ANGLE APPROXIMATION FROM PRESSURE MEASUREMENTS

CANNON TUTTLE, CURTIS EVANS, SPENCER ASHTON, TYLER SANDERS

Abstract.

1. Problem Statement and Motivation

A BYU Acoustics Research Group project is currently using a microphone array and triangulation to calculate the direction of arrival of a source in relation to the center of the array. They've implemented some rudimentary filtering, but so far the results of the calculation are somewhat unreliable. We want to take this process and model it as a time series/Hidden Markov model in order to optimize the calculation process. We will potentially employ Kalman filtering in order to filter angle measurements. We will also explore whether cross-correlation or GCC-PHAT is the better measure of coherence between microphone signals for this purpose, and whether the hidden state should be modeled as an angle or be modeled directly as a series of coherence measurements. We will also experiment with ways to represent the observation space. This will also include trying to predict whether there is or isn't a speaker currently present.

Several of the noise reduction algorithms they have implemented depend on a highly accurate angle measurement. As those algorithms have already been implemented, we would be primarily concerned with the step of direction of arrival angle estimation and optimization, as well as optimally estimating the corresponding time delay. This would provide the research group with enhanced measurements for use in their noise processing algorithms.

This relates to the hidden markov model because we don't know what the angles are that we are looking for, but we do know how to take a measurement of the current pressure at each microphone. We will use these microphones to be our observed data to then figure out what these angles are by creating a HMM. The angles are to be calculated at discrete time steps according to the current pressure measurement. The Kalman filter might accurately represent the system and more optimally combine current and prior information about the angle in order to calculate a less noisy current angle estimate.

2. Data

The data for this project will be provided by the research team (Curtis Garner) that is currently working on it. The data includes measurements where the sound source is and is not moving, is and is not present, and measurements that do and don't include machine noise in the microphone signal.

3. Methods

We have tried several methods to compute the hidden angle measurement. We tried the following methods...

3.1. **State Space Model.** First we had to set up our continuous state space model. We needed angular velocity in our state space so to do that we do a finite difference approximation where $\theta_t' \approx \frac{\theta - \theta_{t-1}}{\Delta t}$. Now that we have angular velocity we can use it to make a simple forward Euler step in our state space. With the finite difference we need to save θ_{t-1} in the state space. This yields the following setup

$$\mathbf{x}_t = \begin{pmatrix} \theta_t \\ \theta_{t-1} \\ \theta' \end{pmatrix}, \; F = \begin{pmatrix} 1 & 0 & \Delta t \\ 1 & 0 & 0 \\ \frac{1}{\Delta t} & \frac{1}{\Delta t} & 0 \end{pmatrix}, \; H = \begin{pmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \end{pmatrix}$$

where H is of dimension $n \times 3$ for n observed angles. We don't have any control in our situation so our state space is

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \mathbf{w}_t,$$
$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{v}_t$$

.

3.2. **Kalman Filter.** We implemented the Kalman filter from the Volume 3 Textbook [1]. To understand how angle measurements worked in the Kalman Filter we tried it with stimulated data done in the lab manuals and class. We discovered a problem with the update step in angle wraparound from the angles in the range of

$$2\pi - \epsilon \leq \theta \leq 2\pi + \epsilon$$

for some $\epsilon \geq 0$. We notices that the more noise we added the larged this ϵ got. See figure 1 to get a visual.

We discovered that the problem was occurring in the update step where

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - H\hat{\mathbf{x}}_{k|k-1}.$$

A toy example would be if we are looking at an observed angle $\mathbf{z}_i = 15^\circ$ and a predicted angle of $(H\mathbf{x})_i = 355^\circ$ then $\mathbf{z}_i - (H\mathbf{x})_i = 15^\circ - 355^\circ = -340^\circ$ and we desire 20°. We could simply do this by noticing that $-340 \equiv 20 \pmod{360}$, however, if we said $\mathbf{z}_i = 355^\circ$ and $(H\mathbf{x})_i = 15^\circ$ then $\mathbf{z}_i - (H\mathbf{x})_i = 355^\circ - 15^\circ = 340^\circ$ which isn't what we want. We want the Kalman

filter to think of this difference as $-20°$. The best way to do this was by implementing Algorithm 1 which will get us the right differenced needed by the correct sign so the Kalman filter can function correctly.

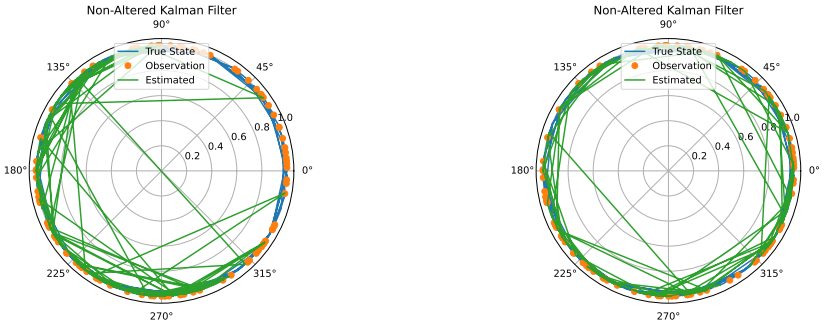

FIGURE 1. The non-altered and altered Kalman Filter

Algorithm 1 Process to Fix Wraparound

$\mathbf{v} \leftarrow \mathbf{z}_t - H\mathbf{x}_t$
if $|\mathbf{v}| \geq \pi$ **and** $\mathbf{v} \leq 0$ **then**
  $\mathbf{v} \leftarrow \mathbf{v} + 2\pi$
else if $|\mathbf{v}| \geq \pi$ **and** $\mathbf{v} \geq 0$ **then**
  $\mathbf{v} \leftarrow \mathbf{v} - 2\pi$
end if
$\tilde{\mathbf{y}}_t = \mathbf{v}$

3.3. **Particle Filter.** The second method we employed was another type of Bayesian filter, the particle filter [2]. The particle filter employs a large number of possible current states, each referred to as a "particle". Each particle has an associated weight, interpreted as the probability that particle accurately reflects the hidden state, with all weights summing to 1. As the source can possibly come in at any angle, we initialize the particles with uniformly distributed states and weights. At each time step, the filter propagates all of the particles forward in time according to the state space model, including adding state noise in the update. Then, using the data, it updates the weights of each particle by computing the posterior probabilities given data. Finally, it computes the state estimate as the weighted average of the states of all the particles.

In the update step, the filter computes $P(\theta|\text{data}) \approx P(\text{data}|\theta)P(\theta)$ for each particle. The prior $P(\theta)$ is given by the particle's previous weight, while $P(\text{data}|\theta)$ is calculated according to the probability distribution defined for the measurement noise. One of the benefits of the particle filter is that, unlike the normal distribution requirement of the Kalman filter, any arbitrary noise distribution may be employed. In this case, we used the Von

Mises distribution TODO [DEFINE,show plot]. This closely approximates a Normal distribution that is "wrapped" around the angle boundary between 0 and 2π. Thus, no other consideration of angle wrapping must be made for the particle filter. After the posterior weights have been calculated, they are again normalized by their sum so they represent a probability distribution.

TODO graph of Von Mises distribution with explanation of how Kappa parameter correlates to certainty on particle's location.

Particle filters suffer from what is called the "Degeneracy Problem", in which the weights become concentrated in a small number of the total particles. This is solved by resampling, a process in which low probability particles are replaced with higher probability particles. Detailing this problem and its solution is beyond our scope, but we used the TODO [fix format of this] systematic_resample function from filterpy to accomplish resampling.

As the particle filter keeps track of a large number of possible states which are distributed around the circle, it is much more flexible to nonlinear behavior. This is desirable in our case, as the source location may possibly reappear at any location in the state space. Unfortunately, the particle filter performs a similar number of computations for each individual particle as the Kalman filter performs in total. As we are using NUM_PARTICLES, our particle filter takes N_PARTICLES times as long to compute each step.

The Particle filter is defined with the following hyperparameters:

- $K1$ defines the analogous variance term for the distribution of the state noise,
- $K2$ defines the same for the measurements noise,
- N_particles defines the number of particles the model employs, and
- N_eff_particles defines the minimum number of "effective" particles allowed before particles are resampled.

Similar to the Kalman filter, we tuned the hyperparameters $K1$, $K2$, N_particles, and N_eff_particles by performing a gridsearch over many different possible values. A filter was constructed for each hyperparameter combination, with its performance tested across several different datasets. We then chose the hyperparameters that produced minimum average angular error over the different datasets.

3.4. **Leaky Filter.** TODO explain leaky filter and how it is working

4. Results

We had success with circular filtering techniques by using Algorithm 1 allowed us to use Kalman Filters with circular mechanics.

5. Analysis

TODO tradeoffs of Particle filter - the compute time 500x longer for 500 particles - particle filter can have a more flexible prior because it's already

got particles everywhere. - particle filter better at nonlinearities bc particles can be reassigned

- Both systems have a mechanism to capture how certain they are about the current state, - and both have a way to specify how much we "belive" current measurements

6. Ethical Considerations

This project seeks to augment the safety of those outside the machinery by increasing the awareness of the machine operators. However, before deployment, this model must be shown to improve safety as much as or more than it increased the safety perception of the operators. Risk compensation "is a theory which suggests that people typically adjust their behavior in response to perceived levels of risk, becoming more careful where they sense greater risk and less careful if they feel more protected" [citation]. This should be accounted for before deploying these methods in the field to ensure that the model performs robustly enough to truly enhance overall safety, despite risk compensation. Furthermore, it should be emphasized that operators should still use sight and radio communication to identify nearby individuals, and that people near machinery must still exercise caution.

7. Conclusion

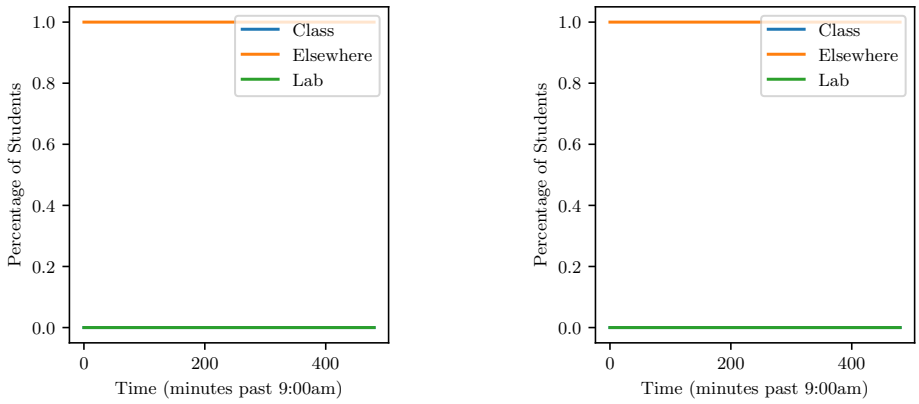

Figure 2. The constant alpha functions (left) along with the timeplot using IVP (right).

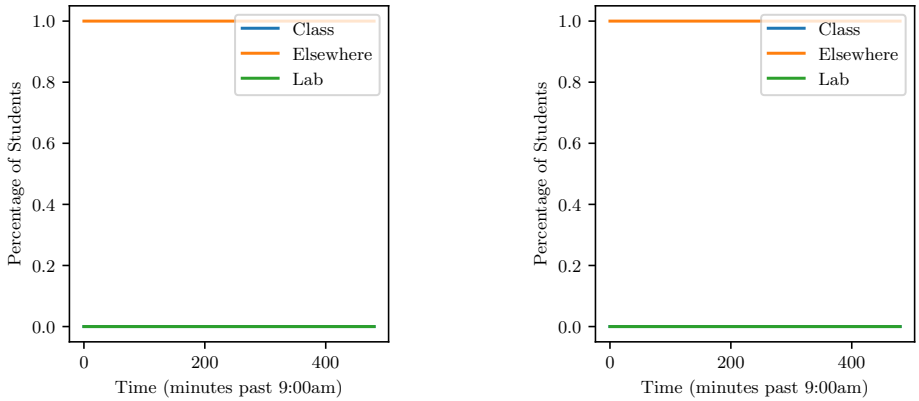

FIGURE 3. The discontinuous alpha functions (left) along with the timeplot using IVP (right).

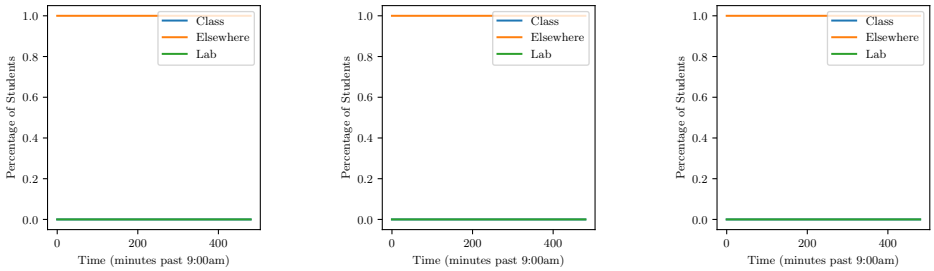

FIGURE 4. Simulating with $A_{interp.12}$ (left), giving the timeplots using both IVP (center) and BVP (right).

REFERENCES

[1] The ACME Volume 3 Textbook
[2] Labbe, Roger. "Kalman and Bayesian Filters in Python".
[3] Pierre Auger, Jean-Christophe Poggiale, Emergence of Population Growth Models: Fast Migration and Slow Growth, Journal of Theoretical Biology, Volume 182, Issue 2, 1996, Pages 99-108, ISSN 0022-5193, https://doi.org/10.1006/jtbi.1996.0145.