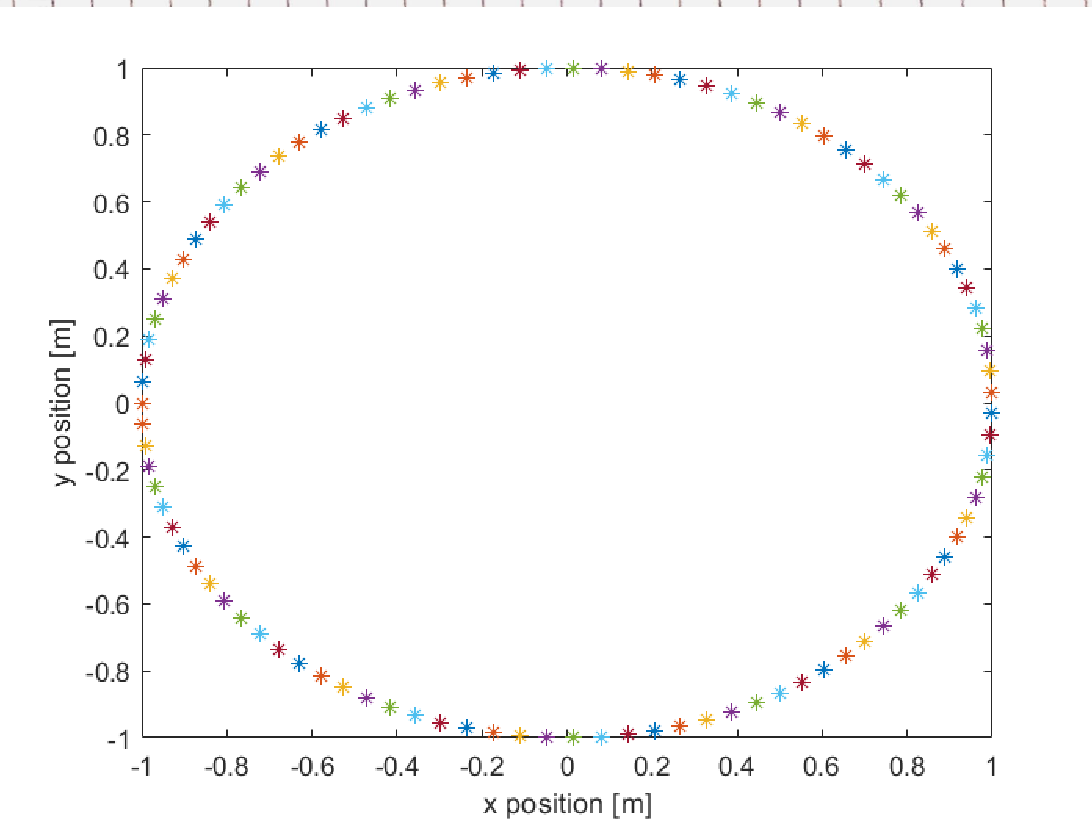
Kerim MORAL
518191043

a) Robot's heading can change between -pi and pi. According to those angles, robot position has been plotted for 100 steps.

```
%% Initialize states
d = 1;
N=100;
tet = linspace(-pi,pi,N);                    %teta variable between 0 and 1
x =  d*cos(tet);                          %x variable between 0 and 1
y = d*sin(tet);                       %y variable between 0 and 1
xt = [x; y; tet];                     %state matrix
%% Plot the positions
for i=1:N
    plot(x(i),y(i),'*');
    hold on;
end
xlabel('x position [m]'); ylabel('y position [m]');
```



b) $$x_t = \underbrace{\begin{bmatrix} x_{t-1} + \cos\theta_{t-1} \\ y_{t-1} + \sin\theta_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{g(x_{t-1})} + \varepsilon$$

Linearizing;

$$\frac{\partial g(x_{t-1})}{\partial x_{t-1}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \frac{\partial g(x_{t-1})}{\partial y_{t-1}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\frac{\partial g(x_{t-1})}{\partial \theta_{t-1}} = \begin{bmatrix} -\sin\theta_{t-1} \\ \cos\theta_{t-1} \\ 1 \end{bmatrix}$$

$$\frac{\partial g(x_{t-1})}{\partial x_{t-1}} = \boxed{G_t = \begin{bmatrix} 1 & 0 & -\sin\theta_{t-1} \\ 0 & 1 & \cos\theta_{t-1} \\ 0 & 0 & 1 \end{bmatrix}}$$

$$g(x_{t-1}) \approx g(\mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) + \varepsilon$$

So Gaussian is:

$$p(x_t \mid x_{t-1}) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[x_t - g(\mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})\right]^T \Sigma^{-1} \left[x_t - g(\mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})\right]\right\}$$

CamScanner ile tarandı

c) For calculation, $t=1$, $t-1=0$,

$$g(\mu_{t-1}) = g(\mu_0) = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad G_t(\mu_0) = \begin{bmatrix}1&0&0\\0&1&1\\0&0&1\end{bmatrix}$$

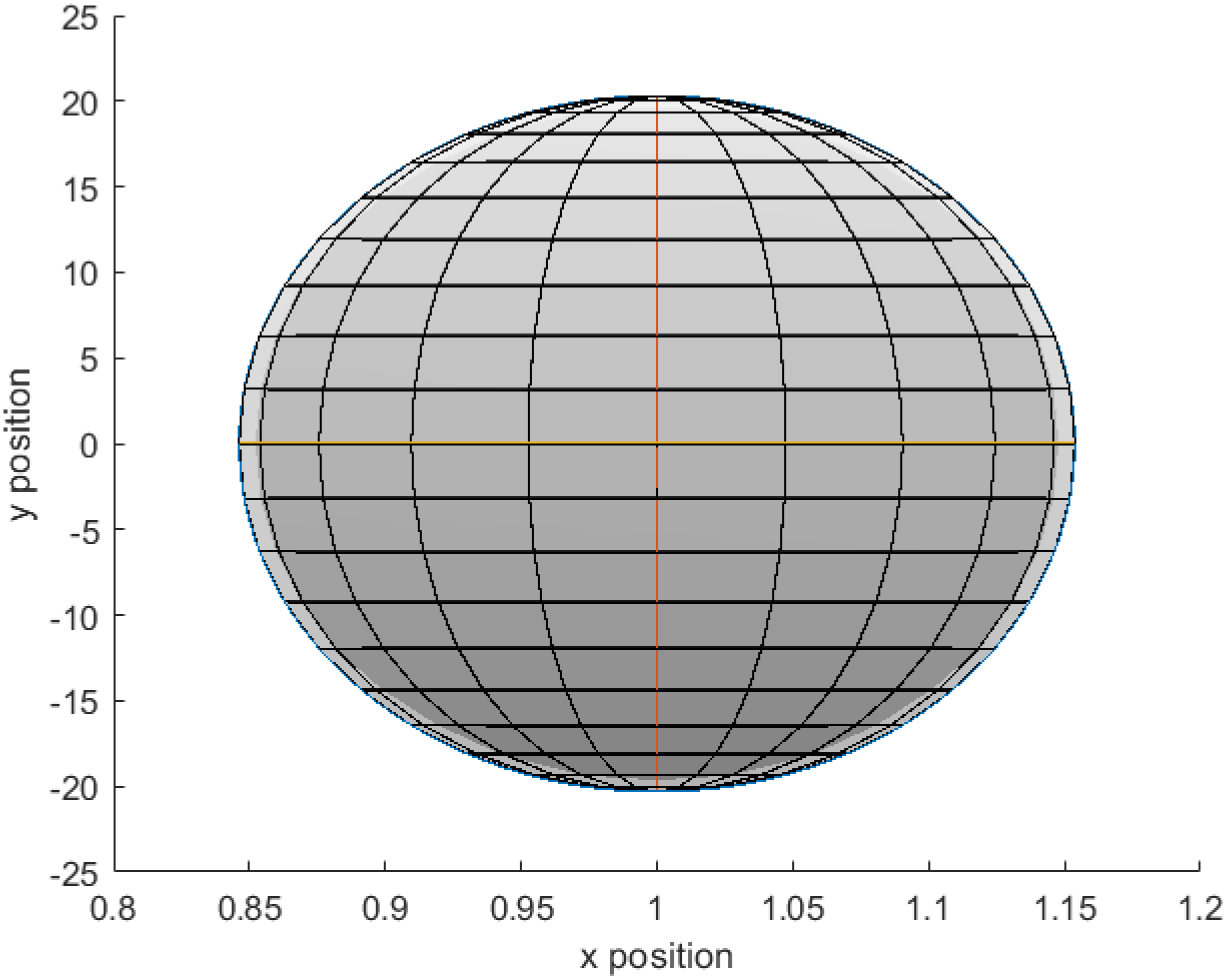
$(x_{t-1}-\mu_{t-1}) = 0$, so $\bar{\mu}_1 = g(\mu_{t-1}) = \begin{bmatrix}1\\0\\0\end{bmatrix}$

$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$, since there is no movement noise at $t=1$, $R_1 = 0$

$\bar{\Sigma}_1 = G_1 \Sigma_0 G_1^T$ so prediction Gaussian:

$$p(x_t) = \det(2\pi\bar{\Sigma}_1)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[(x_t-\bar{\mu}_1)\right]^T \bar{\Sigma}_1^{-1}\left[x_t-\bar{\mu}_1\right]\right\}$$

```matlab
%% Initialize states
d = 1;
N=1000;
tet = linspace(-pi,pi,N);                    %teta variable between 0 and 1
x = d*cos(tet);                             %x variable between 0 and 1
y = d*sin(tet);                                %y variable between 0 and 1
xt = [x; y; tet];                            %state matrix
E0 = [0.01 0 0; 0 0.01 0; 0 0 10000*pi/180]; %Sigma matrix
Mu1_ = [1; 0; 0;];                          %t=1 mean values
G1 = [1 0 0; 0 1 1; 0 0 1];                 %       %Jacobian for Mu0
E1_ = G1*E0*G1.';
px = zeros(1,N);
%% Draw Uncertianty Ellipse
error_ellipse(E1_,'mu',Mu1_)
xlabel('x position'); ylabel('y position'); zlabel('teta');
clear all; clc
```



25
20
15
10
5
0
-5
-10
-15
-20
-25
y position
0.8
0.85
0.9
0.95
1
1.05
1.1
1.15
1.2
x position

d) Measurement model $h(x_t)$:

$$h(x_t) = \underbrace{[1\ 0\ 0]}_{C} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$ which is linear.

so $H_t = C$.

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$$

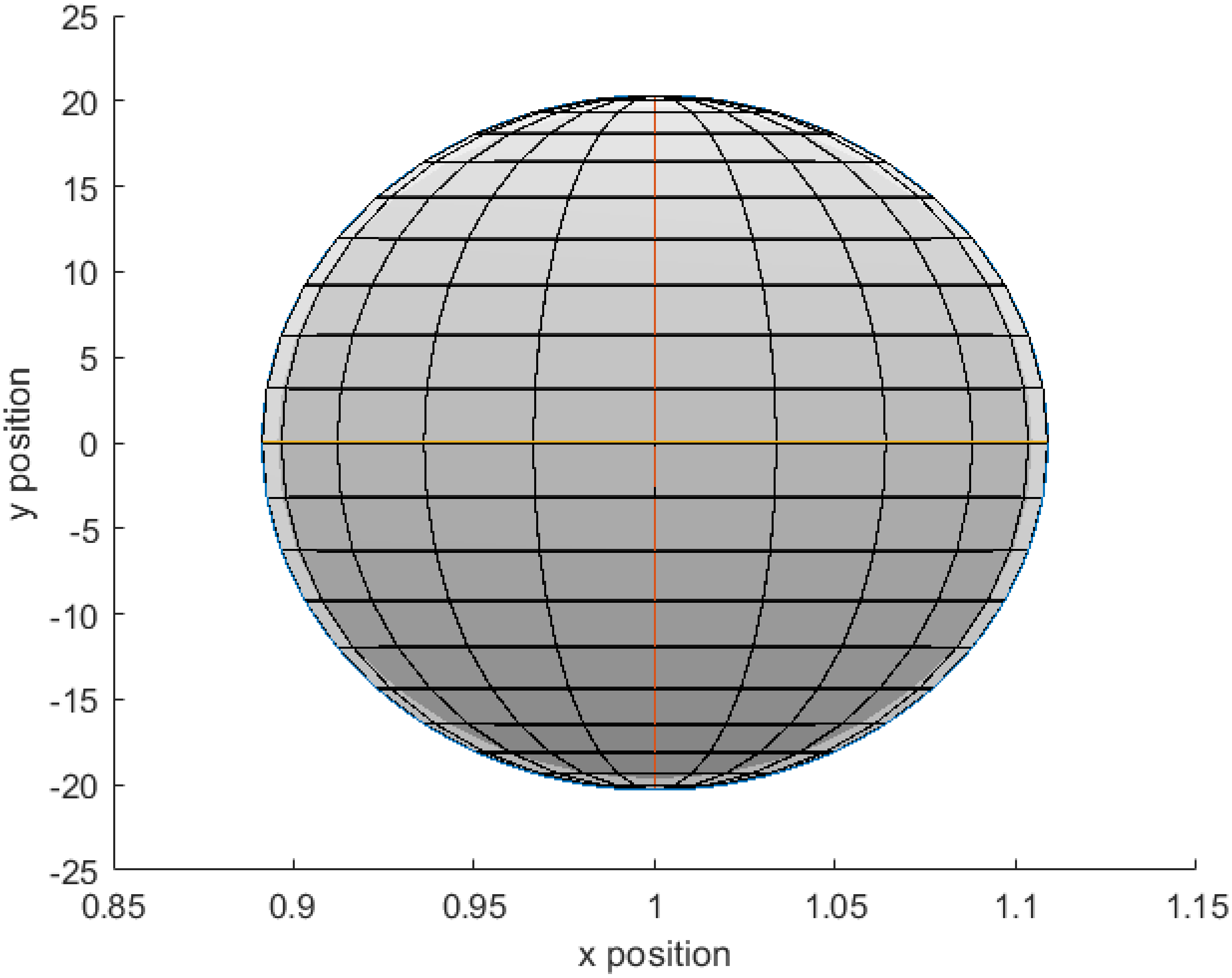
$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

Covariance in x position of the robot is very low.

Hence measurement z_t has been taken as $z_t = \bar{\mu}_1(1) = 1$

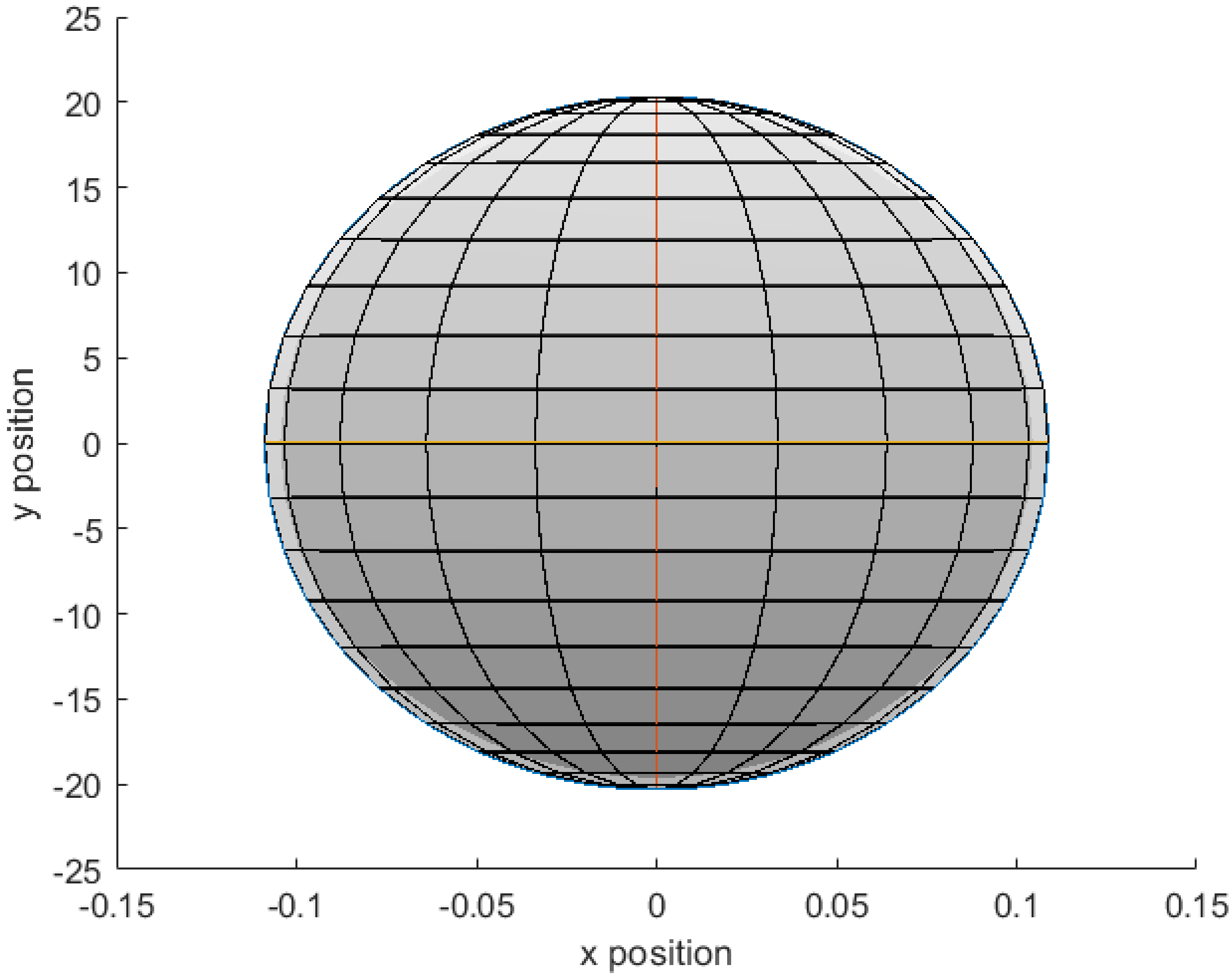
```
%% Initialize states
dist = 1;
N=1000;
tet = linspace(-pi,pi,N);                    %teta variable between 0 and 1
x = dist*cos(tet);                           %x variable between 0 and 1
y = dist*sin(tet);                           %y variable between 0 and 1
xt = [x; y; tet];                            %state matrix
E0 = [0.01 0 0; 0 0.01 0; 0 0 10000*pi/180]; %Sigma matrix
Mu1_ = [1; 0; 0;];                           %t=1 mean values
G1 = [1 0 0; 0 1 1; 0 0 1];                  %Jacobian for Mu0
E1_ = G1*E0*G1.';
H1 = [1 0 0];                                %Measurement Jacobian
Q1 = 0.01;                                   %Measurement noise covariance
zt = Mu1_(1);
I = eye( 3 );
%% Calculate Kalman gain and posteriors
K1 = E1_*H1.'*inv(H1*E1_*H1.'+ Q1);
Mu1 = Mu1_ + K1 * (zt - 1);
E1 = (I - K1*H1) * E1_;
%% Plot
error_ellipse(E1,'mu',Mu1)
xlabel('x position'); ylabel('y position'); zlabel('teta');
clear all; clc
```



25
20
15
10
5
0
-5
-10
-15
-20
-25
y position
0.85
0.9
0.95
1
1.05
1.1
1.15
x position

For intuitive check, ~~xt~~ $z_t = -1$ has been taken. This measurement is possible for a zero centered 1 radius circle

```matlab
%% Initialize states
dist = 1;
N=1000;
tet = linspace(-pi,pi,N);                      %teta variable between 0 and 1
x = dist*cos(tet);                             %x variable between 0 and 1
y = dist*sin(tet);                             %y variable between 0 and 1
xt = [x; y; tet];                              %state matrix
E0 = [0.01 0 0; 0 0.01 0; 0 0 10000*pi/180];  %Sigma matrix
Mu1_ = [1; 0; 0;];                             %t=1 mean values
G1 = [1 0 0; 0 1 1; 0 0 1];                    %Jacobian for Mu0
E1_ = G1*E0*G1.';
H1 = [1 0 0];                                  %Measurement Jacobian
Q1 = 0.01;                                     %Measurement noise covariance
zt = -1;
I = eye( 3 );
%% Calculate Kalman gain and posteriors
K1 = E1_*H1.'*inv(H1*E1_*H1.'+ Q1);
Mu1 = Mu1_ + K1 * (zt - 1);
E1 = (I - K1*H1) * E1_;
%% Plot
error_ellipse(E1,'mu',Mu1)
xlabel('x position'); ylabel('y position'); zlabel('teta');
clear all; clc
```



As can be seen from above, even the mesaurement is zt=-1 the EKF algorithm produces likelyhood x position as 0, which doesnt add up with measurement since measurement is almost the same with xt because of low covariance Q.

e) Intuitive posterior was a circle with radius of 1 and center as 0. So it was possible for robot to move anywhere in x[-1,1] and y[-1,1]. But Gaussian estimate shows that x position is likely to be around 1 meter with high certainty while y is not certain (y position can be anywhere between [-20,20]. If theta covariance was small, the uncertainty in y would become much more smaller hence more certain.

If initial oriantation was known but now the y position, we would start computing with some random y position and after a few computation of EKF the median of y would become much closer to the real value.