Exercise 2 – Odometry, Dead Reckoning and Error Predictions

Ola Bengtsson

School of Information Science, Computer and Electrical Engineering, Halmstad University, P O Box 823, Halmstad, Sweden

1 Introduction

A crucial thing for any mobile system is to know how the system responses to different inputs, e.g. how the state variables (x, y, θ) are affected by a motion of the vehicle. To fully understand this it is also very important to know how the uncertainties of these variables change due to the same motion, i.e. how the wheel motion affects the state variables. This exercise therefore focuses on odometry (encoder values that are transformed into linear motions) and dead reckoning (angles and speed that are sampled with a certain time interval, which are then integrated during the time interval which gives a linear motion). The exercise also focuses on the uncertainty caused by the latest movement and how they are transformed to the state variables.

The exercise consists of data from two robots, one with a differential drive (the Khepera mini robot) and another with a driving wheel in the front of the vehicle (the Snowhite robot, i.e. a three-wheeled robot).

1.1 Plotting the uncertainty ellipses (in two dimensions)

To plot the uncertainty ellipses (1 standard deviation) you can use the function `plot_uncertainty(…)` which is defined and works as follows:

```
function plot_uncertainty(X, C, dim1, dim2)
```

Where `X` is a 3x1 vector containing the means (i.e. the state variables (x, y, θ) of the robot), `C` is the 3x3 co-variance matrix of the state variables and `dim1` and `dim2` is the dimensions that should be plotted (typically if `X = (x, y, θ)`^{T} and you want to plot the uncertainties in x and y then `dim1 = 1` and `dim2 = 2`). The function plots the ellipse in the figure that is currently active.

2 Differential drive (Khepera mini robot)

The first dataset (which you find in the file 'khepera_circle.dat' – simply write `ENC = load('khepera.txt');` in your matlab script) is collected using the Khepera mini robot and consists of the left and right encoder values. Khepera uses incremental encoders with 600 pulses / revolution. The wheel diameter, D, is 15.3mm and the wheel base (distance between the two wheels), WB, is 53mm. See Figure 1, which shows a schematic picture of the Khepera robot. Use the matlab script 'khepera.m' and fill in the missing parts in the text, i.e. parts looking like; `%MM_PER_PULSE = ??; %`.

1) Calculate the changes in the forward direction (δd) and heading (δθ) of the robot, i.e. the changes expressed in the robot co-ordinate system caused by the latest movement. Also calculate the variances (co-variances) of δd and δθ. What uncertainties do you assume?
2) Calculate the new state variables of the robot under the assumption that the robot moves according to a circular trajectory (at this point, you could skip the compensation term given in Wang 1988). At the same time, calculate the co-variance matrix of these new positions. You can assume that the robot always starts in origin and with a heading of 90°, i.e. $(x_0, y_0, \theta_0) = (0, 0, 90*\pi/180)$.

3) Run the entire sequence of encoder but also incorporate the compensation term given in the paper Wang. Are the estimated state variables and co-variance matrices the same? If the encoder values where not read as often as they are, i.e. if they where read 2, 5 or maybe 10 times as seldom, what would then happen to the state variables? When do the state variables start to differ – and why? (Hint: Check e.g. the δd and $\delta\theta$ values).
4) How are the covariance matrices evolving during the run (plot the entire run together with the calculated co-variance matrices)? You don't have the true values of the state variables (a common problem in the mobile robot community) – but still, is the uncertainties realistic?
5) Once more, do the same run but use worse known values of the wheel diameter and the wheel base, e.g. D = 14mm and WB = 45mm. How does it affect the estimated state variables? Plot the trajectories in the same plot as the other trajectories. Can you incorporate these new uncertainties in your model, i.e. can you also include the contributions of these new uncertain parameters to the overall uncertainty in your state variables? What assumptions on these errors would you make?
6) Do the same thing again, but use the data file 'khepera.txt'.

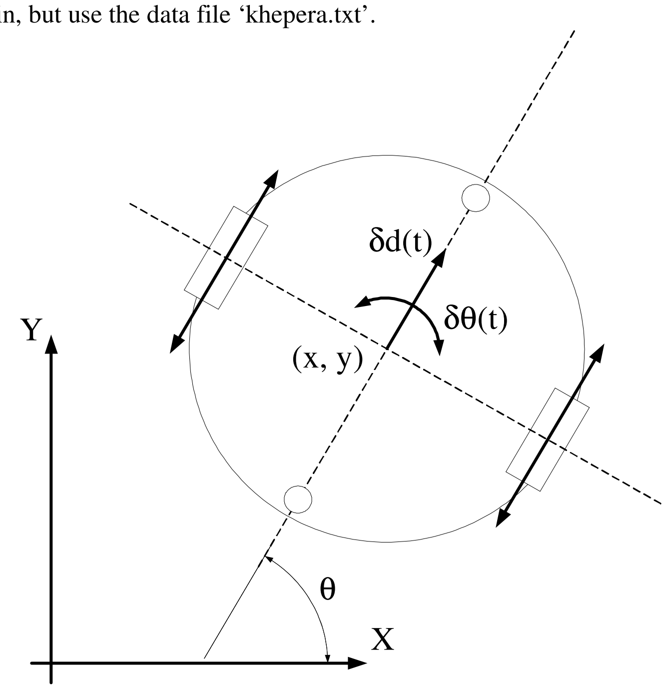

Figure 1: Khepera mini robot.

3 Three-Wheeled vehicle (Snowhite)

The second dataset (which you find in the file 'snowhite.txt' – simply write `DATA = load('snowhite.txt');` in your matlab script) is collected using the Snowhite robot and contain the speed and angle of the steering / driving wheel and the true positions in each time step (the sampling period is 50ms). The data file thus looks as follows:

$v(0)\ \ \alpha(0)\ \ True_X(0)\ \ True_Y(0)\ \ True_\theta(0)$
$v(1)\ \ \alpha(1)\ \ True_X(1)\ \ True_Y(1)\ \ True_\theta(1)$
….
$v(n)\ \ \alpha(n)\ \ True_X(n)\ \ True_Y(n)\ \ True_\theta(n)$

The wheel base, L, (distance between the front and rear wheel axis) is 680mm. See Figure 2, which shows a schematic picture of the Snowhite robot. From the velocities {v(0), v(1), …, v(n)} it is possible to calculate the change in the robot's forward and rotational displacements (multiplying the velocities with the sample period, T, gives the displacements), which become:

$$\delta d(t) = v(t)\cos(\alpha(t))T \tag{3.1}$$

$$\delta\theta(t) = \frac{v(t)\sin(\alpha(t))T}{L} \tag{3.2}$$

1) Calculate the new state variables of the robot. Also calculate the co-variance matrix of these new positions. (To do this, you have to derive the Jacobian matrices w.r.t. the uncertain parameters. Follow either the example given in the Wang paper or in chapter 5 in the text book.) The robot always starts in the first true position and with the first true heading, i.e. $(x_0, y_0, \theta_0) = (True_x(0), True_Y(0), True_\theta(0))$.
2) Run the entire sequence of speeds and steering angles and compare the estimated state variables to the true values. Also compare the error in the state variable estimates, i.e. the difference between the estimated state variable and the true state variables, to the standard deviations of the estimated variances (square root of the diagonal elements in the co-variance matrix). What errors do you assume in the steering angle and in the speed? Make sure the estimated standard deviations (the uncertainty of the estimated state variables) stay close to the error of the state variables.
3) Don't forget to plot the error ellipses along the path taken by the robot. How long (distance and time) is the path?

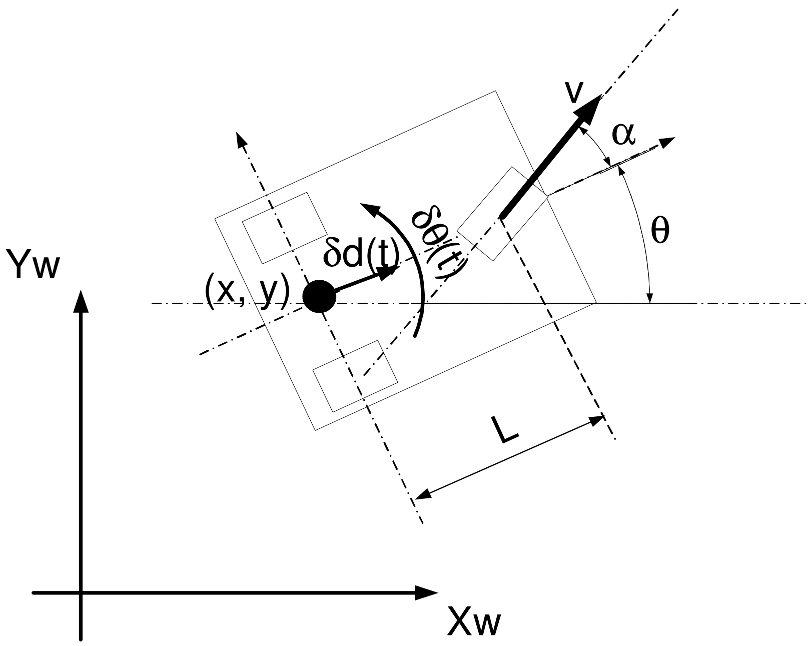

Figure 2: Schematic model of the Snowhite robot. The robot has a steering / driving wheel in the front and two passive fixed wheels in the rear.

4 Others

The reports (which are individual, although you are more than welcome to solve the exercises in smaller groups, e.g. by groups of two students) should contain all necessary equations with all assumptions motivated. You should also interpretive your results, i.e. handing in a plot without any explanations is not enough.