Sound Classification and Localization by Acoustic Sensing on Raspberry Pi 4

Jaidon Lybbert, University of Washington

January 24, 2024

Abstract

In this article a network of distributed acoustic sensing nodes is used to localize a sound signature using a time-difference-of-arrival (TDoA) algorithm. Accurate wall-clock times are obtained by GPS at each node, and a frame of acoustic data is recorded through a microphone on each node. Time differences are determined by correlation, and the resulting system of equations solved by the Chan-Ho algorithm to determine location. In a concurrent process, each node performs classification inference on a data frame using a pre-trained convolutional neural network (CNN). The resulting class identification is used to assign a process model to the tracked object for use in an Extended Kalman Filter (EKF) applied to the Chan-Ho localization result to make the final estimate of object position and trajectory.

1 Introduction

This document is organized as follows. In Section 2 5 6 7. I conclude in 8 by discussing the overall takeaways from my experiments and things I would like to try in the future.

2 Background

3 TensorFlow Lite

4 Kalman Filters

The Extended Kalman Filter (EKF) is a well-described algorithm, where discrete samples from one or more sensors is combined with an analytic model of a system to estimate system state. A block diagram of the Kalman Filter from *Kalman Filters for Beginners* by Phil Kim [1] is shown in Fig 1.

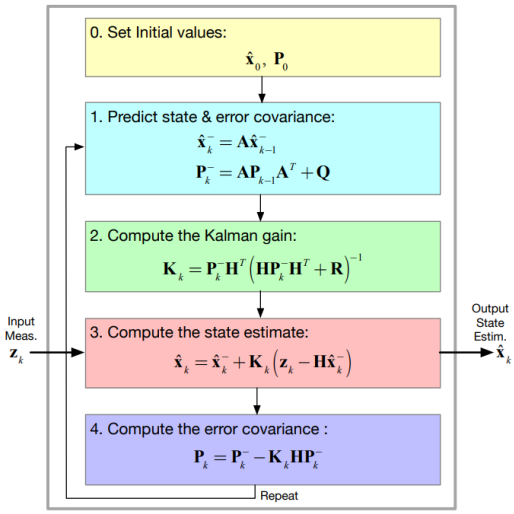

Figure 1: Block diagram of the Kalman Filter algorithm. $\hat{\mathbf{x}}_0$ represents the initial state.

5 Methods

5.1 Environment Setup

For classification, I will be using TensorFlow Lite

5.1.1 GPS

I used the Adafruit Ultimate GPS USB breakout board to get GPS data through **gpsd** [3].

```
sudo apt install gspd gpsd-clients
sudo systemctl stop gpsd.socket
sudo systemctl disable gpsd.socket
sudo killall gpsd
sudo gpsd /dev/ttyUSB -F /var/run/gpsd.sock
```

To test the device I used the command **cgps -s**, which prints out GPS status information to the console, as shown in Fig. 2.

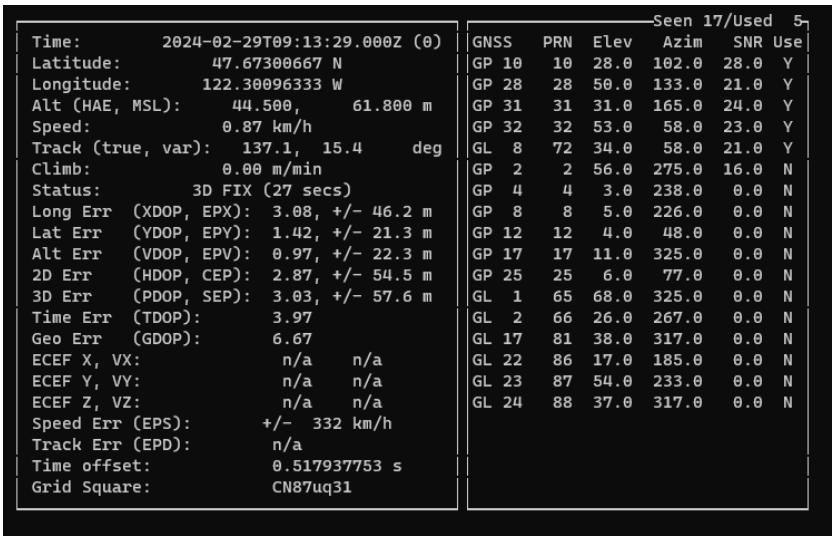

Figure 2: Terminal display of GPS information from the **cgps -s** command.

6 Results

7 Discussion

8 Conclusion

Note *All my code and project files for this and future reports, can be found on my GitHub repository [2].*

References

[1] P. Kim and L. Huh. *Kalman Filter for Beginners: With MATLAB Examples*. CreateSpace Independent Publishing Platform, 2011.

[2] Jaidon Lybbert. Classwork for embedded and real-time systems, university of washington, winter 2024. `https://github.com/jaidonlybbert/EEP522A-W24`, 2024. Accessed: 2024-01-23.

[3] Kevin Townsend. Adafruit ultimate gps with gpsd. `https://learn.adafruit.com/adafruit-ultimate-gps-on-the-raspberry-pi/setting-everything-up`, 2023. Accessed: 2024-02-29.