ADAPTING BAYESIAN FILTERS FOR USE WITH CIRCULAR DATA

CANNON TUTTLE, CURTIS EVANS, SPENCER ASHTON, TYLER SANDERS

Abstract. We adapt Bayesian filtering methods for use with circular data. We focus on a specific acoustics application, but accurate angle estimation is crucial in several fields. Due to the wrapping property of circular data, traditional filters are unsuitable for filtering circular data. We employ a simple continuous state-space model and successfully adapt the Kalman and Particle filters to account for the nature of circular data. We test these methods on both simulated and real data, finding our filters to perform far superior to an unmodified Kalman filter.

1. Problem Statement and Motivation

Imagine a construction site with heavy machinery, such as excavators, dump trucks, etc., in operation. The machine operator typically communicates with designated spotters via radio, but this leaves any other workers on the site unable to interface directly with the machine operators. A BYU acoustics project seeks to remedy this with a sound system that can pick up on the sound of human voice from outside the machine, filter out the noise, and reproduce the speech, including its relative direction to the machine, for the machine operator. This will augment the safety of those outside the machinery by increasing the awareness of the machine operators.

In order to replicate the directionality of the sound, the angle of arrival of the sound must be accurately estimated over time. Currently, this angle in relation to the center of the microphone array (see Figure 1) is calculated using pressure measurements and triangulation. Distance is also calculated, but is unimportant for this application. According to the researchers, a sufficiently good angle approximation for this application is within $\pm 15^\circ$ of the true angle. However, the noisy environment makes raw triangulation insufficiently accurate. Somewhat rudimentary filtering has been implemented to improve stability, but so far the results of the calculation are still unreliable.

Our aim is to improve the robustness of the angle estimation in order to allow the researchers to more accurately reproduce the sound directionality. Previous work has shown that an unmodified Kalman Filter does not work well on circular data [1]. Various distributions, including a wrapped normal distribution and the Von Mises distribution have been successfully implemented to be used with Kalman type filters. However, these adaptations are complex and not easily implemented, as they include approximating the

Date: April 12, 2024.

FIGURE 1. Microphone array used in data collection.

distributions with a wrapped Dirac mixture [1]. Particle filters have shown to be useful for filtering angle data, for example in robotics [6].

We modeled the angle of arrival evolution process as a continuous state space model and employed various Bayesian filtering techniques in order to optimize the angle estimation. This included adapting Kalman filters, normally used with data defined for all real numbers and gaussian distributed noise, to work with circular data that are defined on $[0, 2\pi]$ with the wrapping property at the boundary. We also adapted a Particle filter, a type of Bayesian filter that allows for the use of an arbitrary distribution, for use with circular probability distributions.

2. DATA

The data used for this project included both data given to us by the research team and simulated data. The original data is in the form of microphone pressure measurements that we then ran through a triangulation algorithm, provided to us by the research team, to produce angle measurements in time. Due to symmetry, each pair of microphones produces two possible angles. These can each be taken directly as possible measurements, producing 42 measurements per time step in our case, or all synthesized in order to produce a single angle measurement per time step. The provided triangulation algorithm produces data for both cases. However, the provided data didn't include certain extreme scenarios relevant to this application, including harsh discontinuities, frequent angle wraparounds, etc. To account

for these scenarios, we produced representative synthetic data by plotting a "true" angle, then adding gaussian noise to it to be used as the angle measurement. We also reserved real data to test our filter on after tuning our filter parameters.

3. Methods

We utilized several methods to compute the hidden angle measurement. Throughout this paper we use degrees in plots and examples, but all the measurements and computations were done in radians.

3.1. **State Space Model.** In order to represent the angle evolution over time, we used a state space model that included current angle, previous angle, and angular velocity. In this case, the speaker will either be stationary or moving near a constant velocity, so we did not include an acceleration term. Angular velocity was approximated as $\theta'_t \approx \frac{\theta_t - \theta_{t-1}}{\Delta t}$ and θ_t was approximated using the forward Euler step $\theta_t \approx \Delta t \theta'_t$. This yields the following:

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

where H is dimension $n \times 3$ for n observed angles. The state evolution is

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

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{v}_t.$$

All of our filters used the same state evolution equations.

3.2. **Kalman Filter.** We implemented two Kalman Filter [2][3] approaches. We term the Kalman filter illustrated in the textbooks the naïve Kalman filter and our altered version the circular Kalman filter.

3.2.1. *Naïve Kalman Filter.* We first implemented a naïve Kalman filter using the efficient Kalman Gain form. In the update step, the state estimation is updated as follows, where K_t is Kalman Gain:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + K_t(\mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}).$$

The quantity $\mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}$ is the difference between the observation, $\mathbf{z}_t$, and expected observation given the predicted state, $H\hat{\mathbf{x}}_{t|t-1}$. The naïve Kalman filter computes this as an arithmetic difference, however, it would be more accurate to compute a circular difference, as no two angles are more than 180° apart. This is illustraded by the examples in the table below:

Observed	Expected Observation	Arithmetic Diff	Circular Diff
15°	355°	−340°	20°
355°	15°	340°	−20°
120°	90°	30°	30°

3.2.2. *Circular Kalman Filter.* We adapted the Kalman filter for circualr data by changing the aforementioned update step to account for circular arithmetic. Specifically, we changed the calculation of the difference between true and expected observations to ensure that the difference calculated was never greater than 180° (i.e. measuring the distance between angles on a circle), while preserving sign to account for their relative position. This computes the circular difference illustrated in the table. The new update step is illustrated in Algorithm 1.

Algorithm 1 Altered Update Step

$\mathbf{v} \leftarrow \mathbf{z}_t - H\mathbf{x}_t$
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 [4]. 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 the particle accurately reflects the hidden state. All weights sum to 1 to form a probability distribution. 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 step. 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 $P(\theta|\text{data}) \propto P(\text{data}|\theta)P(\theta)$ 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 (see Figure 2, right side). 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.

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

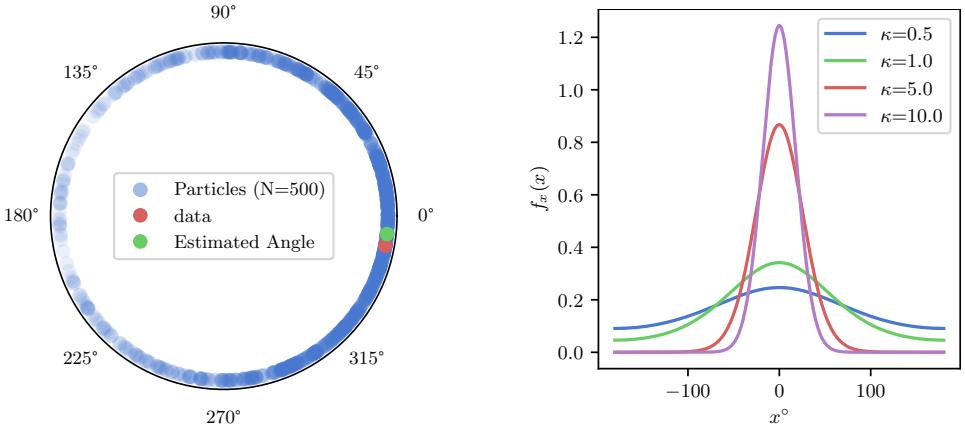

FIGURE 2. Left, an example particle filter at a given timestep. The particles are colored according to weight, with the heaviest weights clustering around the data angle. Right, the Von Mises distribution for various shape parameters (κ).

are replaced with higher probability particles. Detailing this problem and its solution is beyond our scope, but we used the `systematic_resample` function from `filterpy` to accomplish resampling.

As the particle filter keeps track of a large number of possible states around the circle, it is much more flexible to nonlinear behavior. This is desirable in our case, as a source 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 `n_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 shape term of the state noise distribution, `K2` defines the same for the measurement noise distribution, `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. We tuned the hyperparameters by performing a gridsearch over several different possible values using the metrics discussed below. A filter was constructed for each hyperparameter combination, with its performance tested across multiple datasets. We then chose the model that produced minimum average circular error over the different datasets (see Figure 3).

3.4. **Evaluation Metrics.** We also used the average absolute angle error as our primary evaluation metric. This took the average of the absolute circular difference between the true state and predicted state at each time step across the whole dataset. We referred to this as average circular error, `E`. We also used the total time to filter a given dataset as our other evaluation metric. We refer to this as filter time, `T`.

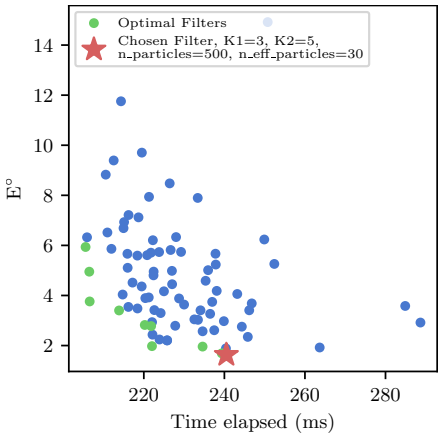

FIGURE 3. Hyperparameter tuning of the Particle filter. Any filters colored green may be considered "optimal".

4. Results and Analysis

We examine the success of the circular Kalman Filter (Figure 4), review filtering several simultaneous angle observations(Figure 5) , then consider the speed and accuracy of the various filters (Figures 6 and 7). Note, the figures best illustrate the results and performance of each filter.

4.1. **Circular vs Naïve.** As a simple demonstration, we compared the performance of the naïve and circular Kalman filters in tracking a source moving continuously in a circle. The circular Kalman filter modification resulted in a marked improvement in behavior across the boundary between 360° and 0° as compared to the naïve filter. The naïve filter tended toward the arithmetic mean of 180° of the data, while the circular Kalman filter accurately tracked the source around the entire circle. See Figure 4 below.

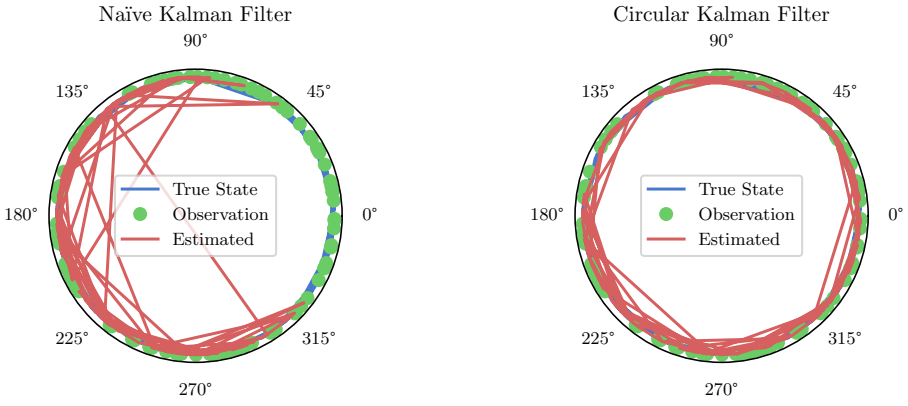

FIGURE 4. Naïve vs Circular Kalman Filter performance

4.2. **Filtering Multiple Angle Estimates.** As discussed in Data, each microphone pair technically calculates two possible measured angles, which are normally synthesized by the provided triangulation algorithm to find the most likely angle. Since the observation matrix H allows for an arbitrary number of measurements, we attempted filtering using all 42 angle measurements directly in our filters. See Figure 5. With this data, the Kalman filters failed to accurately track the data in any of the measured datasets. We believe this is because the array is not symmetric with respect to the angle of arrival of the sound, leading to angle measurements that are not distributed normally around the true angle. However, if the microphone array were to be circular, this method may be viable. In all other cases we treated the one angle returned by the triangulation algorithm as the measurement angle. The Particle filter successfully tracked the real angle in this case, and even ignored certain abberations that were returned by the single measurement operation. However, incorporating this number of measurements caused the specified particle filter to take $\frac{1}{3}$ as long as the total length of the data, possibly reaching filter times unsuitable for real time filtering.

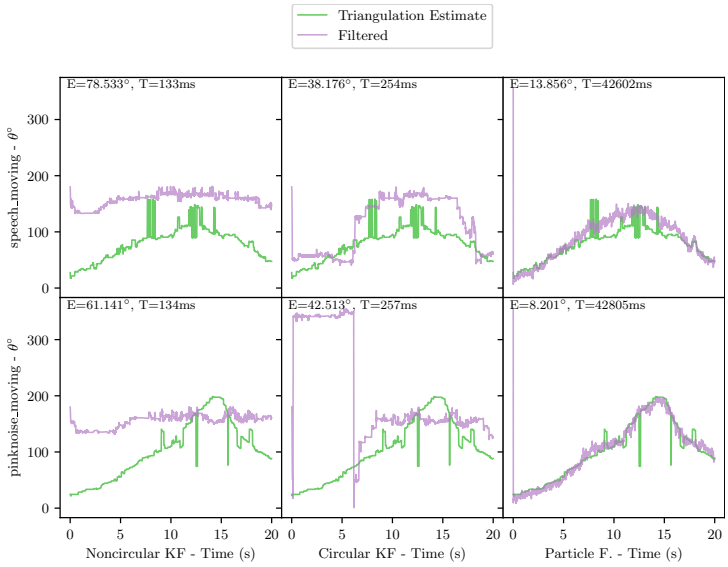

FIGURE 5. Filtering various real data with multiple simultaneous angle measurements. The triangulation estimate angle (green) is output from the provided triangulation algorithm. The estimated angle (pink) is being filtered.

4.3. **Speed.** The Kalman filters ran in about 3-5ms on 1 second of data, while the Particle filter took about 125ms, in total about 30 times longer than the Kalman filters. Thus, all of our filters can run in real time.

4.4. **Accuracy.** We tested the accuracy of the filters on the following synthetic and real datasets. See Figures 6 and 7 for quantitative results, as well as the discussion below. Note that on the real datasets, the true angle is unknown.

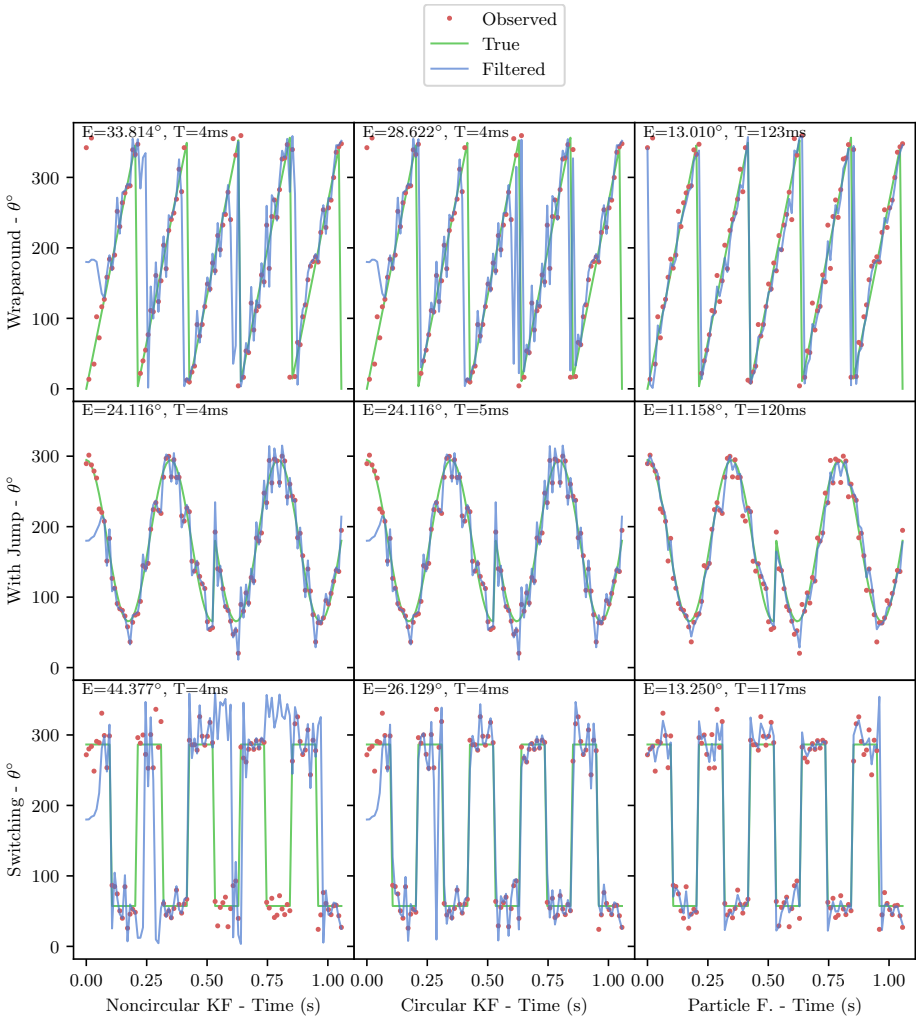

FIGURE 6. Running each filter (columns) on various simulated datasets (rows). The red dots represent observations of the green true angle. The observations alone are filtered to produce the blue estimate angles. Each cell displays average error (E) and the total time taken, in seconds (T).

4.4.1. *Wraparound.* This is a synthetic dataset that represents a source moving in a circle at a constant angular velocity to test a simple case involving wraparound. This was also the data used in 4. As discussed above, the

Naïve Kalman filter failed to track the angle, while the Circular Kalman filter successfully navigated the wraparound point. The particle filter also accurately tracked the angle around the wraparound.

4.4.2. *With Jump.* This is a synthetic dataset that represents a source moving back and forth then switching trajectory midway in order to account for a simple nonlinearity. Since there was no wraparound to consider, the Naïve and Circular Kalman filters both had equivalent performance, both accurately tracking the angle. The Particle filter tracked the angle about twice as accurately as the Kalman filters.

4.4.3. *Switching.* This is a synthetic dataset that represents the nonlinearity of jumping between angles across 0°. The Naïve Kalman filter performed poorly on this dataset. The Circular Kalman and Particle filters performed much better, with the Particle filter being about twice as accurate as the Circular Kalman filter.

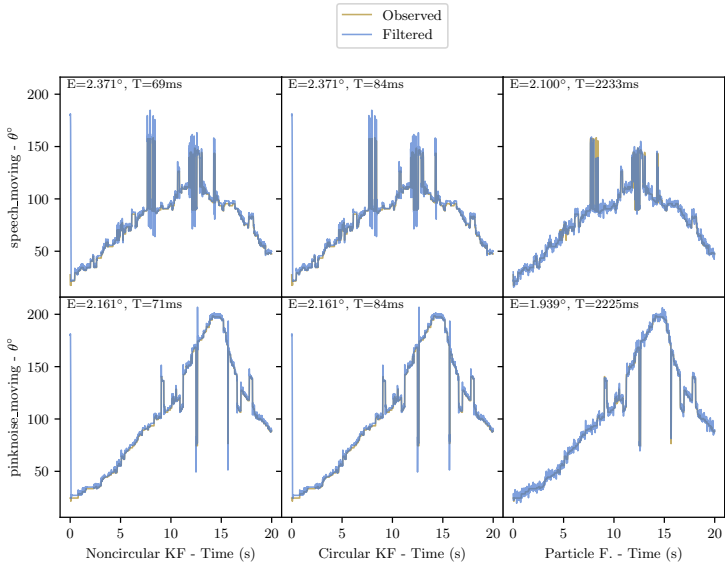

FIGURE 7. Running each filter (columns) on various real datasets (rows). Observed angles (yellow) are calculated using triangulation. The observed angle is then filtered (blue).

4.4.4. *Pink Noise Moving.* This dataset is a recording of a source producing pink noise, a type of filtered gaussian noise, while moving around the microphone array. All three filters behaved well on this dataset, with the particle filter again being slightly more accurate.

4.4.5. *Speech Moving.* This dataset is a recording of a source producing pink noise, a type of filtered gaussian noise, while moving around the microphone array. All three filters accurately tracked the source in this dataset, with the particle filter performing slightly more accurately.

5. Ethical Considerations

Although there are no data privacy concerns, we must consider the safety of operators using direction-filtered cabin audio in heavy machinery. Before deployment, this model must be shown to be a net positive to safety, despite operator risk compensation (acting less carefully because of perceived safety features [5]). Even then, operators should still use sight and radio communication to identify nearby individuals, and people near machinery must still exercise caution.

6. Conclusion

Our goal was to improve robustness of the angle estimation by adapting filtering methods for use with circular data. We derived a very simple modification to the Kalman filter that allows it to work for circular data, and we implemented a particle filter using circular probability distributions. We found the particle filter and the circular Kalman filter suitable for circular data. The particle filter handles nonlinearities well but comes at a computational cost, while the circular Kalman filter is computationally efficient but has its accuracy suffer in cases of strong nonlinearity. The Particle filter was around twice as accurate across all datasets as the Circular Kalman filter, but took about 30 times longer to run. If enough computing headroom is available, the Particle filter is a suitable choice for this application, but the Circular Kalman filter also performed well with very few computations. Only the Particle filter was consistently within the desired 15° of accuracy on all datasets. Due to the wraparound property of circular data, the naïve Kalman filter is unsuitable for circular data.

For the BYU Acoustics project specifically, if the computational cost of the filtering step is a concern, we would recommend use of the circular Kalman filter for this project due to its low cost and accurate performance except in cases of strong nonlinearity. If more computational headroom is allowed, the particle filter would be ideal, as it performed well even in cases of nonlinearities. Furthermore, the utility of these methods extends to any field in which angle measurements are taken, including robotics, navigation, and astronomy. The hyperparameters of our methods can easily be modified to adjust accuracy, flexibility, and computational cost depending on what is needed for the given scenario.

References

[1] G. Kurz, I. Gilitschenski and U. D. Hanebeck, "Recursive nonlinear filtering for circular data based on circular distributions," *2013 American Control Conference*, Washington, DC, USA, 2013, pp. 5439-5445, doi: 10.1109/ACC.2013.6580688.

[2] The ACME Volume 3 Textbook

[3] J. Humpherys and T. Jarvis, "Labs for Foundations of Applied Mathematics, Volume 3, Modeling with Uncertainty and Data."

[4] Labbe, Roger. “Kalman and Bayesian Filters in Python”.

[5] Masson, Maxime; Lamoureux, Julie; de Guise, Elaine (October 2019). "Self-reported risk-taking and sensation-seeking behavior predict helmet wear amongst Canadian ski and snowboard instructors". *Canadian Journal of Behavioural Science.* 52 (2): 121–130. doi:10.1037/cbs0000153. S2CID 210359660.

[6] I. Marković and I. Petrović, "Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering," *Robotics and Autonomous Systems*, Volume 58, Issue 11, 2010, pp. 1185-1196, doi: 10.1016/j.robot.2010.08.001