

# ME780 Assignment 1

Stan Brown & Chris Choi

## 1 Derivation of the Omni-directional Robot

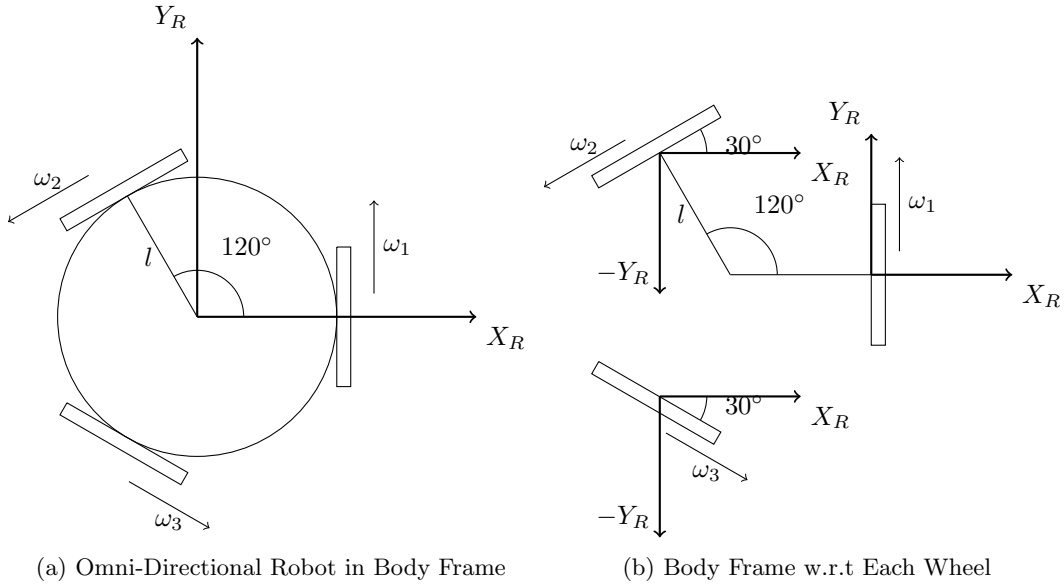


Figure 1: Omni-Directional Robot

Here we derive the kinematics of the Omni-directional robot. We followed the generalized approach in [?] in resolving how each wheel contributes to the robot chassis. The generalized equation is:

$$\xi_I = R(\theta)^{-1} J_1^{-1} J_2 \omega \quad (1)$$

Where:

- $\xi_I$ : State in the body frame
- $R(\theta)$ : Rotation matrix
- $J_1$ : Wheel constraints
- $J_2$ : Diagonal matrix denoting the radii  $r$  of each wheel
- $\omega$ : Angular velocity  $\omega$  of each wheel

Deriving the robot kinematics for each wheel we get in the form of 1 we obtain:

$$J_1 = \begin{bmatrix} 0 & 1 & l \\ -\cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \\ \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \end{bmatrix} \begin{matrix} \} Wheel1 \\ \} Wheel2 \\ \} Wheel3 \end{matrix} \quad (2)$$

$$J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \quad (3)$$

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (4)$$

Putting it all together we have:

$$\xi_I = R(\theta)^{-1} J_1^{-1} J_2 \omega \quad (5)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 & l \\ -\cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \\ \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \end{bmatrix}^{-1} \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3l} & \frac{1}{3l} & \frac{1}{3l} \end{bmatrix} \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \sin(\theta) & -\frac{1}{\sqrt{3}} \cos(\theta) - \frac{1}{3} \sin(\theta) & \frac{1}{\sqrt{3}} \cos(\theta) - \frac{1}{3} \sin(\theta) \\ \frac{2}{3} \cos(\theta) & \frac{1}{\sqrt{3}} \sin(\theta) - \frac{1}{3} \cos(\theta) & -\frac{1}{\sqrt{3}} \sin(\theta) - \frac{1}{3} \cos(\theta) \\ \frac{1}{3l} & \frac{1}{3l} & \frac{1}{3l} \end{bmatrix} \begin{bmatrix} r\omega_1 \\ r\omega_2 \\ r\omega_3 \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \sin(\theta)r\omega_1 - \frac{1}{\sqrt{3}} \cos(\theta)r\omega_2 - \frac{1}{3} \sin(\theta)r\omega_2 + \frac{1}{\sqrt{3}} \cos(\theta)r\omega_3 - \frac{1}{3} \sin(\theta)r\omega_3 \\ \frac{2}{3} \cos(\theta)r\omega_1 + \frac{1}{\sqrt{3}} \sin(\theta)r\omega_2 - \frac{1}{3} \cos(\theta)r\omega_2 - \frac{1}{\sqrt{3}} \sin(\theta)r\omega_3 - \frac{1}{3} \cos(\theta)r\omega_3 \\ \frac{1}{3l}r\omega_1 + \frac{1}{3l}r\omega_2 + \frac{1}{3l}r\omega_3 \end{bmatrix} \quad (9)$$

$$(10)$$

$$\dot{g}(\theta, \omega) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (11)$$

Where:

- $\dot{g}$  is our velocity motion model of the omni-directional robot
- $\theta$  is the angle (or bearing) of the robot in the inertia frame.
- $\omega$  is the wheel inputs of the robot  $\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$

## 2 Applying the Derived Robot Model

### 2.1 Predefined Rotation Inputs

Applying the following rotation inputs to the wheel for 15 seconds:

- $\omega_1 = -15.5 \text{ rads}^{-1}$
- $\omega_2 = 10.5 \text{ rads}^{-1}$
- $\omega_3 = 1.5 \text{ rads}^{-1}$

We obtain the following motion.

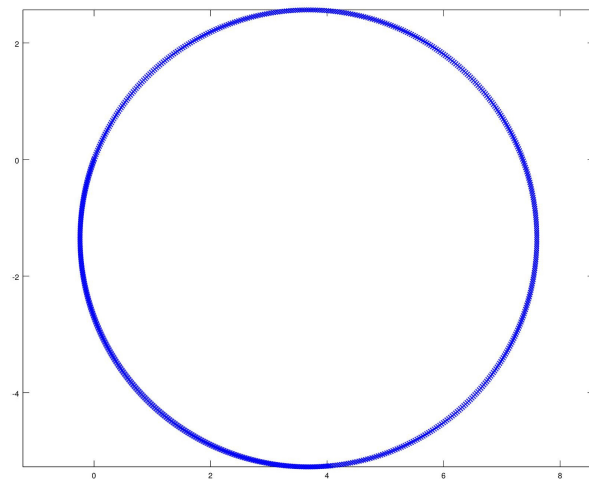


Figure 2: Traverse with inputs:  $\omega_1 = -15.5 \text{ rads}^{-1}$   $\omega_2 = 10.5 \text{ rads}^{-1}$   $\omega_3 = 1.5 \text{ rads}^{-1}$

## 2.2 Traversing East, West, North East, South East and Inplace

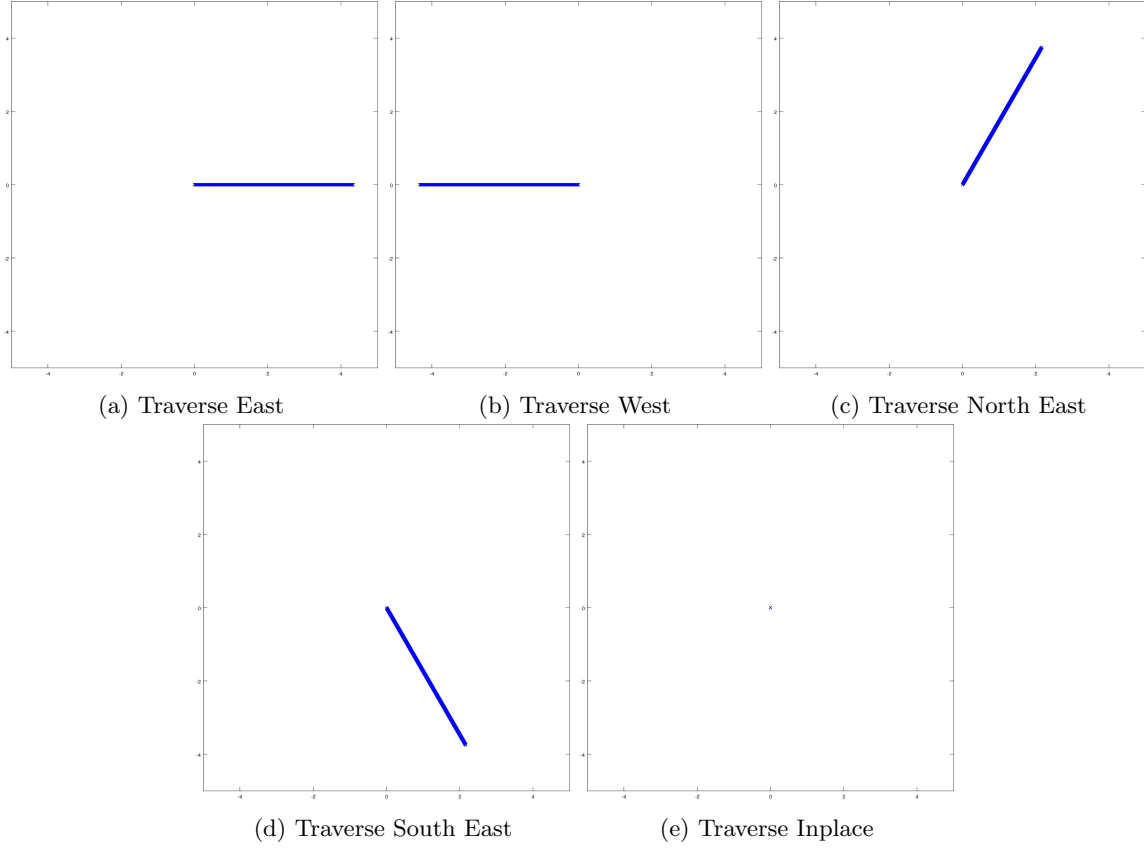


Figure 3: Traversing East, West, North East, South East and Inplace

The inputs that drove the robot to traverse in the directions shown in Fig 3 for 15 seconds are as follows:

- East:  $\omega_1 = 0, \omega_2 = -1, \omega_3 = 1$
- West:  $\omega_1 = 0, \omega_2 = 1, \omega_3 = -1$
- North East:  $\omega_1 = 1, \omega_2 = -1, \omega_3 = 0$
- South East:  $\omega_1 = -1, \omega_2 = 0, \omega_3 = 1$
- Inplace:  $\omega_1 = 1, \omega_2 = 1, \omega_3 = 1$

### 2.3 Moving in a Circle

To traverse in a circle of diameter of 2m, we had to be a little creative with the rotation inputs of the wheels. To do this we applied the following intuition:

1. Constrained the robot's  $y$ -axis movement to 0, and so  $\dot{y}$  is  $\dot{y} = 0$ . Use wheels 2 and 3 as fixed wheels pushing the robot forward, and use wheel 1 as a steering wheel. (See fig 1)
2. For the robot to traverse a circle, the robot must traverse the circumference of a circle, i.e.  $2\pi R$ , where  $R$  is the radius of the circle we want the robot to make  $d = 2\pi R$ . Converting that to velocity, the equation becomes  $\dot{x} = \frac{2\pi R}{t}$
3. For the robot to make a full circle, it must change angle  $\theta = 2\pi$ . Converting this to the angular rate the equation becomes  $\dot{\theta} = \frac{2\pi}{t}$

Putting the above all together we have the following, originally we had:

$$\xi_I = R(\theta)^{-1} J_1^{-1} J_2 \omega \quad (12)$$

Where we are trying to figure out what the  $\omega$  inputs are to drive the robot to make a full circle in the inertial frame, we already know what  $R(\theta)$ ,  $J_1$ ,  $J_2$  are, the only unknown left is  $\xi_I$ , but using the above intuition we can relate what it should be to make a circle.

$$\omega = \xi_I R(\theta) J_1 J_2^{-1} \quad (13)$$

$$\xi_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{2\pi R}{t} \\ 0 \\ \frac{2\pi}{t} \end{bmatrix} \quad (14)$$

Using the above equation we found the  $\omega$  inputs required to traverse a 2m diameter circle in 15 seconds is  $\omega_1 = 0.50265$ ,  $\omega_2 = -0.94838$   $\omega_3 = 1.95369$  starting from the origin  $(0,0)$ .

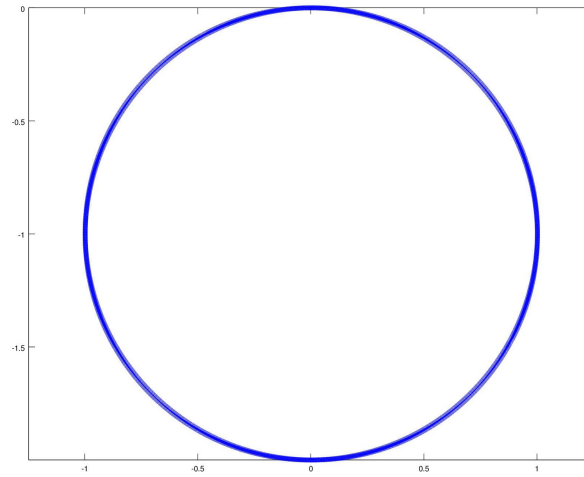


Figure 4: Omni-Robot Traversing a Circle

## 2.4 Traversing in a Spiral

For traversing in a spiral we used a similar approach as traversing in a circle. We constrained the robot so that it was only traversing in the  $x$  direction in the robot's body frame, and set the speed of wheel 1 to have initial angular velocity  $\omega_1 = 10$ , with its angular velocity set to decay at a rate of 0.999 per  $dt = 0.1$ . Below is a figure of the robot traversing in a spiral pattern for 50 seconds.

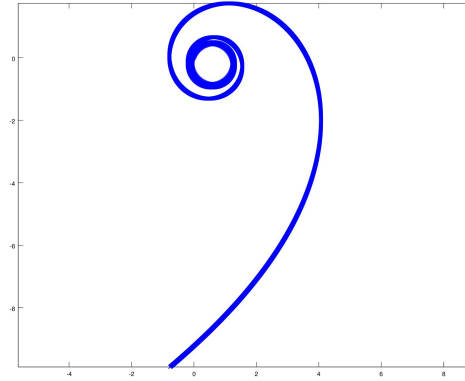


Figure 5: Omni-Robot Traversing a Spiral

### 3 Measurement Model

Lets suppose the robot has a GPS and magnometer sensor. It is given that “The GPS output is in true North, East (and Down, not used here) and the magnetometer points to magnetic north (your declination is 9.7 degrees West in Waterloo, ON). Define additive noise distributions, with standard deviations in North and East of 0.50m, and in magnetic north of 10 degrees.”

Given the above information, our measurement noise covariance matrix  $Q$  is:

$$Q = \begin{bmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & 0.1745^2 \end{bmatrix} \quad (15)$$

Note that 10 degrees is 0.1745 radians. With a mean  $\mu$  of:

$$\mu = \begin{bmatrix} 0 \\ 0 \\ -0.1692 \end{bmatrix} \quad (16)$$

Note that 9.7 degrees is 0.1692 radians. Now that we have  $\mu$  and  $Q$ , the Gaussian Additive Noise then becomes:

$$\delta_t \sim \mathcal{N}(\mu, Q) \quad (17)$$

To complete the measurement model it becomes:

$$y(t) = C(t)x(t) + \delta_t \quad (18)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \delta_t \quad (19)$$

## 4 Defining the Extended Kalman Filter for the Omni-directional Robot

The Extended Kalman filter (EKF) has two stages, the prediction and measurement updates. To complete the EKF we need to linearize the motion and measurement model, we detail that in the following sections.

### 4.1 Linearizing the Motion Model

Normally if our motion model and/or measurement model is linear we can simply use the Kalman Filter, however in the non-linear case we have to use the Extended Kalman Filter (EKF). In our case we have to linearize the non-linear motion model  $g(x_{t-1}, u_t)$  using 1st order Taylor Series Expansion:

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \left. \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}} \right|_{x_{t-1}=\mu_{t-1}} (x_{t-1} - \mu_{t-1}) \quad (20)$$

$$= g(\mu_{t-1}, u_t) + G_t \cdot (x_{t-1} - \mu_{t-1}) \quad (21)$$

Where  $G_t$  is the first order differential of the non-linear motion model, a.k.a the Jacobian of  $g(x_{t-1}, u_t)$ .

$$G_t = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} \end{bmatrix} \quad (22)$$

Since we only know  $\dot{g}(\theta, \omega)$  we need to convert it to  $g(x_{t-1}, u_t)$ , it is simply:

$$g(x_{t-1}, u_t) = x_{t-1} + \dot{g}(\theta, \omega)dt \quad (23)$$

$$= \begin{bmatrix} x_{1,t-1} + \dot{g}_1(\theta, \omega)dt \\ x_{2,t-1} + \dot{g}_2(\theta, \omega)dt \\ x_{3,t-1} + \dot{g}_3(\theta, \omega)dt \end{bmatrix} \quad (24)$$

Now that we have  $g(x_{t-1}, u_t)$  we can work out the Jacobian. One important note, the partial derivate of  $x_3$  in our case is the bearing  $\theta$  of the robot, the only term affected in  $\dot{g}(\theta, \omega)$  is the rotation matrix  $R(\theta)$ , since  $\dot{g}(\theta, \omega)$  is:

$$\dot{g}(\theta, \omega) = R(\theta)^{-1} J_1^{-1} J_2 \omega \quad (25)$$



Where  $R(\theta)$ ,  $J_1$ ,  $J_2$  and  $\omega$  are (derived in Section 1):

$$R(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$J_1^{-1} = \begin{bmatrix} 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3l} & \frac{1}{3l} & \frac{1}{3l} \end{bmatrix} \quad (27)$$

$$J_2 = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \quad (28)$$

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (29)$$

Therefore to differentiate rotation matrix  $R(\theta)$  with respect to  $x_3$  (or  $\theta$ ), we can treat  $J_1$   $J_2$  and  $\omega$  as constant vector  $A$ . The partial derivative of  $R(\theta)$  then becomes:

$$\frac{\partial R(\theta)^{-1}}{\partial x_3} = \begin{bmatrix} -\sin(x_3) & \cos(x_3) & 0 \\ -\cos(x_3) & -\sin(x_3) & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (30)$$

$$A = J_1^{-1} J_2 \omega \quad (31)$$

Or in terms of  $g$  and  $x$ .

$$\frac{\partial g}{\partial x_3} = \frac{\partial R(\theta)^{-1}}{\partial x_3} A \quad (32)$$

$$= \begin{bmatrix} -\sin(x_3) & \cos(x_3) & 0 \\ -\cos(x_3) & -\sin(x_3) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \quad (33)$$

$$= \begin{bmatrix} -A_1 \sin(x_3) + A_2 \cos(x_3) \\ -A_1 \cos(x_3) - A_2 \sin(x_3) \\ 0 \end{bmatrix} \quad (34)$$

Completing our Jacobian  $G_t$  we get:

$$G_t = \begin{bmatrix} 1 & 0 & -A_1 \sin(x_3) + A_2 \cos(x_3) \\ 0 & 1 & -A_1 \cos(x_3) - A_2 \sin(x_3) \\ 0 & 0 & 1 \end{bmatrix} \quad (35)$$

## 4.2 Linearizing the Measurement Model

As for the measurement model, it does not require linearization since in Section 3 we derived a linear measurement model. Therefore  $H_t = C_t$ .

$$H_t = C_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (36)$$

## 4.3 Summary

Now that we have defined what  $g(x_{t-1})$ ,  $G_t$  and  $H_t$  (along with initial values for the states) we have enough information to incorporate them into the EKF model.

## 5 Implementing the Extended Kalman Filter

In the EKF we have used the following values:

$$\begin{aligned}
 \mu_0 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
 u_0 &= \begin{bmatrix} -15.5 \\ -10.5 \\ 1.5 \end{bmatrix} \\
 H_t &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 G_t &= \begin{bmatrix} 1 & 0 & -A_1 \sin(x_3) - A_2 \cos(x_3) \\ 0 & 1 & A_1 \cos(x_3) - A_2 \sin(x_3) \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & -A_1 \sin(x_3) - A_2 \cos(x_3) \\ 0 & 1 & A_1 \cos(x_3) - A_2 \sin(x_3) \\ 0 & 0 & 1 \end{bmatrix}^{-1} A \\
 A_t &= J_1^{-1} J_2 \omega \\
 &= \begin{bmatrix} 0 & 1 & l \\ -\cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \\ \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & l \end{bmatrix}^{-1} \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}
 \end{aligned}$$

## 6 Adding GPS corrections to the EKF

Previously it was assumed that the GPS and Magnetometer both work at a rate of 10hz, however it is given that "corrections are sent for the x, y positions of the vehicle at an update rate of 1Hz. These corrections improve the GPS measurements at that instance only, with an improved standard deviation of 0.01m". This means that the measurement disturbance  $Q$  changes over time (i.e. every 1Hz) from:

$$Q = \begin{bmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & 0.1745^2 \end{bmatrix} \quad (37)$$

to (at every 1Hz):

$$Q = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & 0.1745^2 \end{bmatrix} \quad (38)$$