Sheet 2
Motion Models and Robot Odometry

Group 4
Urs Borrmann, Caner Hazirbas, FangYi Zhi

May 12, 2013

Exercise 1

c) Group Picture

d) Graph Visualization

e) Specify the odometry model

Exercise 2

d) Kalman Filter Covariance Ellipse Screenshot

Figure 1 is the screenshot of the covariance ellipse from the original Q matrix visualized by rviz.[1]

Figure 2 shows the screenshot of the estimated two-dimensional trajectory from the given bag file.

e) Kalman Filter with Higher Noise Screenshots

Figure 3 shows the screenshot of the covariance ellipse from the modified Q matrix, which drifts two times more in the global x-direction.

Figure 4 shows the screenshot of the estimated two-dimensional trajectory from the bag file with modified Q matrix.[2]

[1] In order to easily compare the difference between the effect of the two diffenrent Q matrices, we took both screenshots at the last moment of the given bagfile.

[2] There is no difference of the trajectories between the two different Q matrix, since until now, the Q matrix just adjusts the covariance ellipse but not the state vector, hence the trajectory.

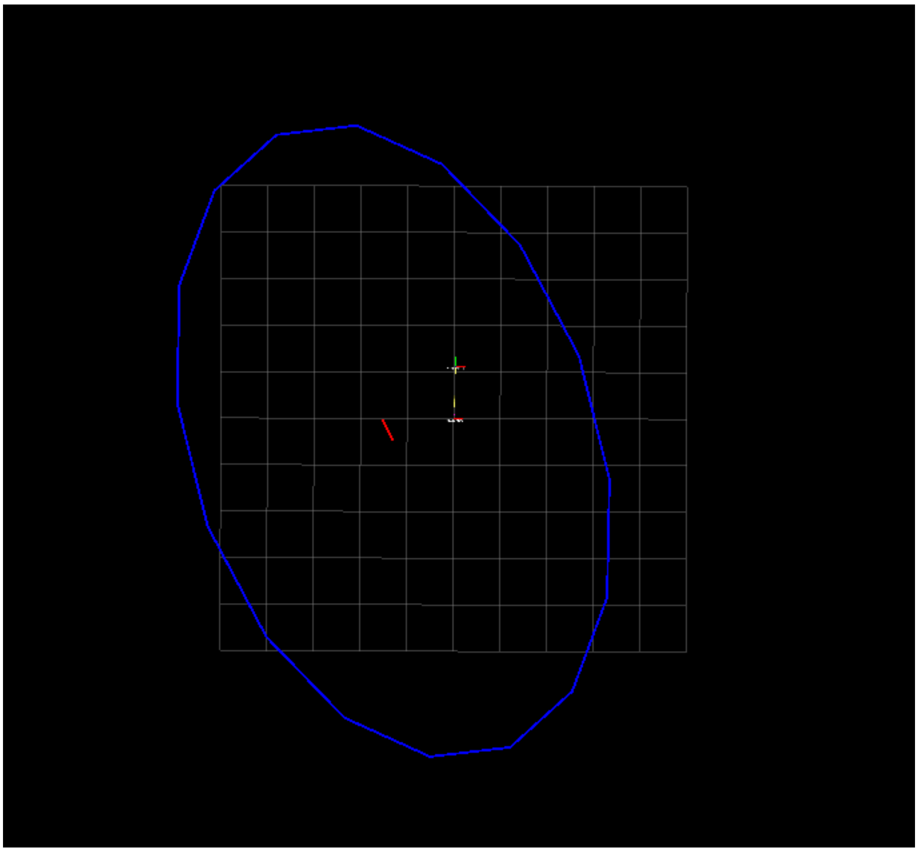

Figure 1: Covariance ellipse with the origin Q matrix

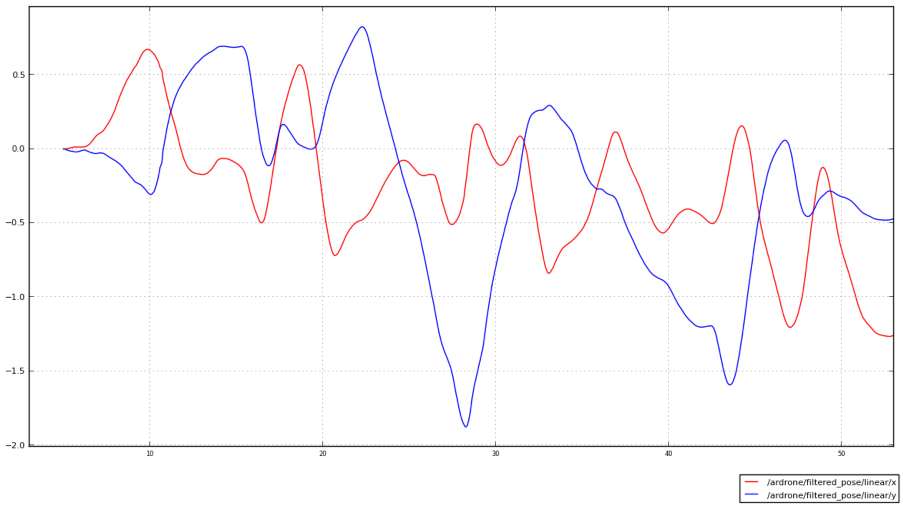

Figure 2: Estimated two-dimensional trajectory from the given bag file

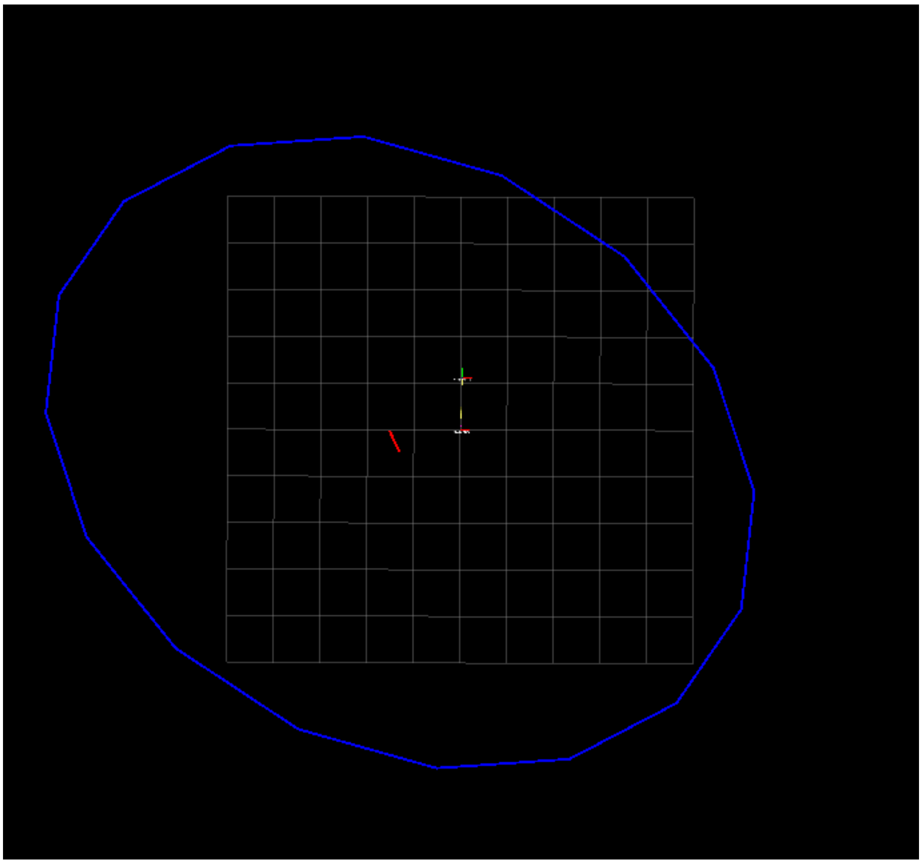

Figure 3: Covariance ellipse with the modified Q matrix

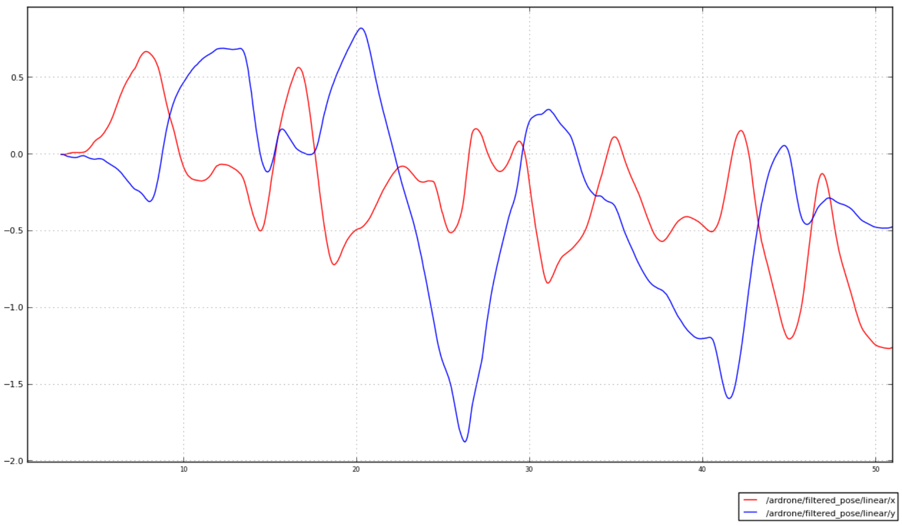

Figure 4: Estimated two-dimensional trajectory from the bag file with modified Q matrix.

f) Noise Prediction for Experimental Setup

g) Observation Function and Its Jacobian

Observation Function

Observed marker pose is calculated with function $h(x)$ (eq. 1). This $h(x)$ observation function predicts the marker pose z_{pre} given x, estimated robot world state(eq 2), and z_g ,the marker pose in global frame(eq. 3).

1. $$\overrightarrow{z_{pre}} = h(\vec{x})$$
   $$\overrightarrow{z_{pre}} = (x_{pre} \quad y_{pre} \quad \psi_{pre})^T$$
2. $$\vec{x} = (x_w \quad y_w \quad \psi_w)^T$$
3. $$\vec{z_g} = (x_g \quad y_g \quad \psi_g)^T$$

In order to find the observation, we need to transform the global marker pose to local frame.

If X is homogeneous transformation matrix of x, robot pose,

$$X = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\psi_w & -\sin\psi_w & x_w \\ \sin\psi_w & \cos\psi_w & y_w \\ 0 & 0 & 1 \end{pmatrix}$$

then we can transform any local frame to global frame as follows;

$$\vec{t_g} = X\overrightarrow{t_{pre}}$$

We want to transform from global to local. In order to do that we should take inverse of X transformation matrix;

$$X^{-1} = \begin{pmatrix} R^{-1} & -R^{-1}t \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\psi_w & \sin\psi_w & -x_w\cos\psi_w - y_w\sin\psi_w \\ -\sin\psi_w & \cos\psi_w & x_w\sin\psi_w - y_w\cos\psi_w \\ 0 & 0 & 1 \end{pmatrix}$$

Now we can compute the local marker position from global marker position;

$$\vec{t_g} = \begin{pmatrix} x_g \\ y_g \\ 1 \end{pmatrix}$$

$$\tilde{t_{pre}} = X^{-1}\tilde{t_g} = \begin{pmatrix} (x_g - x_w)\cos\psi_w + (y_g - y_w)\sin\psi_w \\ -(x_g - x_w)\sin\psi_w + (y_g - y_w)\cos\psi_w \\ 1 \end{pmatrix}$$

Since yaw angle is always in the global frame, predicted yaw angle is

$$\psi_{pre} = (\psi_g - \psi_w)$$

At the end we get the following observation function

$$h(\vec{x}) = \begin{pmatrix} (x_g - x_w)\cos\psi_w + (y_g - y_w)\sin\psi_w \\ -(x_g - x_w)\sin\psi_w + (y_g - y_w)\cos\psi_w \\ (\psi_g - \psi_w) \end{pmatrix}$$

Its derivative (Jacobian) can be then calculated as:

$$H(\vec{x}) = \frac{\partial h(\vec{x})}{\partial \vec{x}} = \begin{pmatrix} -\cos\psi_w & -\sin\psi_w & -(x_g - x_w)\sin\psi_w + (y_g - y_w)\cos\psi_w \\ \sin\psi_w & -\cos\psi_w & -(x_g - x_w)\cos\psi_w - (y_g - y_w)\sin\psi_w \\ 0 & 0 & -1 \end{pmatrix}$$

i) Trajectory

Figure 5 shows the screenshot of the from EKF corrected two-dimensional trajectory from the given bag file.

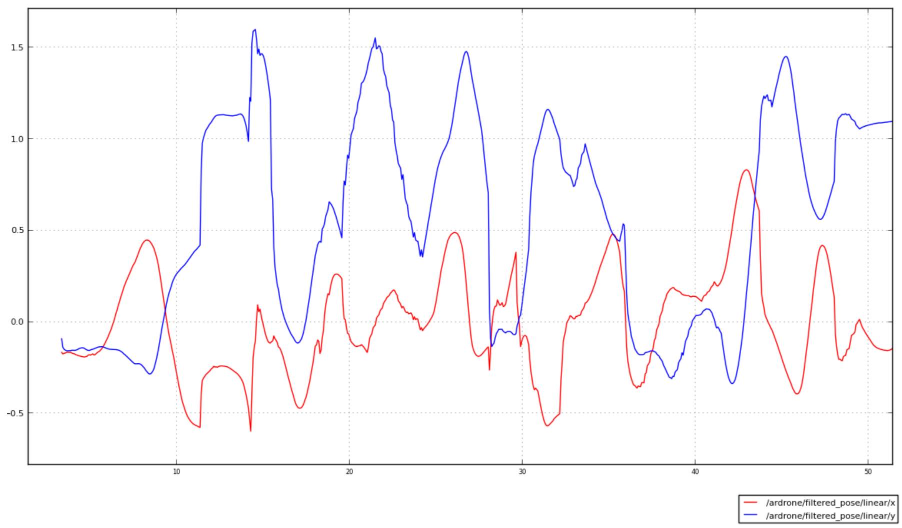

Figure 5: From the EKF corrected two-dimensional trajectory from the given bag file

j) Drift on Pose Estimation