Sequential Monte Carlo - particle filters

Bayes formula

$$
\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{y}) &= \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}{\int p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{x}} \\
&\propto p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})
\end{aligned}
$$

$p(\boldsymbol{x})$ is prior.
$p(\boldsymbol{y}|\boldsymbol{x})$ is likelihood.

Example of such a model

$$\boldsymbol{x} \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$$

$$\boldsymbol{y}|\boldsymbol{x} \sim N(\boldsymbol{Hx}, \boldsymbol{T}), \quad \boldsymbol{y} = \boldsymbol{Hx} + \epsilon, \ \ \epsilon \sim N(\boldsymbol{0}, \boldsymbol{T})$$

$p(\boldsymbol{x}|\boldsymbol{y})$ is Gaussian with

$$\begin{aligned} E(\boldsymbol{x}|\boldsymbol{y}) &= \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_x \boldsymbol{H}'[\boldsymbol{H}\boldsymbol{\Sigma}_x\boldsymbol{H}' + \boldsymbol{T}]^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\mu}_x), \\ \text{Var}(\boldsymbol{x}|\boldsymbol{y}) &= \boldsymbol{\Sigma}_x - \boldsymbol{\Sigma}_x \boldsymbol{H}'[\boldsymbol{H}\boldsymbol{\Sigma}_x\boldsymbol{H}' + \boldsymbol{T}]^{-1}\boldsymbol{H}\boldsymbol{\Sigma}_x. \end{aligned}$$

Mean is linear in conditioning variable.
Variance is not dependent on conditioning variable, only correlations and variances.
In most other models the step from prior-posterior is not obvious (non-conjugate).

Particle representation of model

$$\boldsymbol{x}^1, \ldots, \boldsymbol{x}^B \sim p(\boldsymbol{x})$$

Samples (independent) from prior model. Samples are equally weighted $w^b = 1/B$, $b = 1, \ldots, B$.
In many application the goal is to update this sample representation to an approximate posterior sample.

$$\boldsymbol{x}^1, \ldots, \boldsymbol{x}^B \sim p(\boldsymbol{x}|\boldsymbol{y})$$

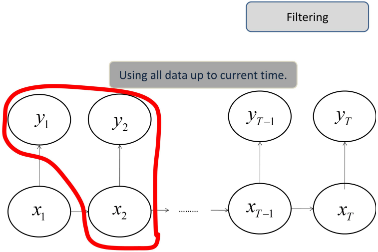
Samples can have non-equal weights w^b, $\sum_{b=1}^B w^b = 1$, or equal weights $1/B$.
For some methods the approximation converges to samples from the true posterior, under some regularity conditions.

Sequential Bayesian data assimilation

Methods like the particle filter or ensemble Kalman filtering methods have been very useful in **data assimilation** problems.

- Particle filter - introduced in 1990s for target tracking in real time and robotic applications.
- Ensemble Kalman filter - introduced in the 1990s for oceanography or meteorological applications.

Sequential Bayesian assimilation



$$p(\boldsymbol{x}_1), \quad p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad p(\boldsymbol{y}_t|\boldsymbol{x}_t), \; t = 2, 3, \ldots, T.$$

Dynamic model

Process model is described by:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\ldots,\boldsymbol{x}_1) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}),$$

This could be a differential equation, or it could be a simple linear process, or even a static process ($\boldsymbol{x}_t = \boldsymbol{x}_{t-1}$).

The data gathering process is described via the likelihood:

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t,\ldots,\boldsymbol{x}_1,\boldsymbol{y}_{t-1},\ldots,\boldsymbol{y}_1) = p(\boldsymbol{y}_t|\boldsymbol{x}_t)$$

This could also be nonlinear, or it could represent picking a subset of variables (with noise).

General formula

Filtering, solution:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1}) &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})d\boldsymbol{x}_{t-1}. \\ p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t) &= \frac{p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,,\boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})} \end{aligned}$$

Note Markov assumption in process, and conditionally independent data.

Kalman filter

For a Gaussian prior $\boldsymbol{x}_1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, linear Gaussian dynamic model $\boldsymbol{x}_t | \boldsymbol{x}_{t-1} \sim N(\boldsymbol{G}_t \boldsymbol{x}_{t-1}, \boldsymbol{Q})$, and linear Gaussian likelihood $\boldsymbol{y}_t | \boldsymbol{x}_t \sim N(\boldsymbol{H}_t \boldsymbol{x}_t, \boldsymbol{R})$, there exists an exact recursion for the filtering distribution: $p(\boldsymbol{x}_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) = N(\boldsymbol{m}_t, \boldsymbol{V}_t)$.

- Initialization:

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_1, \quad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_1,$$

- Recursive updating for $j = 1, \ldots, T$:

$$\begin{aligned}
\boldsymbol{S}_t &= \boldsymbol{H}_t \boldsymbol{\Sigma}_t \boldsymbol{H}_t^t + \boldsymbol{R}, \\
\boldsymbol{K}_t &= \boldsymbol{\Sigma}_t \boldsymbol{H}_t^t \boldsymbol{S}_t^{-1}, \\
\boldsymbol{m}_t &= \boldsymbol{\mu}_t + \boldsymbol{K}_t (\boldsymbol{y}_t - \boldsymbol{H}_t \boldsymbol{\mu}_t), \\
\boldsymbol{V}_t &= \boldsymbol{\Sigma}_t - \boldsymbol{K}_t \boldsymbol{H}_t \boldsymbol{\Sigma}_t. \\
\boldsymbol{\Sigma}_{j+1} &= \boldsymbol{G}_t \boldsymbol{V}_t \boldsymbol{G}_t^t + \boldsymbol{Q} \\
\boldsymbol{\mu}_{j+1} &= \boldsymbol{G}_t \boldsymbol{m}_t
\end{aligned}$$

Non-Gaussian or non-linear

In other situations there is usually no exact solution to the filtering distribution. Approximations:

- Extended Kalman filter (EKF) : linearization
- Unscented Kalman filter (UKF) : design points and 'numerical' integration
- Ensemble Kalman filter (EnKF) : Monte Carlo samples and linear updates
- Particle filter (PF) : Simulation and likelihood weighting.

Algorithms

Summary of some filtering methods ; pros and cons.

Criterion	EKF	UKF	EnKF	PF
Analytic conditioning	V	V	V	
MC based			V	V
Non-linear	w	V	V	V
Scales with dim.	V		V	
Reliable UQ		w		w

Sequential Monte Carlo methods for static problems

Most of the methods have also been used for static problems;

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t) &= \frac{p(\boldsymbol{x}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x})}{p(\boldsymbol{y}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})} \\ &\propto p(\boldsymbol{x}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}) \end{aligned}$$

Data are gradually incorporated in the model.
(Could also be done with same data many times: Multiple data assimilation / likelihood increase: $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{T}[p(\boldsymbol{y}|\boldsymbol{x})^{1/T}]$)

Particle filtering

Sample representation $(\boldsymbol{x}^b, w^b)$, $b = 1, \dots, B$.

The sample approximation is asymptotically exact, meaning that functional estimators of the sample converge to the theoretical counterpart, and that a central limit theorem holds for these estimators under certain regularity conditions.

Gordon et al. (1993), Doucet et al. (2000)

Recall general formula

Filtering, solution:

$$
\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1}) &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})d\boldsymbol{x}_{t-1}. \\
p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t) &= \frac{p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,,\boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})}
\end{aligned}
$$

The original particle filter propagates samples in the prediction step and re-weights them in the updating step.

Simulated Importance resampling : part I

Prediction:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})d\boldsymbol{x}_{t-1}.$$

For each (equally weighted) **propagate or sample $\boldsymbol{x}_{t-1}$** from $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})$, draw $\boldsymbol{x}_t^b \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^b)$.

The samples $\boldsymbol{x}_t^b$, $b = 1,\ldots,B$ are from the one-step predictive model. $p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})$.

Simulated Importance resampling : part II

Updating:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) \propto p(\boldsymbol{x}_t|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t).$$

For each sample $\boldsymbol{x}_t^b \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^b)$, evaluate $v^b = p(\boldsymbol{y}_t|\boldsymbol{x}_t^b)$ and the weight

$$w^b = \frac{v^b}{\sum_{c=1}^B v^c}$$

.
Resample B times from the probability vector defined by $(w^1, \ldots, w^B)$, to get samples x_t^b, $b = 1, \ldots, B$ from the filtering model. $p(\boldsymbol{x}_t|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$.

Particle filter representation

At each time t, the filtering distribution is approximated by (weighted) samples.

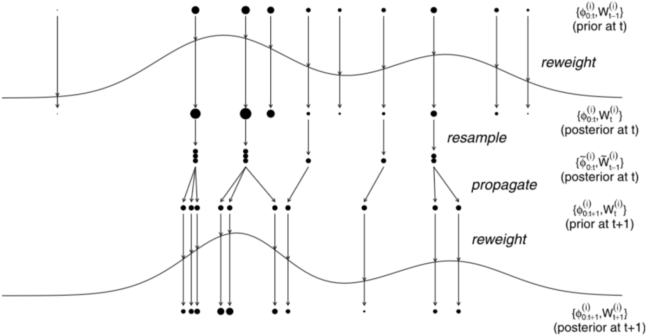
- Weighted samples $\{\boldsymbol{x}_t^b, w^b\}$, $b = 1, \dots, B$. $\sum_{b=1}^B w^b = 1$.
- Equally likely samples: $\{\boldsymbol{x}_t^b\}$, $w^b = 1/B$.

Monte Carlo approximation of a function

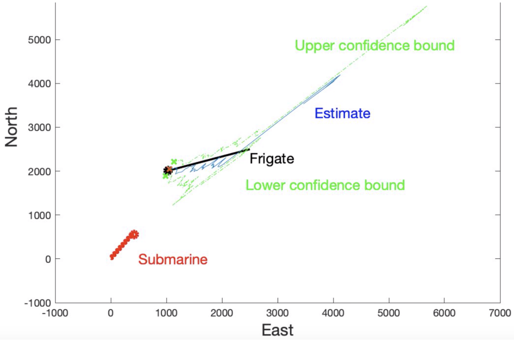
$$E(f(\boldsymbol{x}_t)|\boldsymbol{y}_1, \dots, \boldsymbol{y}_t) = \int f(\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_1, \dots, \boldsymbol{y}_t)d\boldsymbol{x}_t \approx \sum_{b=1}^B w^b f(\boldsymbol{x}_t^b)$$

.
With convergence in probability and in Gaussian asymptotic distribution, under some regularity condition on function f.

Illustration



Application: target tracking



$B = 1000$ samples. 4 dimensional system, positions and velocities. Submarine measures bearings only (non-linear), and so must move in pattern to determine states.

Variants of the Particle filter / Sequential Monte Carlo

- Re-sampling from w^b, $b = 1, \ldots, B$, brings random error to the system. Not always required, but can also add robustness to the system.
- Proposals from the 'prior' $p(\boldsymbol{x}_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$ are inefficient. Can improve this by importance sampling (or MCMC sampling within the particle filter).

Efficient implementation must balance effective approximation (fast and easy) and yet avoid **degeneracy** of samples meaning that one weight has all probability mass: $w^{(1)} \approx 1$, $w^{(2)} \approx 0$, ..., $w^{(B)} \approx 0$.

Sampling weights in particle filters

Proposals from $\boldsymbol{x}_t^b \sim p(\boldsymbol{x}_t|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$. (Unnormalized) weights become:

$$v^b \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^b)$$

No re-sampling is done, and next time, (unnormalized) weights become:

$$v^b \propto p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1}^b)v^b$$

Resampling is introducing random error in the filter. Weights w^b can instead be computed sequentially.

Sampling weights degeneracy

Even though this procedure works for $B \to \infty$, in practice the weights get very small and one weights dominates over the others (degeneracy). Resampling is then required.

$$\text{ESS} = \frac{1}{\text{Var}(w^b)}$$

Resampling when the effective sample size (ESS) is small. Then the weights are $1/B$ again.

The optimal strategy for resampling depends on the model. If there is much noise in the dynamic model, there is less need for resampling. For static models, resampling is required, often with some kernel to add variability in samples.

Importance sampling in particle filters

Proposal density $\boldsymbol{x}_t^b \sim q(\boldsymbol{x}_t)$. Weights become:

$$w^b \propto \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^b)p(\boldsymbol{x}_t^b|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})}{q(\boldsymbol{x}_t^b)}$$

Often, the proposal density approximates the dynamics or likelihood part. This will focus particles in interesting regions.

Idea of Importance sampling

Monte Carlo approximation of a function

$$E(f(\boldsymbol{x})) = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \int \frac{f(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x})d\boldsymbol{x} \approx \sum_{b=1}^{B} w^b f(\boldsymbol{x}^b)$$

.

$$\boldsymbol{x}^b \sim q(\boldsymbol{x}^b), \quad w^b \propto \frac{p(\boldsymbol{x}^b)}{q(\boldsymbol{x}^b)}$$

Optimal proposal density $q(\boldsymbol{x}) \propto f(\boldsymbol{x})p(\boldsymbol{x})$. Gives minimal Monte Carlo variance. (But hard to sample from this in general.)

Practical Importance sampling in dynamic problems

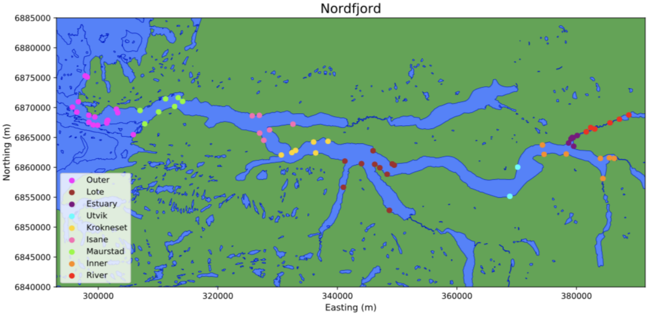
It is often useful to account for data $\boldsymbol{y}_t$ in the proposal q:

$$q(\boldsymbol{x}_t) \propto \hat{p}(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})$$

This guides the samples to interesting regions much more than 'prior' sampling from $q(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{t-1})$.

Improvements can be made by using MCMC in the sampling, at each time step (rather computationally demanding).

Application: telemetric fish tracking



Data are presence / absence of fish over time at sensors placed in buoys along the fjord. Each fish has a code.

(*Kaia A Høyheim*, MSc proj.)

Application: telemetric fish tracking

Gaussian proposal density (for entire path/velocity) using presence data only:

$$q(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) = Normal(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$$

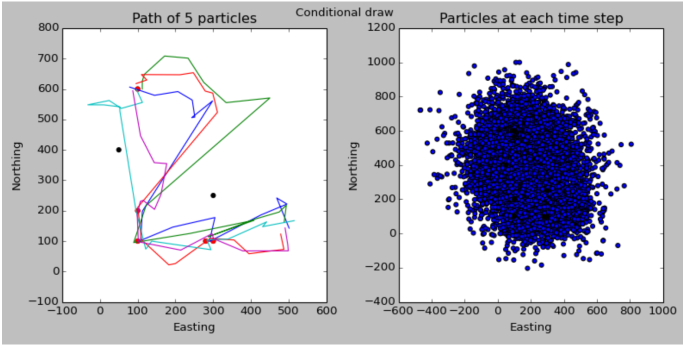
Absence / presence data (at time t only) for sequential correction in the weight

$$v^b = \frac{\prod_j p_j(\boldsymbol{x}_t^b)^{I(y_{t,j}=1)}(1 - p_j(\boldsymbol{x}_t^b))^{I(y_{t,j}=0)}}{q(\boldsymbol{x}^b|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)}$$

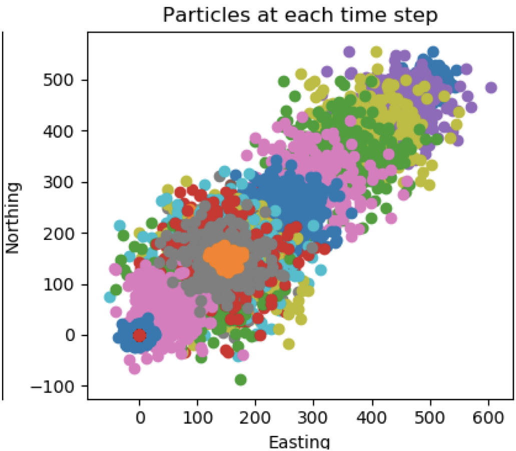
.

$p_j(\boldsymbol{x}_t^b)$ is detection probability of a sensor j, when true position is $\boldsymbol{x}_t^b$ at time t.

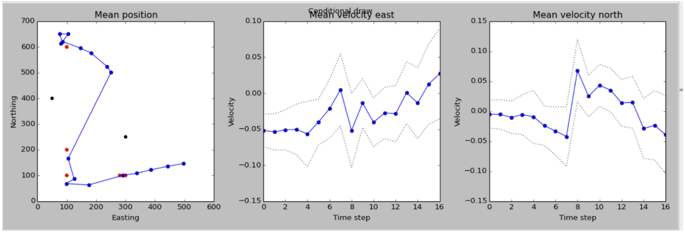
Application: telemetric fish tracking



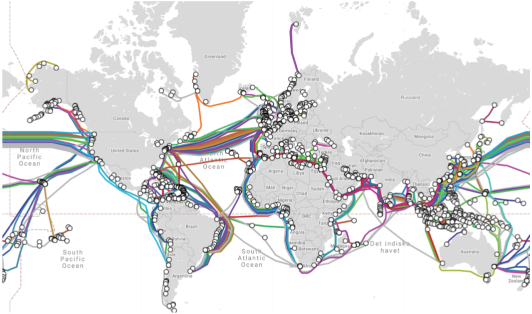
Application: telemetric fish tracking



Application: telemetric fish tracking



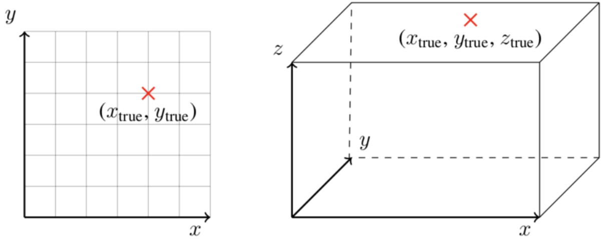
Application: fiber optic ship traffic monitoring



Dark fiber channels can measure vibrations caused by natural (earth quakes, landslides, etc) or man-made source (ships, sea-bed activity, etc).

(*Maia H Tømmerbakk*, MSc proj.)

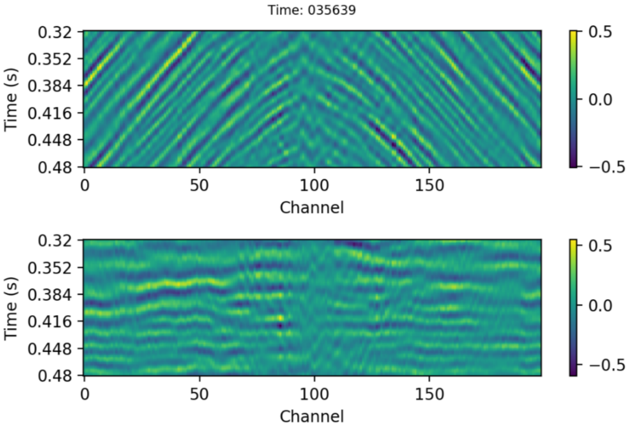
Application: fiber optic ship traffic monitoring



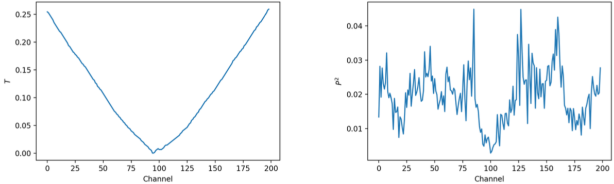
(a) Experimental setup in 2D.

(b) Experimental setup in 3D.

Application: fiber optic ship traffic monitoring



Application: fiber optic ship traffic monitoring



(a) Extracted travel time curve at time stamp *Time: 035639*.

(b) Extracted energy curve at time stamp *Time: 035639*.

Application: fiber optic ship traffic monitoring