X_k = X_{k-1} + \begin{pmatrix} \triangle d\cos(\theta_{k-1} + \frac{\triangle\theta}{2}) \\ \triangle d\sin(\theta_{k-1} + \frac{\triangle\theta}{2}) \\ \triangle\theta \end{pmatrix} = \begin{pmatrix} x_{k-1} + v\cos(\alpha)T\cos(\theta_{k-1} + \frac{v\sin(\alpha)T}{2L}) \\ x_{k-1} + v\cos(\alpha)T\sin(\theta_{k-1} + \frac{v\sin(\alpha)T}{2L}) \\ \theta_{k-1} + \frac{v\sin(\alpha)T}{2} \end{pmatrix}$$

There, α is the steering angle, v is the speed, T is the sampling time.
After we calculate the state variable recursively, we plot them in Figure 3.14:

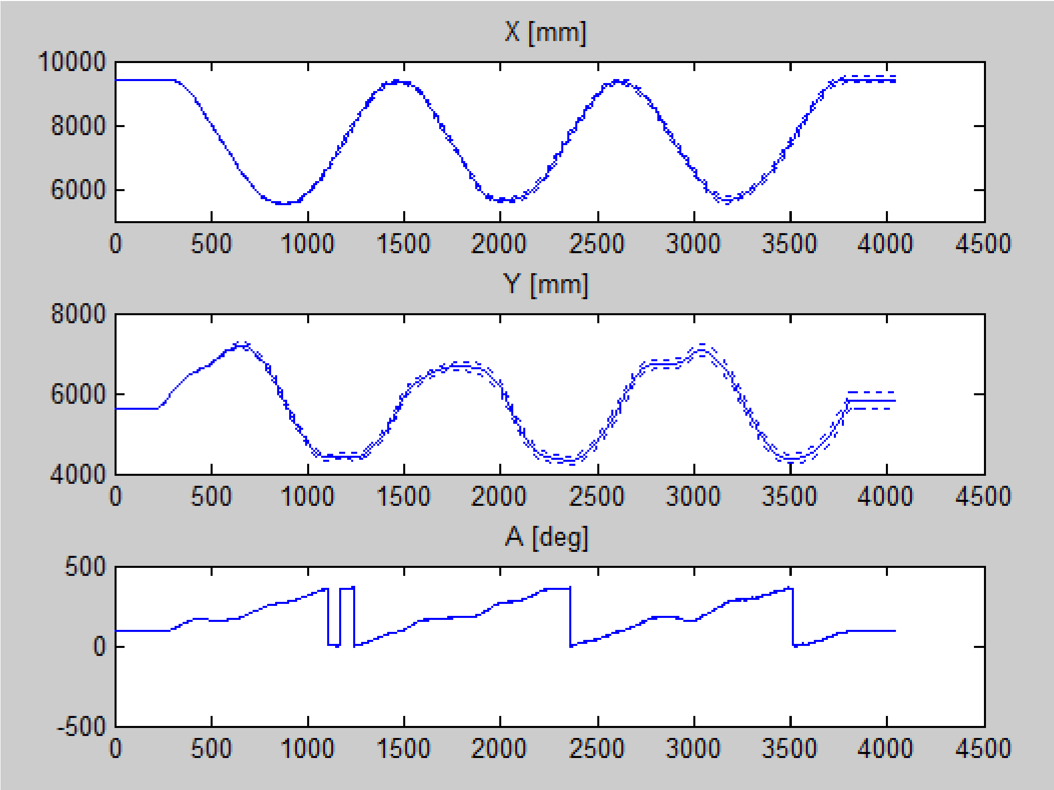

Figure 3.14: State variable x,y,θ of snowhite.

To calculate the co-variance matrix of the new positions, the law of error propagation is used.

$$\Sigma_{X_k} = J_{X_{k-1}} \Sigma_{X_{k-1}} J_{X_{k-1}}{}^T + J_{v\alpha T} \Sigma_{v\alpha T} J_{v\alpha T}{}^T$$

The Jacobian matrix is calculated just like what we calculated before, so we don't repeat it again. For initial states, we assumed the original position of the robot is (0, 0, 90*pi/180), and the initial uncertainty

$$\Sigma_{X1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & (\frac{\pi}{180})^2 \end{bmatrix}$$

For the uncertainty of the angle, speed and sampling time, we assume the co-variance of them is:

$$\Sigma_{v\alpha T} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.000001 \end{bmatrix}$$

And the co-variance of the position parameters is a 4050×90 matrix, which can be shown in the Figure 3.15:

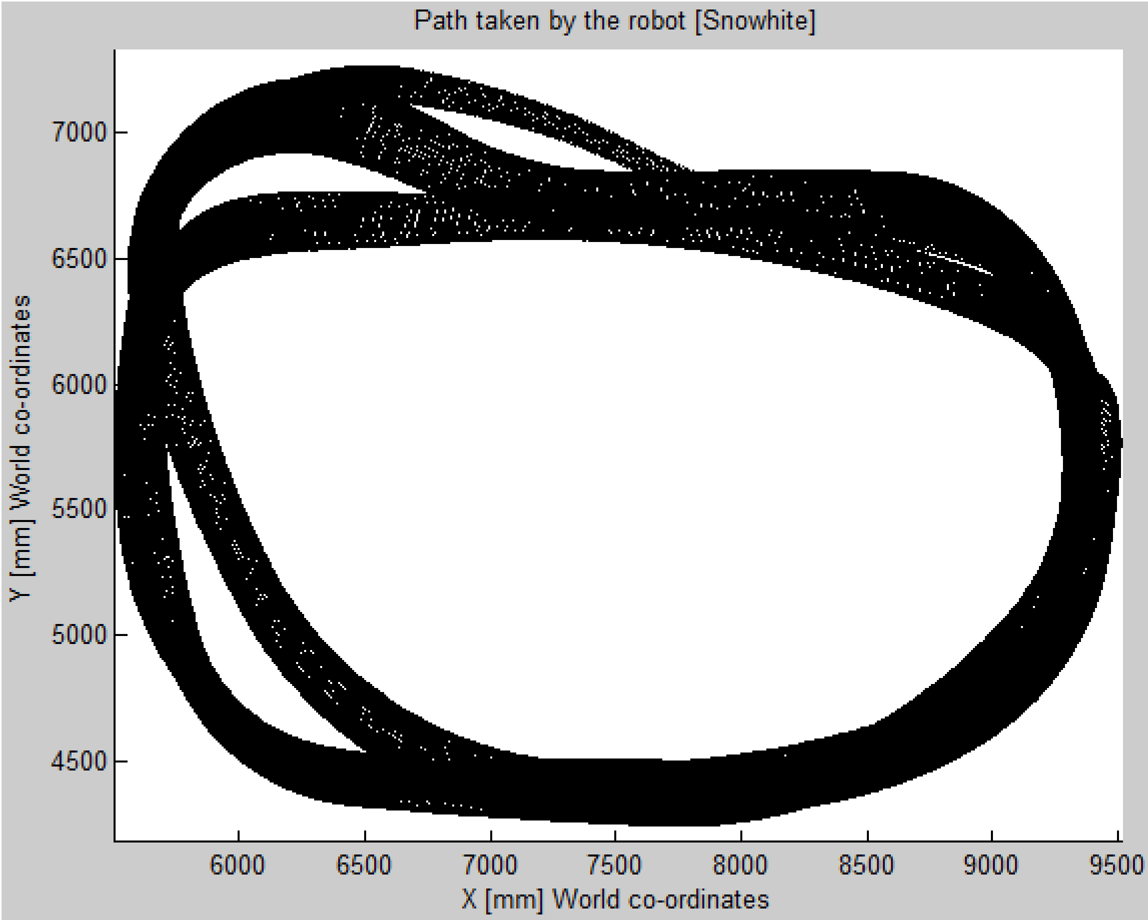

Figure 3.15: Uncertainty of each sample point of Snowhite

(2). Under our assumption of errors in the steering angle and speed, which is 10 and 0.01 respectively, we do the further analysis. In the requirements of this question, we need to compare the difference between true values and the estimated variables of the position. As we have defined the error between true variables and estimated variables, and the standard deviations in our code, we just need to plot it.

Run the whole sequence of the code, we get the error graph of figure 3.16.

As we see from the plots, we find that the two varieties of errors have the same trend in x, y, θ, which means when the errors increase, the standard deviation increase conversely, when the error decreases, the standard deviation also decreases at most time. Besides, not only the variation trend, but also the actual value of the estimated standard deviations (the uncertainty of the estimated state variables) stays close to the error of the state variables.

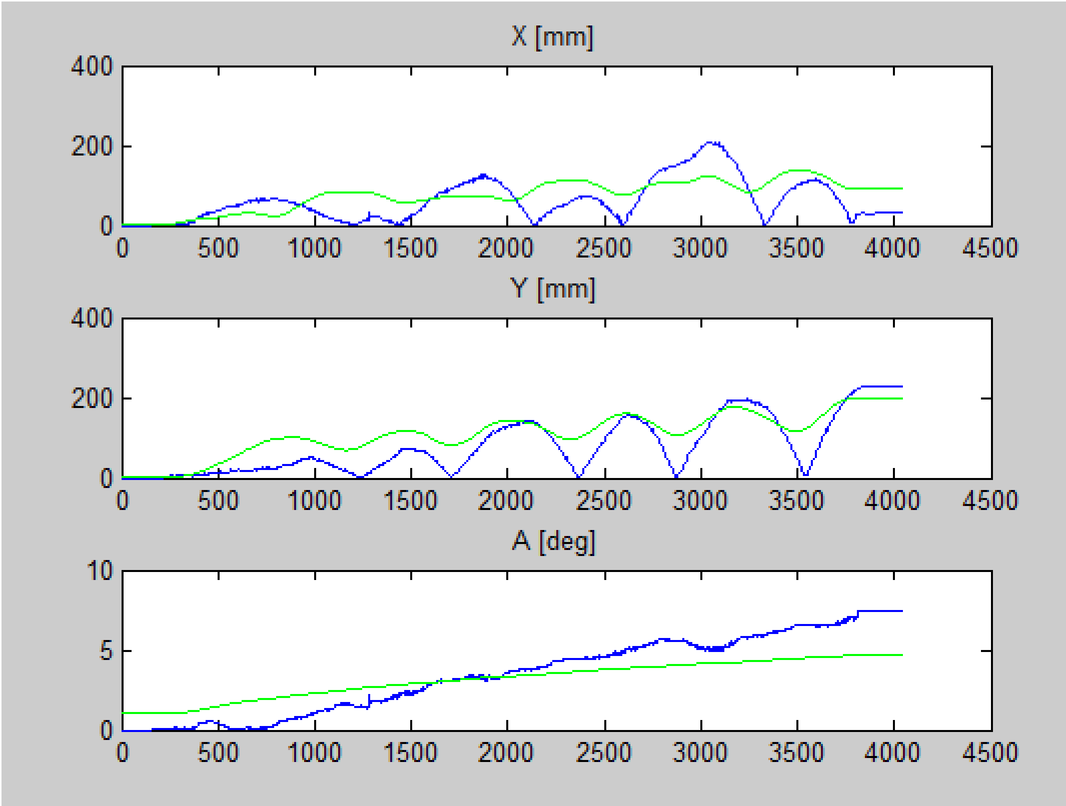

Figure 3.16: The blue polygonal lines in the plots are the errors between estimated state variables and true values separately, and the green lines are the standard deviations of x, y , θ themselves.

(3). Last we should calculate the whole distance and the time of the robot's moving process. As the data we get has 4050 samples and between each of the 2 samples is the sample time of 50ms. So the whole time of the robot's motion is:

$$(4050 - 1) \times 0.05 \ = \ 202.45 \text{ s}$$

For the distance calculation, we use the formula of the distance calculation based on the coordinate:

$$D = \sum_{i=1}^{4049} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Then we get the result that the distance passed by the robot is approximately:

$$3.1586 \times 10^4 \ mm$$

4. Conclusion

In this lab, we implement odometry and dead reckoning which are two methods often used in the robot localization, and we also discuss the effects of different species of errors on the measurement of the position parameters. In the differential drive(Khepera mini robot), as we use the line path instead of the actual arc locus, it is not error-free. With the sample time we take become longer, errors get more dramatically. Among all the parameters, errors are also caused by the wheel base and diameter's uncertainties. And these two parameters can affect the uncertainty greatly. In the three-wheel robot's model, uncertainties of speed, steering angel and sampling time are taken into account, which have influence on the survey of the position variables. For making the results more precision, we use the Jacobian matrix of different uncertainties in calculations of our analyzing model. Above all, for the estimation of the robot's position parameters, we should focus on the uncertainty errors.

Reference

[1] C. Ming Wang, Location estimation and uncertainty analysis for mobile robots.
[2] Björn Åstrand, Intelligent Vehicle – Error propagation, 2011, ver. 1.3