# Development of Cube Swarm for Search and Rescue Applications - Appendix D

## Extended Kalman Filter Calculations

William Albert (RBE), Phillip Brush (RBE/ME), Benjamin Dodge (ECE), Timothy Klein (RBE/CS), Andrew McCammon (ME), Jason Rockmael (RBE), Dang Tran (RBE)

Project Advisors: Professor Gregory Lewin (RBE/ME), Professor Shubbhi Taneja (CS), Professor Reinhold Ludwig (ECE)

Extended Kalman Filter Math Calculations

Dang Tran, Timothy Klein

April 25, 2024

**Abstract**

An overview of our calculations for the Extended Kalman filter

# 1 Introduction

To estimate the position of our robots, we will be using an Extended Kalman Filter (EKF). For the prediction step, we will be using the velocity of the wheels calculated from the encoders. For the correction step, we will be using the Arducam readings.

The wheel velocities will be notated by $V_R$ and $V_L$ for right and left respectively. The Arducam will be returning the distances from an AprilTag in the camera frame ($x_c$, $y_c$) and the yaw ($\gamma$), the angle between the camera x-axis and the heading of the robot with the AprilTag. A visualization of this can be seen in Figure 1. The global pose that we're estimating is in the form of $\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}$

# 2 Prediction Step

From [WB95], we have the following equations for the prediction step.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{1}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{2}$$

The f function in Equation 1 calculates the predicted current pose of the robot with the previous pose ($\hat{x}_{k-1}$), the control input ($u_{k-1}$), and some weights representing the zero-mean process noise $w_k$ in our calculations. For the prediction step, f can be approximated by setting $w_k$ to 0. To find f we'd need the angular velocity and rotational velocity of the differential which can be determined by:

$$\omega = \frac{V_R - V_L}{b}$$

$$v = \frac{V_R + V_L}{2}$$

The equations used to calculate f are as follows:

$$x_k = x_{k-1} + \Delta t \frac{V_R + V_L}{2} \cos\theta_{k-1}$$

$$y_k = y_{k-1} + \Delta t \frac{V_R + V_L}{2} \sin\theta_{k-1}$$

$$\theta_k = \theta_{k-1} + \Delta t \frac{V_R - V_L}{b}$$

The weights would be added to the variables $V_R$ and $V_L$. However, to avoid clutter we are going to state that $V_R = V_R + W_R$ and the same for $V_L$. b is the wheel track. The matrix version of these equations is as follows:

$$f = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t \cos\theta_{k-1}}{2} & \frac{\Delta t \cos\theta_{k-1}}{2} \\ \frac{\Delta t \sin\theta_{k-1}}{2} & \frac{\Delta t \sin\theta_{k-1}}{2} \\ \frac{\Delta t}{b} & -\frac{\Delta t}{b} \end{bmatrix} \begin{bmatrix} V_R + W_R \\ V_L + W_L \end{bmatrix}$$

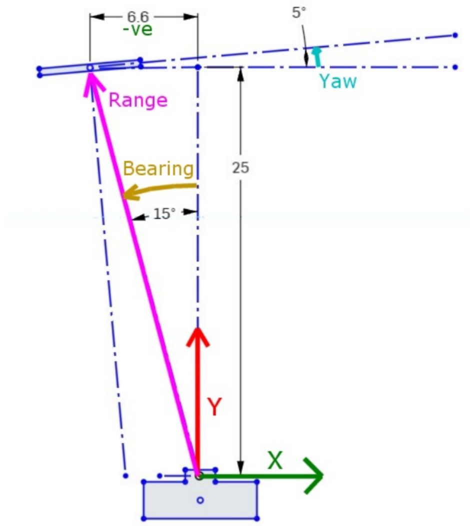


Figure 1: AprilTag measured values

Matrix A in Equation 2 is the Jacobian matrix of partial derivatives of f w.r.t. the pose,

$$A = \begin{bmatrix} 1 & 0 & -\Delta t \frac{V_R + V_L}{2} \sin \theta_{k-1} \\ 0 & 1 & \Delta t \frac{V_R + V_L}{2} \cos \theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix W is the Jacobian matrix of partial derivatives of f w.r.t. our weights $W_R$ and $W_L$,

$$W = \begin{bmatrix} \Delta t \frac{\cos \theta_{k-1}}{2} & \Delta t \frac{\cos \theta_{k-1}}{2} \\ \Delta t \frac{\sin \theta_{k-1}}{2} & \Delta t \frac{\sin \theta_{k-1}}{2} \\ \frac{\Delta t}{b} & -\frac{\Delta t}{b} \end{bmatrix}$$

P will be initialized as $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and Q will consist of the variance of the weights:

$$Q = \begin{bmatrix} \sigma_{W_R}^2 & 0 \\ 0 & \sigma_{W_L}^2 \end{bmatrix}$$

# 3 Correction Step

Using the paper [WB95], we have the following equations for the correction steps.

$$K_k = P_k^- H_k^T (H_K P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \tag{4}$$

$$P_k = (I - K_k H_k) P_k^- \tag{5}$$

The h function in Equation 4 relates the current prediction of the robot's pose to the camera measurements, which is the distance range and the bearing angle as shown in the following equations:

$$d = \sqrt{(y_2 - y_k)^2 + (x_2 - x_k)^2}$$

$$\phi = \theta_k - \arctan\left(\frac{y_2 - y_k}{x_2 - x_k}\right)$$

The derivation of $\phi$ can be seen in Figure 3 and in the following equation:

$$\tan(\theta_k - \phi) = \frac{y_2 - y_1}{x_2 - x_1}$$

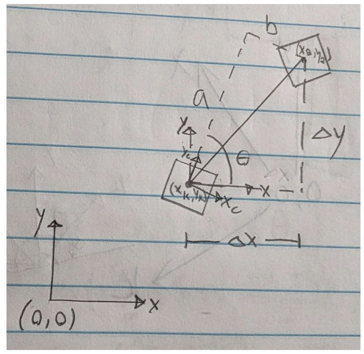


Figure 2: Rotation Matrix Illustration

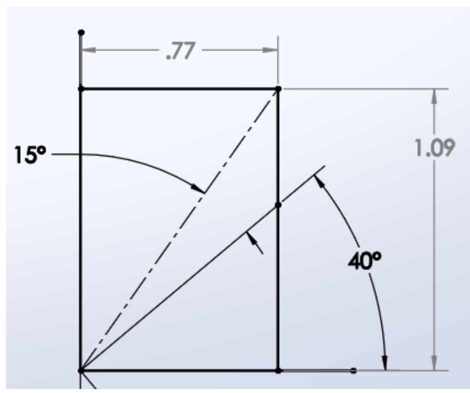


Figure 3: Illustration for $\phi$. With $\theta_k = 40^o$, $\phi = -15^o$ (The dotted range is to the left of the solid y-axis), $y_2 - y_k = 1.09$, and $x_2 - x_k = 0.77$.

$x_2$ and $y_2$ are the pose predictions of the other robot that the camera sees. This will be communicated over Bluetooth. There will also be weights attached to each $x_2, x_k, y_2, y_k$, and $\theta_k$. These weights aren't in the equations to avoid clutter so they'll be expressed as $x_2 = x_2 + V_{x_2}$ and so on.

The matrix H in Equations 3 and 5 are calculated by finding the Jacobian matrix of partial derivatives of the function h w.r.t. the pose:

$$H = \begin{bmatrix} -\frac{x_2 - x_k}{\sqrt{(y_2-y_k)^2+(x_2-x_k)^2}} & -\frac{y_2-y_k}{\sqrt{(y_2-y_k)^2+(x_2-x_k)^2}} & 0 \\ -\frac{y_2-y_k}{(y_2-y_k)^2+(x_2-x_k)^2} & -\frac{x_2-x_k}{(y_2-y_k)^2+(x_2-x_k)^2} & 1 \end{bmatrix}$$

The matrix V in Equation 3 is the Jacobian matrix of partial derivatives of the function h w.r.t $v = \begin{bmatrix} V_{xk} & V_{yk} & V_{\theta_k} & V_{x2} & V_{y2} \end{bmatrix}^T$:

$$V = \begin{bmatrix} (x_2 - x_k)\cos(90 - \theta_k) & -(y_2 - y_k)\sin(90 - \theta_k) & (x_2 - x_k)\sin(90 - \theta_k) + (y_2 - y_k)\cos(90 - \theta_k) \\ (x_2 - x_k)\sin(90 - \theta_k) & (y_2 - y_k)\cos(90 - \theta_k) & -(x_2 - x_k)\cos(90 - \theta_k + (y_2 - y_k)\sin(90 - \theta_k \\ 0 & 0 & -\theta_2 + \theta_k \end{bmatrix}$$

The matrix R will be a diagonal covariance matrix consisting of the variances of the weights:

$$R = \begin{bmatrix} \sigma^2_{W_x} & 0 & 0 \\ 0 & \sigma^2_{W_y} & 0 \\ 0 & 0 & \sigma^2_{W_\theta} \end{bmatrix}$$

# References

[WB95] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, 1995.