On various applications of message passing on factor graphs

Justin Dauwels

www.dauwels.com

Signal and Information Processing Laboratory
ETH Zurich

December 13
Two threads

1. **Particular** problem of carrier-phase synchronization in single-carrier communications systems.

2. **Message-passing algorithms** for various applications.
Forney-style factor graphs (FFGs)

- **Factor graphs** represent the factorization of a function.
- **Example**

\[
f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_3, x_4, x_5)f_C(x_4).
\]

- **Rules** for drawing a factor graph
  - A **node** for every factor
  - An **edge** for every variable
  - Node \( g \) is connected to edge \( x \) iff variable \( x \) appears in factor \( g \)
Forney-style factor graphs (FFGs)

- **Factor graphs** represent the factorization of a function.

- **Example**
  \[
  f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_3, x_4, x_5)f_C(x_4)f_D(x_3).
  \]

- **Rules** for drawing a factor graph
  - A node for every factor
  - An edge for every variable
  - Node \( g \) is connected to edge \( x \) iff variable \( x \) appears in factor \( g \)

\[
f_{\boxdot} = \delta(x_3 - x_3')\delta(x_3 - x_3'')
\]
Forney-style factor graphs (FFGs)

- **Factor graphs** represent the factorization of a function.
- **Example**
  \[ f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_3, x_4, x_5)f_C(x_4)f_D(x_3). \]

- **Rules for drawing a factor graph**
  - A node for every factor
  - An edge for every variable
  - Node \( g \) is connected to edge \( x \) iff variable \( x \) appears in factor \( g \)

\[ f_D = \delta(x_3 - x'_3)\delta(x_3 - x''_3) \]
Computing marginals

- **Given**: Discrete probability mass function

\[ f(x_1, \ldots, x_8) = (f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3, x_4)) \cdot (f_4(x_4, x_5, x_6)f_5(x_5)(f_6(x_6, x_7, x_8)f_7(x_7))) \]

- **Wanted**: Marginal probability

\[ p(x_4) = \sum_{x_1, x_2, x_3, x_5, x_6, x_7, x_8} f(x_1, \ldots, x_8) \]

- This factorization can be represented by a factor graph.
Computing marginals

\[ f(x_1, \ldots, x_8) = \left(f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3, x_4)\right) \cdot \left(f_4(x_4, x_5, x_6)f_5(x_5)(f_6(x_6, x_7, x_8)f_7(x_7))\right) \]
Computing marginals

\[
p(x_4) = \left( \sum_{x_1} \sum_{x_2} \sum_{x_3} f_3(x_1, x_2, x_3, x_4) f_1(x_1) f_2(x_2) \right) \cdot \mu_{f_3 \rightarrow x_4}
\]

\[
\left( \sum_{x_5} \sum_{x_6} f_4(x_4, x_5, x_6) f_5(x_5) \left( \sum_{x_7} \sum_{x_8} f_6(x_6, x_7, x_8) f_7(x_7) \right) \right) \cdot \mu_{f_6 \rightarrow x_6}
\]

\[
\mu_{f_4 \rightarrow x_4}
\]
Sum-product algorithm ("belief propagation")

**Sum-product rule**

\[ \mu(y) \propto \sum_{x_1, \ldots, x_n} g(x_1, \ldots, x_n, y) \cdot \mu(x_1) \cdot \ldots \cdot \mu(x_n). \]

**Marginal**

\[ p(y) \propto \mu_{\rightarrow}(y) \mu_{\leftarrow}(y) \]

**Cyclic graphs**

Still applicable, but approximate marginals; may not convergence!
Two threads

1. **Particular** problem of carrier-phase synchronization in single-carrier communications systems.

2. **Message-passing algorithms** for various applications.
**Single-Carrier Communications System**

**Block diagram**

\[
\begin{align*}
    s_{BB}(t) &= \sum_i c_i g(t - iT) \\
    s(t) &= \text{Re}\left\{ s_{BB}(t) e^{j(\omega_c t + \phi_c)} \right\}
\end{align*}
\]

- \(\tau\): channel delay
- \(\omega_c\): carrier frequency
- \(\phi_c\): carrier phase
- \(T\): symbol length
- \(g(t)\): pulse shape
- \(c_i\): data symbols
- \(s_{BB}(t)\): baseband signal
- \(s(t)\): passband signal
- \(N(t)\): AWGN

\[r(t) = s(t - \tau) + N(t)\]
Single-Carrier Communications System

Block diagram

\[ s_{BB}(t) = \sum_i c_i g(t - iT) \]
\[ s(t) = \text{Re}\left\{ s_{BB}(t)e^{j(\omega_c t + \phi_c)} \right\} \]
\[ r(t) = s(t - \tau) + N(t) \]

**Labels:**
- \( \tau \): channel delay
- \( \omega_c \): carrier frequency
- \( \phi_c \): carrier phase
- \( T \): symbol length
- \( g(t) \): pulse shape
- \( c_i \): data symbols
- \( s_{BB}(t) \): baseband signal
- \( s(t) \): passband signal
- \( N(t) \): AWGN

---

12/79
## Questions

### Modeling
Which **physical mechanisms** are responsible for (phase) noise?  
How can (phase) noise be **modeled**?

### Algorithms
How can phase-estimation algorithms **systematically** be derived?

### Performance limits
How well can the (noisy) carrier phase be **estimated**?  
How much does the **information rate** decrease due to phase noise?
## Contributions

### Modeling
Which **physical mechanisms** are responsible for (phase) noise?
How can (phase) noise be **modeled**?

Simple intuitive model for phase noise.

### Algorithms
How can phase-estimation algorithms **systematically** be derived?
As message passing on factor graph of the system a hand.

### Performance limits
How well can the (noisy) carrier phase be **estimated**?
How much does the **information rate** decrease due to phase noise?

Computation of Cramér-Rao bounds/information rates/capacities.
Introduction  Model  Phase Estimation  Cramér-Rao-type Bounds  Information Rates  Analog Circuit  Summary  Outlook

Channel model

**Model for single-carrier system with slowly varying phase offset**

\[ Y_k = X_k e^{j\Theta_k} + W_k, \quad W_k \sim \mathcal{CN}_{0,\sigma_N^2}. \]

**Constant-phase model**

\[ \Theta_k = \Theta \in [0, 2\pi). \]

**Random-walk phase model**

\[ \Theta_k = (\Theta_{k-1} + N_k) \mod 2\pi, \quad N_k \sim \mathcal{N}_{0,\sigma_N^2}. \]

\(\sigma_N^2\) and \(\sigma_W^2\) are assumed to be known. The input symbols \(X_k\) are protected by an error-correcting code.
## Contributions

### Modeling

Which physical mechanisms are responsible for (phase) noise? How can (phase) noise be modeled?

Simple intuitive model for phase noise.

### Algorithms

How can phase-estimation algorithms *systematically* be derived?

As message passing on factor graph of the system a hand.

### Performance limits

How well can the (noisy) carrier phase be estimated? How much does the information rate decrease due to phase noise?

Computation of Cramér-Rao bounds/information rates/capacities.
Algorithms for joint decoding and phase estimation

Estimation task

Given a block of observations $Y \equiv (Y_1, Y_2, \ldots, Y_N)$, infer:

- the **coded symbols** $X \equiv (X_1, X_2, \ldots, X_N)$
- the **phase** $\Theta \equiv (\Theta_1, \Theta_2, \ldots, \Theta_N)$.

Derivation of message-passing estimation algorithms [Wiberg, 1996]

1. Draw factor graph of joint pdf $p(x, y, \theta)$.
2. Apply sum-product rule at each node.
3. If sum-product rule is **infeasible** at a certain node, then apply an **approximation** = choose appropriate **message types**.
4. Choose an update schedule.
**Factor graphs**

**Factor graph of** \( p(x, y, \theta) \)

- **Constant-phase model**
  - \( X_1 \) \( \times \) \( S_1 \) \( g \) \( Z_1 \)
  - \( X_2 \) \( \times \) \( S_2 \) \( g \) \( Z_2 \)
  - \( \ldots \)
  - \( X_L \) \( \times \) \( S_L \) \( g \) \( Z_L \)

- **Random-walk phase model**
  - \( X_1 \) \( \times \) \( S_1 \) \( g \) \( Z_1 \)
  - \( \Theta_1 \) \( p(\theta_2|\theta_1) \) \( Z_2 \)
  - \( \Theta_2 \) \( p(\theta_L|\theta_{L-1}) \) \( Z_L \)

**Factor graphs**

**Factor graph of** $p(x, y, \theta)$

- **Constant-phase model**
  - $X_1 \rightarrow S_1 \rightarrow Z_1$
  - $X_2 \rightarrow S_2 \rightarrow Z_2$
  - $\cdots$
  - $X_L \rightarrow S_L \rightarrow Z_L$

- **Random-walk phase model**
  - $X_1 \rightarrow S_1 \rightarrow Z_1$
  - $X_2 \rightarrow S_2 \rightarrow Z_2$
  - $\cdots$
  - $X_L \rightarrow S_L \rightarrow Z_L$

\[ p(y_1|z_1) \triangleq (2\pi \sigma_N^2)^{-1} e^{-|y_1 - z_1|^2/2\sigma_N^2} \]

\[ \delta_f (\cdot) \triangleq \delta \left[ f \left( b_k^{(1)}, \ldots, b_k^{(\log_2 M)} \right) - x_k \right] \]
Factor graphs

Factor graph of $p(x, y, \theta)$

Constant-phase model

Random-walk phase model

\[
S_k \triangleq e^{j\Theta_k} \quad g(\theta_k, s_k) \triangleq \delta(s_k - e^{j\theta_k}) \quad Z_k \triangleq X_k S_k \quad f(x_k, s_k, z_k) \triangleq \delta(z_k - x_k s_k)
\]
**Factor graphs**

**Factor graph of** \( p(x, y, \theta) \)

**Constant-phase model**

**Random-walk phase model**

\[
p(\theta_k | \theta_{k-1}) \triangleq (2\pi \sigma_W^2)^{-1/2} \sum_{n \in \mathbb{Z}} e^{-((\theta_k - \theta_{k-1}) + n2\pi)^2 / 2\sigma_W^2}
\]
Sum-product rule
Example

Sum-product rule

\[ \mu(x_k) \propto \int_0^{2\pi} \int_{z_k} \delta(z_k - x_k e^{j\theta_k}) \mu(\theta_k) \mu(z_k) \, d\theta_k \, dz_k, \]

\[ \propto \int_0^{2\pi} \mu(\theta_k) \mu(x_k e^{j\theta_k}) \, d\theta_k, \]

\[ \propto \int_0^{2\pi} \mu(\theta_k) e^{-|x_k e^{j\theta_k} - y_k|^2 / 2\sigma^2_{x_k}} \, d\theta_k \]

Intractable integral!
Message Types

\[ \mu(x_k) \propto \int_0^{2\pi} \mu(\theta_k) e^{-|x_k e^{j\theta_k} - y_k|^2 / 2\sigma_W^2} d\theta_k \]

Numerical integration

\[ \mu(x_k) \propto \sum_i \mu(\hat{\theta}_k^{(i)}) e^{-|x_k e^{j\hat{\theta}_k^{(i)}} - y_k|^2 / 2\sigma_W^2} \]

Particle method

\[ \mu(x_k) \propto \sum_i e^{-|x_k e^{j\hat{\theta}_k^{(i)}} - y_k|^2 / 2\sigma_W^2} \]

Decision based

\[ \mu(x_k) \approx e^{-|x_k e^{j\hat{\theta}_k} - y_k|^2 / 2\sigma_W^2} \]
Scheduling

Factor graph of \( p(x, y, \theta) \)

```
1⃝ 2⃝ 3⃝ 4⃝ 5⃝ 6⃝ 7⃝

Z_1 Z_2 Z_3 Z_4 Z_5 Z_6 Z_7

 X_1 X_2 X_3 X_4 X_5 X_6 X_7

Θ_1 Θ_2 Θ_3 Θ_4 Θ_5 Θ_6 Θ_7
```

Constant-phase model

```
X_1 X_2 ... X_L

Z_1 Z_2 Z_L

Θ_1 Θ_2 Θ_L
```

Random-walk phase model

```
X_1 X_2 ... X_L

Z_1 Z_2 Z_L

Θ_1 Θ_2 Θ_L
```
**Unification**

**Particle methods**
- Importance Sampling, Markov-Chain Monte Carlo,
- Metropolis-Hastings Algorithm, Gibbs Sampling, Simulated Annealing, Particle Filtering

**Decision based**
- Iterative Conditional Modes, Gradient Methods, Stochastic Approximation, Expectation maximization, SAGE, Gradient EM,
- Natural-Gradient Methods, Backpropagation Algorithm

**Combinations**
- Monte-Carlo EM, Stochastic EM

**Interpretation as message passing on factor graphs**
Identified generic local message-update rules for each approach.

*Joint work with Sascha Korl*
### Why we care...

#### Divide and conquer

Global estimation/detection problem accomplished by **simple local** computations. Complicated mathematical derivations avoided.

#### Disciplined approach

Deriving novel algorithms **systematically** by listing possible message update rules at each node in the graph.

#### Mish mash

Straightforward to **combine** several approaches, e.g., decision-based, particle-based etc., in one single algorithm.

#### Plug and play

Deriving novel algorithms by combining **tabulated** message update rules. Efficient use of earlier work.
## Contributions

### Modeling
Which physical mechanisms are responsible for (phase) noise? How can (phase) noise be modeled?  
Simple intuitive model for phase noise.

### Algorithms
How can phase-estimation algorithms systematically be derived?  
As message passing on factor graph of the system a hand.

### Performance limits
How well can the (noisy) carrier phase be estimated? How much does the information rate decrease due to phase noise?  
Computation of Cramér-Rao bounds/information rates/capacities.
Cramér-Rao-type bounds

What?

Lower bounds on the mean-squared estimation error (MSE)
e.g., \( \text{MSE} = E_{Y\theta} \left[ (\theta - \hat{\theta}(Y))^2 \right] \).

Motivation

Assessment of practical estimators
e.g., phase-estimation algorithms
Cramér-Rao-type bounds

Three different types

- **Standard** Cramér-Rao bounds: parameters
- **Bayesian** Cramér-Rao bounds: random variables
- **Hybrid** Cramér-Rao bounds: parameters and random variables
Bayesian Cramér-Rao bound: scalar case

**Theorem (Bayesian Cramér-Rao bound)**

Let \( p(x, y) \) be the joint pdf of \( x \in \mathbb{R} \) and \( y \triangleq (y_1, \ldots, y_N) \). If \( p(x) \) is zero at boundary of its support, then for any regular \( \hat{x}(y) \):

\[
E_{XY} [(x - \hat{x}(y))^2] \geq J^{-1},
\]

where the Bayesian information matrix \( J \) is defined as:

\[
J \triangleq E_{XY} \left[ \left( \frac{\partial}{\partial x} \log p(x, y) \right)^2 \right].
\]

**Properties**

- **MAP-estimator** achieves bound as \( SNR \) or \( N \to \infty \).
- **BCRB** holds for any regular \( \hat{x}(y) \) as \( SNR \) or \( N \to \infty \).
Bayesian Cramér-Rao bound: simple example

Example (Mean of a Gaussian random variable)

\[ Y = X + Z \text{ with } Z \sim \mathcal{N}(0, \sigma^2) \text{ with } \sigma^2 \text{ known and } X \in \mathbb{R} \text{ unknown.} \]

Estimate \( X \) from observations \( y_1, y_2, \ldots, y_N \) with prior \( p(X) \) for \( X \).

\[
p(x, y_1, y_2, \ldots, y_N) = p(x) \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-y_k)^2/2\sigma^2}.
\]

\[
J = -E_{XY} \left[ \frac{d^2}{dx^2} \log p(X, Y) \right]
= \frac{N}{\sigma^2} - E_X \left[ \frac{d^2}{dx^2} \log p(X) \right].
\]

\[
E_{XY}[(\hat{x}(X) - X)^2] \geq J^{-1} = \left( \frac{N}{\sigma^2} - E_X \left[ \frac{d^2}{dx^2} \log p(X) \right] \right)^{-1}
\]

If \( p(x) \) is Gaussian, then BCRB = \text{minimum achievable MSE!}
Vector case

Bayesian Cramér-Rao bound for component $X_k$

Given: joint pdf $p(x, y)$ of $x \triangleq (x_1, \ldots, x_M)$ and $y \triangleq (y_1, \ldots, y_N)$.

Lower bound for the MSE $E_{X_k Y} [(X_k - \hat{x}_k(Y))^2]$?

From marginal

$$E_{X_k Y} [(X_k - \hat{x}_k(Y))^2] \geq J_k^{-1},$$

with $J_k \triangleq E_{X_k Y} \left[ \left( \frac{\partial}{\partial x_k} \log p(x_k, y) \right)^2 \right].$

From joint pdf

$$E_{X_k Y} [(X_k - \hat{x}_k(Y))^2] \geq [J^{-1}]_{kk},$$

with $J_{ij} \triangleq E_{XY} \left[ \frac{\partial}{\partial x_i} \log p(x, y) \left( \frac{\partial}{\partial x_j} \log p(x, y) \right)^T \right].$

BCRB from marginal is tighter than from joint pdf, but more difficult to compute.
Algorithms

From joint pdf

- J is often **sparse**.
- Only **diagonal** elements of inverse required.
- **Local** computation of “small” matrices (message passing).

From marginal

- $J_k$ is usually **dense**.
- Key to $J_k$: $\frac{\partial}{\partial x_k} \log p(x_k, y) = E_{X \sim X_k | X_k Y} \left[ \frac{\partial}{\partial x_k} \log p(X, Y) \right]$
- **Expectation** computed by sum-product algorithm (or “belief propagation” or “probability propagation”).
Algorithms

Overview
We propose efficient and simple message-passing algorithms:

- for computing standard, Bayesian, hybrid Cramér-Rao bounds
- following both strategies.

Two examples
1. Random-walk phase model: Bayesian CRB from joint pdf
2. AR model: standard CRB from marginal.
Example 1: phase estimation; Bayesian CRB from joint pdf

Infer $X = X_1, \ldots, X_L$ and $\Theta = \Theta_1, \ldots, \Theta_L$ from $Y = Y_1, \ldots, Y_L$

Random-walk phase model with $\sigma^2_W = 10^{-4}\text{rad}^2$ with $L = 100$

MSE/BCRB for $\Theta_k$ (SNR = 4dB)
Example 2: AR model; standard CRB from marginal

Example (AR model)

Let $X_1, X_2, \ldots$ be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \cdots + a_M X_{k-M} + U_k, \quad U_k \sim \mathcal{N}(0, \sigma^2_U)$$

and let the process $Y = Y_1, Y_2, \ldots$ be defined as:

$$Y_k = X_k + W_k, \quad U_k \sim \mathcal{N}(0, \sigma^2_W).$$

Task

Estimate $a_1, \ldots, a_M, \sigma^2_U, \sigma^2_W$, and $X$ from observation $Y$. 
### Example (AR model)

Let $X_1, X_2, \ldots$ be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \cdots + a_M X_{k-M} + U_k, \quad U_k \sim \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \ldots$ be defined as:

$$Y_k = X_k + W_k, \quad U_k \sim \mathcal{N}(0, \sigma_W^2).$$

### Task

Estimate $a_1, \ldots, a_M, \sigma_U^2, \sigma_W^2$, and $X$ from observation $Y$.

### Observation

ML/MAP/MMSE-estimators are infeasible! Neither can their MSE be determined $\Rightarrow$ lower bounds on MMSE.
Theorem (Standard Cramér-Rao bound)

Let \( p(y; \theta) \) be the pdf of \( y \triangleq (y_1, \ldots, y_N) \) parameterized by \( \theta \triangleq (\theta_1, \ldots, \theta_N) \). If \( \hat{\theta}(y) \) is regular and unbiased estimator, then

\[
E_{Y;\theta} \left[ (\theta - \hat{\theta}(y))(\theta - \hat{\theta}(y))^T \right] \geq F^{-1}(\theta),
\]

where the Fisher information matrix \( F(\theta) \) is defined as

\[
F(\theta) \triangleq E_{Y;\theta} \left[ \nabla_{\theta} \log p(y; \theta) \nabla_{\theta}^T \log p(y; \theta) \right].
\]

Properties

- **ML-estimator** achieves bound as SNR or \( N \to \infty \).
- Standard CRB holds for any regular \( \hat{\theta}(y) \) as SNR or \( N \to \infty \).
Standard Cramér-Rao bound for the AR model

**Example (AR model)**

Let \( X_1, X_2, \ldots \) be a real random process defined by:

\[
X_k = a_1 X_{k-1} + a_2 X_{k-2} + \cdots + a_M X_{k-M} + U_k,
\]

\( U_k \) i.i.d. \( \sim \mathcal{N}(0, \sigma_U^2) \)

and let the process \( Y = Y_1, Y_2, \ldots \) be defined as:

\[
Y_k = X_k + W_k,
\]

\( U_k \) i.i.d. \( \sim \mathcal{N}(0, \sigma_W^2) \).

**Ingredients for the Cramér-Rao bound**

\( \theta = (a, \sigma_U^2, \sigma_W^2) \)

\[
F(\theta) \triangleq E_{Y;\theta} \left[ \nabla_{\theta} \log p(Y; \theta) \nabla_{\theta}^T \log p(Y; \theta) \right]
\]

\[
p(y; \theta) \triangleq \int_x p(x, y; \theta)
\]

\[
\nabla_{\theta} \log p(y; \theta) = E_{X|\theta, y} \left[ \nabla_{\theta} \log p(X, y; \theta) \right].
\]
Expectations $\nabla_\theta \log p(y; \theta) = E_{X|\theta,y} [\nabla_\theta \log p(X, y; \theta)]$

$$p(x, y|a, \sigma^2_W, \sigma^2_U) = \prod_k \mathcal{N}(x_k - \sum_{n=1}^{M} a_n x_{k-n} | 0, \sigma^2_U) \mathcal{N}(y_k - x_k | 0, \sigma^2_W).$$

As a consequence:

$$\nabla_\theta \log p(x, y|a, \sigma^2_W, \sigma^2_U) = \sum_k \nabla_\theta \log f_1(x_k, \ldots, x_{k-M}, a, \sigma^2_U) + \sum_k \nabla_\theta \log f_2(x_k, \sigma^2_W, y_k).$$
Expectations $\nabla_{\theta} \log p(y; \theta) = E_{X|\theta y} [\nabla_{\theta} \log p(X, y; \theta)]$ (2)

\[ E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla a_{i} \log f_{1}(X_{k}, \ldots, X_{k-M}, a, \sigma_{U}^{2}) \]

\[ = \frac{1}{\sigma_{U}^{2}} \left( E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k-i} X_{k} - \sum_{\ell=1}^{M} a_{\ell} E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k-i} X_{k-\ell} \right) \]

\[ E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla \sigma_{U}^{2} \log f_{1}(X_{k}, \ldots, X_{k-M}, a, \sigma_{U}^{2}) \]

\[ = -\frac{1}{2\sigma_{U}^{2}} + \frac{1}{2\sigma_{U}^{4}} \left( E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k}^{2} - 2 \sum_{\ell=1}^{M} a_{\ell} E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k} X_{k-\ell} \right) \]

\[ + \sum_{\ell=1}^{M} \sum_{m=1}^{M} a_{\ell} a_{m} E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k-\ell} X_{k-m} \right) \]

\[ E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla \sigma_{W}^{2} \log f_{2}(X_{k}, \sigma_{W}^{2}, y_{k}) \]

\[ = -\frac{1}{2\sigma_{W}^{2}} + \frac{1}{2\sigma_{W}^{4}} \left( y_{k}^{2} - 2y_{k} E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k} + E_{X|a} \sigma_{W}^{2} \sigma_{U}^{2} \nabla X_{k}^{2} \right). \]
Expectations $\nabla_\theta \log p(y; \theta) = \mathbb{E}_{X|\theta y} \left[ \nabla_\theta \log p(X, y; \theta) \right]$ (3)

Computation of means/variances/correlations

$E_{X|a}\sigma^2_W\sigma^2_U y$ computed by forward/backward Kalman recursions.

($=\text{sum-product rule with Gaussian messages}$)
Computing the Fisher information matrix of AR model

1. Generate a list of samples \( \{ \hat{y}(j) \}_{j=1}^{N} \) from \( p(y|\theta) \).

2. For \( j = 1, \ldots, N \):
   - Forward and backward Kalman recursion with \( y = \hat{y}(j) \).
   - Evaluate the expression:
     \[
     E_{X|\theta,\hat{y}(j)} \left[ \nabla_{\theta} \log p(X, \hat{y}(j); \theta) \right].
     \]

3. Compute the estimate \( \hat{F}(\theta) \) for \( F(\theta) \):
   \[
   \hat{F}(\theta) \triangleq \frac{1}{N} \sum_{j=1}^{N} \left[ E_{X|\theta,\hat{y}(j)} \left[ \nabla_{\theta} \log p(X, \hat{y}(j); \theta) \right] \right.
   \]
   \[
   \left. E_{X|\theta,\hat{y}(j)}^{T} \left[ \nabla_{\theta} \log p(X, \hat{y}(j); \theta) \right] \right].
   \]
Results for $\sigma_{W}^{2}$ ($\sigma_{U}^{2} = 0.1; \sigma_{W}^{2} = 0.001, 0.01, 0.1$)

Estimation algorithm by Sascha Korl.
### Summary: Cramér-Rao-type bounds

**What?**
Lower bounds on the mean-squared estimation error (MSE)

**Three different types**
- **Standard** Cramér-Rao bounds: parameters
- **Bayesian** Cramér-Rao bounds: random variables
- **Hybrid** Cramér-Rao bounds: parameters and random variables.

**Two strategies**
Inverse of information matrix of joint pdf or marginal.

**Algorithms**
We propose message-passing algorithms for computing the three types of CRBs following both strategies.
Other applications

- Other types of bounds, e.g., Weiss-Weinstein (discrete variables), Bhattacharyya, etc.
- Information geometry: natural-gradient-based algorithms
## Contributions

### Modeling
Which physical mechanisms are responsible for (phase) noise?  
How can (phase) noise be modeled?  

Simple intuitive model for phase noise.

### Algorithms
How can phase-estimation algorithms systematically be derived?  
As message passing on factor graph of the system a hand.

### Performance limits
How well can the (noisy) carrier phase be estimated?  
How much does the information rate decrease due to phase noise?  

Computation of Cramér-Rao bounds/information rates/capacities.
Information rate: introduction

**Objective**

Information rate $I(X; Y) \triangleq \lim_{n \to \infty} \frac{1}{n} I(X_1, \ldots, X_n; Y_1, \ldots, Y_n)$ between input process $X = (X_1, X_2, \ldots)$ and output process $Y = (Y_1, Y_2, \ldots)$ of time-invariant discrete-time channel with memory.

**State-space representation**

An ergodic stochastic process $S = (S_0, S_1, S_2, \ldots)$ such that

$$p(x^n, y^n, s^n_0) = p(s_0) \prod_{k=1}^{n} p(x_k, y_k, s_k | s_{k-1})$$

for all $n > 0$ and with $p(x_k, y_k, s_k | s_{k-1})$ not depending on $k$. 

![State-space diagram](image_url)
Basic principle

Reminder

\[ I(X; Y) = h(Y) - h(Y|X). \]

Shannon-McMillan-Breiman theorem

\begin{itemize}
\item \(-\frac{1}{n} \log p(X^n) \rightarrow H(X) \text{ w.p.} 1\)
\item \(-\frac{1}{n} \log p(Y^n) \rightarrow h(Y) \text{ w.p.} 1\)
\item \(-\frac{1}{n} \log p(X^n, Y^n) \rightarrow H(X) + h(Y|X) \text{ w.p.} 1.\)
\end{itemize}

Notation: \(X^n \triangleq (X_1, \ldots, X_n)\) and \(Y^n \triangleq (Y_1, \ldots, Y_n)\)

Algorithm

1. Sample two “very long” sequences \(x^n\) and \(y^n\).
2. Compute \(\log p(x^n), \log p(y^n),\) and \(\log p(x^n, y^n).\)
3. \(\hat{I}(X; Y) \triangleq \frac{1}{n} \log p(x^n, y^n) - \frac{1}{n} \log p(x^n) - \frac{1}{n} \log p(y^n).\)

[Arnold et al., Pfister et al., Sharma et al.]
Basic principle

Compute \( \log p(x^n), \log p(y^n), \) and \( \log p(x^n, y^n) \).

Discrete input space \( \mathcal{X} \) and state-space \( S \) [e.g., Arnold et al.]
Forward sum-product sweep = forward BCJR-recursion

Continuous input space \( \mathcal{X} \) and state-space \( S \)
Forward sum-product sweep by particle filtering. Expression \( p(x_k, s_k|s_{k-1}) \) not required!
E.g., stochastic differential/difference equation.
Forward sum-product sweep

Computation of \( p(y^n) \triangleq \int_{x^n} \int_{s_0^n} p(x^n, y^n, s_0^n) \).

Recursion

\[
\mu_k(s_k) = \int_{x_k} \int_{s_{k-1}} \mu_{k-1}(s_{k-1}) p(x_k, y_k, s_k | s_{k-1}) \, dx_k \, ds_{k-1}
\]

\[
= \int_{x_k} \int_{s_{k-1}} p(x^k, y^k, s^k) \, dx^k \, ds_{k-1}^{k-1}
\]

Marginal computed from message

\[
p(y^n) = \int_{s_n} \mu_n(s_n).
\]
Forward sum-product sweep

Computation of $p(y^n) \triangleq \int_{x^n} \int_{s_0^n} p(x^n, y^n, s_0^n)$. 

Recursion with normalization

\[ \mu_k(s_k) = \lambda_k \int_{x_k} \int_{s_k-1} \mu_{k-1}(s_{k-1}) p(x_k, y_k, s_k | s_{k-1}) dx_k ds_{k-1}. \]

\[ \int_{s_k} \mu_k(s_k) = 1 \text{ for all } k. \]

Marginal computed from normalization factors

\[ \frac{1}{n} \sum_{k=1}^{n} \log \lambda_k = -\frac{1}{n} \log p(y^n). \]
Computation of normalization factors

### Discrete input space $\mathcal{X}$ and state-space $\mathcal{S}$

Forward sum-product recursion $=$ forward BCJR recursion

$$
\lambda_k^{-1} = \sum_{s_k} \sum_{x_k} \sum_{s_{k-1}} \mu_{k-1}(s_{k-1}) \ p(x_k, s_k | s_{k-1}) \ p(y_k | x_k, s_k, s_{k-1}).
$$

### Continuous input space $\mathcal{X}$ and state-space $\mathcal{S}$

Forward sum-product recursion $=$ particle filtering

$$
\lambda_k^{-1} = \int_{s_k} \int_{x_k} \int_{s_{k-1}} \mu_{k-1}(s_{k-1}) \ p(x_k, s_k | s_{k-1}) \ p(y_k | x_k, s_k, s_{k-1}) \ ds_k \ dx_k \ ds_{k-1}
$$

$$
= E_{s_{k-1} s_k x_k} [p(y_k | x_k, s_k, s_{k-1})]
$$

$$
\approx \frac{1}{N} \sum_{\ell=1}^{N} p_{Y_k | X_k, S_k, S_{k-1}}(y_k | \hat{x}_k, \ell, \hat{s}_k, \ell, \hat{s}_{k-1}, \ell).
$$
Particle method

Algorithm

1. Begin with list \( \{ \hat{s}_{k-1, \ell} \} \) that represents \( \mu_{k-1} \).
2. Extend each particle \( \hat{s}_{k-1, \ell} \) to three-tuple \((\hat{s}_{k-1, \ell}, \hat{x}_{k, \ell}, \hat{s}_{k, \ell})\) by sampling from \( p(x_k, s_k | \hat{s}_{k-1, \ell}) \).
3. Compute an estimate of \( \lambda_k \).
4. **Resampling**: draw \( N \) samples from list \( \{ (\hat{s}_{k-1, \ell}, \hat{x}_{k, \ell}, \hat{s}_k, \ell) \} \) by choosing each three-tuple with probability proportional to \( p_{Y_k | x_k, s_k, s_{k-1}}(y_k | \hat{x}_k, \ell, \hat{s}_k, \ell, \hat{s}_{k-1, \ell}) \).
5. Drop \( \hat{s}_{k-1, \ell} \) and \( \hat{x}_{k, \ell} \) of each new three-tuple and obtain the new list \( \{ \hat{s}_k, \ell \} \).
Particle method

Algorithm

1. Begin with list \( \{\hat{s}_{k-1, \ell}\}^{N}_{\ell=1} \) that represents \( \mu_{k-1} \).
2. Extend each particle \( \hat{s}_{k-1, \ell} \) to three-tuple \( (\hat{s}_{k-1, \ell}, \hat{x}_{k, \ell}, \hat{s}_{k, \ell}) \) by sampling from \( p(x_{k, s_{k} | \hat{s}_{k-1, \ell}}) \).
3. Compute an estimate of \( \lambda_{k} \).
4. Resampling: draw \( N \) samples from list \( \{(\hat{s}_{k-1, \ell}, \hat{x}_{k, \ell}, \hat{s}_{k, \ell})\}^{N}_{\ell=1} \) by choosing each three-tuple with probability proportional to \( p(y_{k} | x_{k, s_{k}, s_{k-1}}(y_{k} | \hat{x}_{k, \ell}, \hat{s}_{k, \ell}, \hat{s}_{k-1, \ell})) \).
5. Drop \( \hat{s}_{k-1, \ell} \) and \( \hat{x}_{k, \ell} \) of each new three-tuple and obtain the new list \( \{\hat{s}_{k, \ell}\}^{N}_{\ell=1} \).
Observation

Sampling from $p(x_k, s_k | \hat{s}_{k-1,\ell})$

Closed-form expression for $p(x_k, s_k | \hat{s}_{k-1,\ell})$ not required!

State transitions may be modeled by

- Stochastic finite difference equations
- Stochastic ordinary differential equations
- Stochastic partial differential equations.

Allows detailed physical modeling of communications channel e.g., optical fibers, wave guides, hard drives etc.
Random-walk phase model with i.u.d. 4-PSK input symbols $X$
Contributions

Modeling
Which physical mechanisms are responsible for (phase) noise?
How can (phase) noise be modeled?
Simple intuitive model for phase noise.

Algorithms
How can phase-estimation algorithms systematically be derived?
As message passing on factor graph of the system a hand.

Performance limits
How well can the (noisy) carrier phase be estimated?
How much does the information rate decrease due to phase noise?
Computation of Cramér-Rao bounds/information rates/capacities.
Capacity of continuous memoryless channel

**Definition**

Given: memoryless channel with law $p(y|x)$

$$C(X; Y) \triangleq \sup_{p(x)} \int_x \int_y p(x)p(y|x) \log \frac{p(y|x)}{p(y)} dxdy \triangleq \sup_{p(x)} I(X; Y)$$

**Discrete input alphabet $\mathcal{X}$**

Blahut-Arimoto algorithm.

**Continuous input alphabet $\mathcal{X}$**

Particle-based approach: $p(x) \approx \{(\hat{x}_1, w_1), (\hat{x}_2, w_2), \ldots, (\hat{x}_N, w_N)\}$.

Method for channels with memory currently in development.
Blahut-Arimoto algorithm

1. **START** with some $p^{(0)}(x)$.

2. **ITERATE**

\[
p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x) \exp \left( D \left( p(y|x) \parallel p^{(k-1)}(y) \right) \right)
\]

\[
p^{(k)}(y) \triangleq \int_{x \in \mathcal{X}} p^{(k)}(x) p(y|x) dx.
\]

\[
Z^{(k)} \triangleq \int_{x \in \mathcal{X}} p^{(k-1)}(x) \exp \left( D \left( p(y|x) \parallel p^{(k-1)}(y) \right) \right) dx.
\]

**UNTIL**

\[
\max_{x \in \mathcal{X}} D \left( p(y|x) \parallel p^{(n)}(y) \right) - I^{(n)} < \varepsilon
\]

\[
I^{(n)} \triangleq \int_{x \in \mathcal{X}} p^{(n)}(x) D \left( p(y|x) \parallel p^{(n)}(y) \right) dx.
\]
Continuous channels

Blahut-Arimoto algorithm and continuous channels

- Blahut-Arimoto algorithms only **practical** for **discrete** channels.
- **Continuous** channels

\[ p(x) = \mathcal{L} \triangleq \{(\hat{x}_1, w_1), (\hat{x}_2, w_2), \ldots, (\hat{x}_N, w_N)\} \]

with \( \hat{x}_k \in \mathcal{X} \) and \( 0 \leq w_k \leq 1 \).
Algorithm

Non-convex finite-dimensional optimization problem

\[ \mathcal{L}^* \triangleq \arg\max_{\hat{x}, w} I(\hat{x}, w) \]

Solved by alternating maximization

\[ w^{(k)} \triangleq \arg\max_w I(\hat{x}^{(k-1)}, w) \quad \text{(Blahut-Arimoto)} \]

\[ \hat{x}^{(k)} \triangleq \arg\max_{\hat{x}} I(\hat{x}, w^{(k)}) \quad \text{(gradient method)} \]

Method for channels with memory currently in development.
Results: Gaussian channel

\[ Y_k = X_k + N_k \text{ with } N_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_N^2) \text{ and } \Pr[0 \leq X_k \leq 1] = 1 \]
Evolution of the mass points

\[ \text{SNR} = 13 \text{dB} \]
Objectives

Analog circuit that locks unto pseudo-random sequences (GPS, UWB, CDMA).

SNR = 0dB

### Objective

Analog circuit that locks unto pseudo-random sequences (GPS, UWB, CDMA).

### Method

Discrete-time message-passing algorithm for synchronization to LFSR-sequences converted into continuous-time.

### Practical result

Practical circuit built and tested: it works!

### Theoretical result

Connection between entrainment and ideas from estimation theory (“message passing”).

Based on the noisy observation $Y$ of the LFSR-sequence $X$, infer the actual state of the source.
Discrete-time synchronization task

\[ X = [\ldots, X_{k-1}, X_k, X_{k+1}, \ldots] \text{ with } X_k = X_{k-1} \oplus X_{k-3} \]

State diagram
Discrete-time synchronization task

\[ X = [\ldots, X_{k-1}, X_k, X_{k+1}, \ldots] \] with \( X_k = X_{k-1} \oplus X_{k-3} \)
Reminder: SP for EQU and XOR-node

\[ L \triangleq \log \frac{\mu(0)}{\mu(1)} \quad \Delta = \frac{\mu(0) - \mu(1)}{\mu(0) + \mu(1)} \]

\[ \delta[x - y] \delta[x - z] \]

\[ X \quad Z \]
\[ Y \]

\[ \begin{pmatrix} \mu_Z(0) \\ \mu_Z(1) \end{pmatrix} = \begin{pmatrix} \mu_X(0) \mu_Y(0) \\ \mu_X(0) \mu_X(1) \end{pmatrix} \]

\[ L_Z = L_X + L_Y \]
\[ \Delta Z = \frac{\Delta_X + \Delta_Y}{1 + \Delta_X \Delta_Y} \]

\[ \delta[x \oplus y \oplus z] \]

\[ X \quad Z \]
\[ Y \]

\[ \begin{pmatrix} \mu_Z(0) \\ \mu_Z(1) \end{pmatrix} = \begin{pmatrix} \mu_X(0) \mu_Y(0) + \mu_X(1) \mu_Y(1) \\ \mu_X(0) \mu_X(1) + \mu_X(1) \mu_X(0) \end{pmatrix} \]

\[ \tanh(L_Z/2) = \tanh(L_X/2) \cdot \tanh(L_Y/2) \]
\[ \Delta Z = \Delta_X \Delta_Y \]
Forward-only message passing on the factor graph
Forward-only message passing on the factor graph

Interpretation:
Filtering of the sequence $Y$ with a soft version of the LFSR.
Signal Source

- Delay elements replaced by linear filters.
- Output of the filters $X'_1(t)$ and $X'_2(t) \in \mathbb{R}$.
- Introduction of threshold functions $(X_1(t), X_2(t)$ and $X(t) \in \{-1, +1\})$.
- Multiplication corresponds to addition modulo 2.
From Discrete-Time to Continuous-Time

Synchronizing Circuit

A soft version of the signal source.
From Discrete-Time to Continuous-Time (2)

Overview

<table>
<thead>
<tr>
<th>Transmitter + Channel</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{k-3}$ $D$ $X_{k-2}$ $D$ $X_{k-1}$ $D$ $X_k$ memoryless channel $Y_k$</td>
<td>$\mu_{k-3}$ $D$ $\mu_{k-2}$ $D$ $\mu_{k-1}$ $D$ $\mu_k$ memoryless channel $Y_k$</td>
</tr>
<tr>
<td>$X_2(t)$ $\times$ $X_1(t)$ $\mu_{X_2}(t)$ $\times$ $\mu_{X_1}(t)$ $\mu_B(t)$</td>
<td>$\mu_B(t)$</td>
</tr>
<tr>
<td>$X_2(t)$ $H_2(s)$ $X_1(t)$ $H_1(s)$ $X(t)$ memoryless channel $Y(t)$</td>
<td>$\mu_{X_2}(t)$ $H_2(s)$ $\mu_{X_1}(t)$ $H_1(s)$ memoryless channel $Y(t)$</td>
</tr>
</tbody>
</table>
Results

Photograph and measurements by M. Frey and P. Merkli.
Hardware built by T. Schaerer.

\[ \text{SNR} = 0 \text{dB} \]
Summary

- Framework for deriving inference algorithms:
  - Factor graph = graphical representation of system
  - Algorithm = updating messages on factor graph.

- Message-passing algorithms for computing:
  - information rates
  - channel capacities
  - Cramér-Rao-type bounds.

- Analog circuit for PN-synchronization
  = message-passing algorithm as dynamical system.
Outlook

- Lower bounds on the MSE for discrete variables.
- Extension of the particle-based BA-algorithm to continuous channels with memory/feedback/side information.
- Analog electronic circuits for estimation.
- Novel applications of message-passing methods
  - Information geometry
  - Kernel methods.
On various applications of message passing on factor graphs

Justin Dauwels

www.dauwels.com

Signal and Information Processing Laboratory
ETH Zurich

December 13
Noise Sources

### Shot noise

Fluctuations of electrical currents due to **discreteness of charges**:

\[ l(t) = \sum_{k} q\delta(t - t_k), \quad t_k \sim \text{Poisson process} \]

Zero-mean **white noise** \( \tilde{l}(t) \triangleq l(t) - \mathbb{E}[l(t)] \) with spectral density:

\[ S_{\text{shot}}(f) = 2q\mathbb{E}[l(t)]. \]

### Thermal noise

Fluctuations of open-circuit voltage and closed-circuit current of conductor. Is nothing but **shot noise**! [Sarpeshkar et al., 1993]

### Flicker noise (or “1/f noise”)

Fluctuations of **inter-event time** \( \tau_k \triangleq t_k - t_{k-1} \) modeled by AR(1) model. [Kaulakys, 2004]
Phase Noise in Free-Running Clocks

Perturbed autonomous system

\[
\frac{dx}{dt} = f(x) + N(t), \quad x \in \mathbb{R}^n, \quad N(t) \text{ is "noise".}
\]

Phase offset due to "small" perturbation at \( t = t_0 \)

\[
\theta(t_0, n_0) \approx \gamma(x_s(t_0)) \cdot n_0.
\]
Phase Noise in Free-Running Clocks

Perturbed autonomous system

\[ \frac{dx}{dt} = f(x) + N(t), \quad x \in \mathbb{R}^n, \quad N(t) \text{ is "noise".} \]

\[ x(t_{t1^-}) = x_s(t_1 + \theta(t_0, n_0)) \]

Phase offset due to “small” perturbation at \( t = t_1 \gg t_0 \)

\[ \theta(t_0, n_0, t_1, n_1) \approx \gamma(x_s(t_0)) \cdot n_0 + \gamma(x_s(t_1 + \theta(x_s(0), n_0))) \cdot n_1. \]
Phase Noise in Free-Running Oscillator

**Continuous-time phase-noise model**

\[ \Theta(t) = \left[ \int_{-\infty}^{t} \gamma(t' + \Theta(t')) \cdot N(t') dt' \right] \mod 2\pi. \]

**Discrete-time phase-noise model**

\[ \Theta_k = (\Theta_{k-1} + N_k) \mod 2\pi, \quad \Theta_k \triangleq \Theta(kT_s) \]

with

\[ N_k \triangleq \int_{(k-1)T_s}^{kT_s} \gamma(t' + \Theta(t'))N(t')dt'. \]

**Discrete-time phase-noise model: white-noise sources**

\[ \Theta_k = (\Theta_{k-1} + N_k) \mod 2\pi, \quad N_k \sim \mathcal{N}_0,\sigma_N^2. \]
Numerical integration

Integral-product rule evaluated by numerical integration

\[ \mu_{f \rightarrow Y}(y) \propto \sum_{i_1, \ldots, i_N} f(y, \hat{x}_1^{(i_1)}, \ldots, \hat{x}_N^{(i_N)}) \cdot \mu_{X_1 \rightarrow f}(\hat{x}_1^{(i_1)}) \cdots \mu_{X_N \rightarrow f}(\hat{x}_N^{(i_N)}) , \]

where \( \hat{x}_k^{(i_k)} \) is the \( i_k \)-th quantization level of \( X_k \).
Phase estimation: results
EM: initial estimate vs. final estimate
Convergence

Random-walk phase model

**EM**

**Quantization**
FER as a function of number of quantization levels

(SNR = 0dB, 1dB, 2dB, and 3dB)
Particle methods

Integral-product rule evaluated by particle methods

\[ \mu_{f \rightarrow Y}(y) \propto \sum_{i_1, \ldots, i_N} f(y, \hat{x}^{(i_1)}_1, \ldots, \hat{x}^{(i_N)}_N) \cdot w^{(i_1)}_1 \cdots w^{(i_N)}_N, \]

where \( \hat{x}^{(i_k)}_k \) is the \( i_k \)-th particle of the particle list that represents \( \mu_{X_k \rightarrow f} \), and \( w^{(i_k)}_k \) is the weight of that particle.
Gibbs sampling

**Algorithm**

1. Choose an initial value $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$.
2. Choose an index $k$.
3. Draw a sample $\hat{x}_k$ from
   \[
   f(x_k) \triangleq \frac{f(\hat{x}_1, \ldots, \hat{x}_{k-1}, x_k, \hat{x}_{k+1}, \ldots, \hat{x}_N)}{\sum_{x_k} f(\hat{x}_1, \ldots, \hat{x}_{k-1}, x_k, \hat{x}_{k+1}, \ldots, \hat{x}_N)}.
   \]
4. Iterate 2–3 a “large” number of times.
Gibbs sampling

Message passing view

1. Select variable (EQU constraint node) \( Y \) in FG of \( f \).
2. EQU node \( Y \) generates message \( \hat{y} \) by sampling from:

\[
 f(y) \triangleq \frac{\mu_1(y) \cdots \mu_N(y)}{\sum_y \mu_1(y) \cdots \mu_N(y)},
\]

and broadcasts \( \hat{y} \) to its neighboring nodes \( f_k \) \((k = 1, \ldots, M)\).
3. Nodes \( f_k \) update messages \( \tilde{\mu} \) by applying sum-product rule with as incoming messages the samples \( \hat{y} \) and \( \hat{x}_\ell \) \((\ell = 1, \ldots, M)\).
Importance sampling

Suppose we wish to compute:

\[ E_f[g] \triangleq \int_x f(x)g(x), \quad (1) \]

but naive computation is intractable.

Generate a list samples \( \{\hat{x}^{(i)}\}_{i=1}^N \) from \( f \) and evaluate (1) as

\[ E_f[g] \triangleq \frac{1}{N} \sum_{i=1}^N g(\hat{x}^{(i)}). \quad (2) \]

Suppose that sampling from \( f \) is “hard”, hence (2) is infeasible.

Draw samples \( \{\hat{x}^{(i)}\}_{i=1}^N \) from a different function \( h \) with \( \text{supp}(f) \subseteq \text{supp}(h) \), compute (1) as

\[ E_f[g] \triangleq \frac{1}{N} \sum_{i=1}^N w^{(i)} g(\hat{x}^{(i)}) \quad \text{with} \quad w^{(i)} \triangleq \frac{f(\hat{x}^{(i)})}{h(\hat{x}^{(i)})}. \quad (3) \]
Importance sampling (2)

Suppose

\[ f(x) \triangleq f_1(x)f_2(x). \]

Draw samples \( \{\hat{x}^{(i)}\}_{i=1}^{N} \) from \( f_1 \) and weight those samples by the function \( f_2 \):

\[ w^{(i)} \triangleq f_2(\hat{x}^{(i)}). \]
Particle filtering

Particle filtering (or “sequential Monte-Carlo integration”) = forward-only message passing in a state-space model:

\[ f(s_0, s_2, \ldots, s_N, y_1, y_2, \ldots, y_N) \equiv f_A(s_0) \prod_{k=1}^{N} f_A(s_{k-1}, s_k)f_B(s_k, y_k), \]

where messages are represented by lists of samples.

\[ \tilde{\mu}_k \text{ is obtained from } \mu_{k-1} \text{ by weighted or unweighted sampling.} \]
\[ \mu_k \text{ is generated from } \tilde{\mu}_k \text{ by importance sampling.} \]
Smoothing

\[ f_A(s_{k-1}, s_k) = \mu_F^{k-1} \tilde{\mu}_k^F \]

\[ f_B(s_k, y_k) = \mu_B^k - 1 \tilde{\mu}_B^k \mu_U^k \]

\[ \cdots \]

\[ \text{particle list} \quad \text{closed-form} \]

\[ \cdots \quad \text{closed-form} \]

\[ \cdots \quad \text{closed-form} \]

\[ \text{particle list} \quad \text{closed-form} \]

\[ \cdots \quad \text{closed-form} \]
MCMC

Algorithm

1. Choose an initial value $\hat{x}$.
2. Sample $\hat{y}$ from $q(y|\hat{x})$.
3. Set
   $$\hat{x} \triangleq \hat{y} \text{ with probability } p$$
   where
   $$p \triangleq \min \left\{ \frac{f(\hat{y})}{f(\hat{x})}, 1 \right\}$$
4. Iterate 2–3 a sufficient number of times.
MCMC

Message passing view

1. Select variable (equality constraint node) $Y$ in FG of $f$.
2. Edge $Y$ generates the message $\hat{y}^{\text{new}}$ by sampling from $q(y|\hat{y})$.
3. Set $\hat{y} \triangleq \hat{y}^{\text{new}}$ with probability $p$ where
   \[ p \triangleq \min \left\{ \frac{f(\hat{y}^{\text{new}})}{f(\hat{y})}, 1 \right\} \text{ with } f(y) \triangleq \frac{\mu_1(y) \cdots \mu_N(y)}{\sum_y \mu_1(y) \cdots \mu_N(y)}. \]
   The message $\hat{y}$ is broadcast to the neighboring nodes $f_k$.
4. Nodes $f_k$ update outgoing messages $\tilde{\mu}$ by applying the SP rule with as incoming messages the samples $\hat{y}$ and $\hat{x}_\ell$. 

\[
\mu_1 \cdots \mu_N
\]

\[
\tilde{\mu}_1 \cdots \tilde{\mu}_M
\]

\[
\hat{y} \quad \hat{y}
\]

\[
Y
\]

\[
X_1 \cdots X_M
\]

\[
\hat{y} \quad \hat{y}
\]
Simulated Annealing

**Objective**

- to sample from a multivariate function $f(x_1, \ldots, x_N)$,
- to find the mode of the function $f$.

**Algorithm**

1. Choose an initial value $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$.
2. Choose an initial value $\alpha$ (e.g., $\alpha = 0.1$).
3. Sample a new value $\hat{y}$ from $q(y|\hat{x})$.
4. Set $\hat{x} \triangleq \hat{y}$ with probability $p$, where
   
   $$p \triangleq \min \left\{ \left( \frac{f(\hat{y})}{f(\hat{x})} \right)^\alpha, 1 \right\}$$

5. Iterate 3–4 a “large” number of times.
6. Increase $\alpha$ according to some schedule.
7. Iterate 5–6 until convergence or until available time is over.
Single value approximation

Integral-product rule evaluated by means of hard decision

\[ \mu_{f \rightarrow Y}(y) \propto f(y, \hat{x}_1, \ldots, \hat{x}_N), \]

where \( \hat{x}_k \) is a hard estimate of \( X_k \), representing the message \( \mu_{X_k \rightarrow f} \).

Gradient descent / sum-product

\[ \nabla_\Theta f_A(\theta) \propto \sum_{x_1, \ldots, x_n} \nabla_\theta g(x_1, \ldots, x_n, \theta) \cdot \prod_{\ell=1}^{n} \mu_{X_\ell \rightarrow g}(x_\ell). \]
Expectation Maximization: General problem

\[
\theta_{\text{max}} \triangleq \arg\max_{\theta} f(\theta)
\]

\(\theta\) takes values in \(\mathbb{R}\) or \(\mathbb{R}^n\)

\[
f(\theta) \triangleq \int_x f(x, \theta)dx
\]

\(\int_x g(x)dx\) stands for summation or integration.

In principle

1. Determine \(f(\theta)\) by sum-product message passing
2. \(\theta_{\text{max}} \triangleq \arg\max_{\theta} f(\theta)\) by max-product message-passing

Often infeasible, since

- Sum-product rule may lead to intractable integrals
- Maximization step may be infeasible.
Parameter estimation in state-space model

\[ f(x, \theta) = f_A(\theta_1) \prod_{k=1}^{n-1} f_A(\theta_k, \theta_{k+1}) \cdot f_B(x_0) \prod_{k=1}^{n} f_B(x_{k-1}, x_k, \theta_k, y_k) \]
Expectation Maximization

1. Make initial guess $\theta^{(0)}$
2. **Expectation** step
   \[ f^{(\ell)}(\theta) \triangleq \int_x f(x, \hat{\theta}^{(\ell)}) \log f(x, \theta) dx \]
3. **Maximization** step
   \[ \theta^{(\ell+1)} \triangleq \arg\max_{\theta} f^{(\ell)}(\theta) \]
4. Repeat 2–3 until convergence.
EM as message passing

\[ f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta) \]
EM as message passing (2)

Expectation step

\[ f^{(\ell)}(\theta) \triangleq \int_x f(x, \hat{\theta}^{(\ell)}) \log f(x, \theta) dx \]

Maximization step

\[ \theta^{(\ell+1)} \triangleq \arg\max_{\theta} f^{(\ell)}(\theta) \]

\[ \hat{\theta}^{(\ell+1)} = \arg\max_{\theta} \int_x f(x, \hat{\theta}^{(\ell)}) \log f(x, \theta) dx \]

\[ = \arg\max_{\theta} \int_x f_A(\hat{\theta}^{(\ell)}) f_B(x, \hat{\theta}^{(\ell)}) \log (f_A(\theta) f_B(x, \theta)) dx \]

\[ = \arg\max_{\theta} \int_x f_B(x, \hat{\theta}^{(\ell)}) \left( \log f_A(\theta) + \log f_B(x, \theta) \right) dx \]

\[ = \arg\max_{\theta} \left( \log f_A(\theta) + \frac{\int_x f_B(x, \hat{\theta}^{(\ell)}) \log f_B(x, \theta) dx}{\int_x f_B(x, \hat{\theta}^{(\ell)}) dx} \right) \]

\[ = \arg\max_{\theta} \left( \log f_A(\theta) + h(\theta) \right) \]
EM as message passing (3)

\[ f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta) \]

**Upwards message** \( h(\theta) \)

\[
h(\theta) = \frac{\int_x f_B(x, \hat{\theta}(\ell)) \log f_B(x, \theta) dx}{\int_x f_B(x, \hat{\theta}(\ell)) dx} = E_{p_B}[\log f_B(x, \theta)]
\]

**Downwards message** \( \hat{\theta}(\ell+1) \)

\[
\hat{\theta}(\ell+1) = \arg\max_\theta \left( \log f_A(\theta) + h(\theta) \right)
\]
EM as message passing (4)

Remarks

- If $f_A(\theta)$ is constant, then normalization may be omitted.
- Message $h(\theta)$ is not sum-product message!
- $f_A$ and $f_B$ often have a “nice” structure.
EM as message passing (5)

Trellis and state space models

\[ f_A(\theta) \triangleq f_{A_1}(\theta_1)f_{A_2}(\theta_1, \theta_2) \ldots f_{A_n}(\theta_{n-1}, \theta_n) \]
\[ f_B(x, \theta) \triangleq f_{B_0}(x_0)f_{B_1}(x_0, x_1, y_1, \theta_1)f_{B_2}(x_1, x_2, y_2, \theta_2) \ldots f_{B_n}(x_{n-1}, x_n, y_n, \theta_n) \]
EM as message passing (6)

\[ h(\theta) = \sum_{\ell=1}^{n} h_\ell(\theta_\ell) = \sum_{\ell=1}^{n} \int_{x_{\ell-1}} \int_{x_\ell} p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) \log f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_\ell) \, dx_{\ell-1} \, dx_\ell \]

\[ p_B(x_{\ell-1}, x_\ell, |y, \hat{\theta}) = \frac{f_{B_\ell}(x_{\ell-1}, x_k, y, \theta_\ell) \mu_x \rightarrow f_{B_\ell}(x_\ell) \mu_x \rightarrow f_{B_\ell}(x_{\ell-1})}{\int_{x_{\ell-1}} \int_{x_\ell} f_{B_\ell}(x_{\ell-1}, x_\ell, y, \theta_k) \mu_x \rightarrow f_{B_\ell}(x_\ell) \mu_x \rightarrow f_{B_\ell}(x_{\ell-1}) \, dx_{\ell-1} \, dx_\ell} \]

\[ (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n)^T = \arg\max_{\theta_1, \theta_2, \ldots, \theta_n} \left[ \log f_{A_1}(\theta_1) + \sum_{\ell=2}^{n} \log f_{A_\ell}(\theta_{\ell-1}, \theta_\ell) + \sum_{\ell=1}^{n} h_\ell(\theta_\ell) \right] \]
EM as message passing (7)

$h$-messages

\[ h(\theta_k) = \gamma^{-1} \int_z g(z_1, \ldots, z_m, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_m) \log g(z_1, \ldots, z_m, \theta_k) \, dz \]

\[ = \int_z p(z_1, \ldots, z_m|\hat{\theta}_k) \log g(z_1, \ldots, z_m, \theta_k) \, dz \]

\[ = E_{p(z_1, \ldots, z_m|\hat{\theta}_k)}[\log g(z_1, \ldots, z_m, \theta_k)] \]

\[ p(z_1, \ldots, z_m|\hat{\theta}_k) = \gamma^{-1} g(z_1, \ldots, z_m, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_m) \]

\[ \gamma = \int_z g(z_1, \ldots, z_m, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_m) \, dz \]

\[ \mu(z_k) \text{ are sum-product messages} \]
Expectation Maximization: Properties

**Theorem (Main property)**

\[ f(\hat{\theta}^{(k+1)}) \geq f(\hat{\theta}^{(k)}). \]

**Corollary**

*The global maximum* \( \theta^{\text{max}} \) *of* \( f(\theta) \) *is a fixed point of EM.*

**Theorem**

*The fixed points of EM are stationary points of* \( f(\theta) \).

**Theorem**

*A stationary point* \( \hat{\theta}^{\text{stat}} \) *of* \( f \) *is a fixed point of EM, if* \( \bar{f}(\theta, \hat{\theta}^{\text{stat}}) \) *with*

\[ \bar{f}(\theta, \theta') \triangleq \sum_x f(x, \theta') \log f(x, \theta), \]

*is concave in* \( \theta \).
EM and compound nodes
Hybrid EM

\[ h(\theta_k) = \sum_{z_1} \cdots \sum_{z_n} p(z_1, \ldots, z_n | \hat{\theta}_k) \log f(z_1, \ldots, z_n, \theta_k), \]

\[ = \gamma^{-1} \sum_{z_1} \cdots \sum_{z_n} f(z_1, \ldots, z_n, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_n) \cdot \log f(z_1, \ldots, z_n, \theta_k), \]

with

\[ \gamma \triangleq \sum_{z_1} \cdots \sum_{z_n} f(z_1, \ldots, z_n, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_n), \]

and

\[ f(z_1, \ldots, z_n, \theta_k) \propto \sum_{z_1'} \cdots \sum_{z_m'} g(z_1, \ldots, z_n, z_1', \ldots, z_m', \theta_k) \]

\[ \cdot \mu(z_1') \cdots \mu(z_m'), \]

where \( \mu(z_1), \ldots, \mu(z_n), \mu(z_1'), \ldots, \mu(z_m') \) are standard sum-product messages.
Example
(Hybrid) EM: properties

Theorem (Cycle-free $f_B(x, \theta)$)

Assume that a factor graph of a global function $f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta)$ is available whose subgraph $f_B(x, \theta)$ is cycle-free. The fixed points of a hybrid EM algorithm applied on that factor graph are stationary points of the marginal $f(\theta)$.

Theorem (Cyclic $f_B(x, \theta)$)

Assume that a factor graph of a global function $f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta)$ is available (whose subgraph $f_B(x, \theta)$ may be cycle-free or cyclic). The fixed points of a (hybrid) EM algorithm applied on that factor graph are stationary points of the function $\hat{f}(\theta)$, defined as:

$$
\log \hat{f}(\theta) \triangleq \log f_A(\theta) + \int_{-\infty}^{\theta} E_{b(x|\tilde{\theta})} \left[ \nabla_{\theta} \log f_B(x, \tilde{\theta}) \right] d\tilde{\theta},
$$

where the beliefs $b(\cdot|\theta)$ are computed by means of the sum-product messages available at convergence of the sum-product algorithm.
Steepest descent

Tries to solve

\[ \hat{\theta} = \arg\max_{\theta} f(\theta) \]

as follows

1. **Choose some initial guess** \( \hat{\theta}^{(0)} \)
2. **ITERATE**

\[
\hat{\theta}^{(\ell+1)} = \hat{\theta}^{(\ell)} + \lambda \left. \frac{df(\theta)}{d\theta} \right|_{\theta = \hat{\theta}^{(\ell)}}
\]

3. **UNTIL** convergence or available time is over

\( \lambda \) is a real positive number referred to as “step size” or “learning rate”
Gradient EM

\[
\frac{dh(\theta_k)}{d\theta_k} = \gamma^{-1} \sum_z g(z_1, \ldots, z_m, \hat{\theta}_k) \mu(z_1) \cdots \mu(z_m) \frac{d\log g(z_1, \ldots, z_m, \theta_k)}{d\theta_k} \, dz,
\]

\[
= \sum_z p(z_1, \ldots, z_m | \hat{\theta}_k) \frac{d\log g(z_1, \ldots, z_m, \theta_k)}{d\theta_k},
\]

\[
= E_p(z_1, \ldots, z_m | \hat{\theta}_k) \left[ \frac{d\log g(z_1, \ldots, z_m, \theta_k)}{d\theta_k} \right].
\]
Gradient EM: Properties

Theorem (Cycle-free $f_B(x, \theta)$)

Assume that a factor graph of a global function $f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta)$ is available whose subgraph $f_B(x, \theta)$ is cycle-free. The fixed points of gradient EM applied on the graph of $f(x, \theta)$ are the stationary points of $f(\theta)$.

Theorem (Cyclic $f_B(x, \theta)$)

Assume that a factor graph of a global function $f(x, \theta) \triangleq f_A(\theta)f_B(x, \theta)$ is available (whose subgraph $f_B(x, \theta)$ may be cycle-free or cyclic). The fixed points of a gradient EM algorithm applied on that factor graph are the stationary points of the function $\hat{f}(\theta)$, defined as:

$$\log \hat{f}(\theta) \triangleq \log f_A(\theta) + \int_{-\infty}^{\theta} E_{b(x|\tilde{\theta})} \left[ \nabla_\theta \log f_B(x, \tilde{\theta}) \right] d\tilde{\theta},$$

where the beliefs $b(\cdot|\theta)$ are computed by means of the sum-product messages available at convergence of the sum-product algorithm.
Problem statement

• Pseudo-noise signal $X$ is transmitted over noisy channel, resulting in the noisy signal $Y$.

• The analog circuit estimates the signal $X$ from the noisy signal $Y$.

Applications

• Spread spectrum communication systems (CDMA, UWB)

• Positioning systems (GPS)
Pseudo-random sequence $X$ generated by LFSR

$$X = \ldots, X_{k-1}, X_k, X_{k+1}, \ldots$$ with $X_k = X_{k-1} \oplus X_{k-3}$

State diagram
Pseudo-random sequence $X$ generated by LFSR

$X = [\ldots, X_{k-1}, X_k, X_{k+1}, \ldots]$ with $X_k = X_{k-1} \oplus X_{k-3}$

Representation as factor graph.
Synchronization task

Based on the noisy observation $Y$ of the sequence $X$, estimate the actual state of the source.

Approach:
Use the factor graph to define a message-passing algorithm.
Forward-only message passing on the factor graph

Interpretation:
Filtering of the sequence $Y$ with a soft version of the LFSR.
Reminder: SP for EQU and XOR-node

\[ L \triangleq \log \frac{\mu(0)}{\mu(1)} \quad \Delta = \frac{\mu(0) - \mu(1)}{\mu(0) + \mu(1)} \]

\[
\begin{array}{c}
\begin{array}{c}
\text{X} \quad \text{Z} \\
\text{Y}
\end{array}
\text{\delta}[x - y] \text{\delta}[x - z]
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( \begin{array}{c}
\mu_Z(0) \\
\mu_Z(1)
\end{array} \right) = \left( \begin{array}{c}
\mu_X(0) \mu_Y(0) \\
\mu_X(0) \mu_X(1)
\end{array} \right) \\
L_Z = L_X + L_Y \\
\Delta_Z = \frac{\Delta_X + \Delta_Y}{1 + \Delta_X \Delta_Y}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{X} \quad \text{Z} \\
\text{Y}
\end{array}
\text{\delta}[x \oplus y \oplus z]
\end{array}
\begin{array}{c}
\begin{array}{c}
\left( \begin{array}{c}
\mu_Z(0) \\
\mu_Z(1)
\end{array} \right) = \left( \begin{array}{c}
\mu_X(0) \mu_Y(0) + \mu_X(1) \mu_Y(1) \\
\mu_X(0) \mu_X(1) + \mu_X(1) \mu_Y(0)
\end{array} \right) \\
\tanh(L_Z/2) = \tanh(L_X/2) \cdot \tanh(L_Y/2) \\
\Delta_Z = \Delta_X \Delta_Y
\end{array}
\end{array}
\]
Reminder: SP for EQU and XOR-node

Signal Source

- Delay elements replaced by linear filters.
- Output of the filters \( X'_1(t) \) and \( X'_2(t) \in \mathbb{R} \).
- Introduction of threshold functions \((X_1(t), X_2(t) \text{ and } X(t) \in \{-1, +1\})\).
- Multiplication corresponds to addition modulo 2.
From Discrete-Time to Continuous-Time (2)

Synchronizing Circuit

A soft version of the signal source.
From Discrete-Time to Continuous-Time (3)

Overview

<table>
<thead>
<tr>
<th>Transmitter + Channel</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{k-3}$</td>
<td>$D$</td>
</tr>
<tr>
<td>$X_2(t)$</td>
<td>$H_2(s)$</td>
</tr>
</tbody>
</table>

$X_{k-3}$ | $D$ | $X_{k-2}$ | $D$ | $X_{k-1}$ | $D$ | $X_k$ | $Y_k$ |

$X_2(t)$ | $H_2(s)$ | $X_1(t)$ | $H_1(s)$ | $X(t)$ | $Y(t)$ |

$X_{k-3}$ | $D$ | $X_{k-2}$ | $D$ | $X_{k-1}$ | $D$ | $X_k$ | $Y_k$ |

$X_2(t)$ | $H_2(s)$ | $X_1(t)$ | $H_1(s)$ | $X(t)$ | $Y(t)$ |

$\mu_{B,k}$ | $\mu_{k-3}$ | $\mu_{k-2}$ | $\mu_{k-1}$ | $\mu_{k}$ | $\mu_{A,k}$ | $Y_k$ |

$\mu_{B}(t)$ | $\mu_{X_2}(t)$ | $\mu_{X_1}(t)$ | $\mu(t)$ | $\mu_{A}(t)$ | $Y(t)$ |

$\mu_{X_2}(t)$ | $H_2(s)$ | $\mu_{X_1}(t)$ | $H_1(s)$ | $\mu(t)$ | $\mu_{A}(t)$ | $Y(t)$ |
Demonstration System

Signals in the receiver are pseudo probability functions of the corresponding signals in the source

- **Discrete** variables represented as “pseudo-means” (Δ-representation)
  
e.g., \( \hat{E}[X(t)] = \mu_X(t) = \text{Pr}[X(t) = +1] - \text{Pr}[X(t) = -1] \)

- **Continuous** variables \((X'_1, X'_2)\)
  
The pdf for \(X'_1, X'_2(t)\) is assumed to be Gaussian \(\mathcal{N}(\mu_{X'_1, X'_2}(t), \sigma^2_{1,2})\).
  
  - Means \(\mu_{X'_1, X'_2}\) are computed
  - Variances \(\sigma^2_{1,2}\) are fixed and set manually
Demonstration System (2)

Filters

- The means $\mu_X$ and $\mu_{X_1'}$ are filtered by $H_1(s)$ and $H_2(s)$. Indeed, let $y(t) = [h \ast x](t)$, then $E[y(t)] = [h \ast E[x]](t)$.

- Remark: for computing the mean, the variance is not needed!

Soft-Thresholds:

$$
Pr[X_{1,2}(t) = +1] = Pr[X_{1,2}'(t) \geq 0]
$$

$$
Pr[X_{1,2}(t) = -1] = Pr[X_{1,2}'(t) < 0]
$$

$$
\mu_{X_{1,2}}(t) = Pr[X_{1,2}(t) = +1] - Pr[X_{1,2}(t) = -1] = \text{erf}\left(\frac{\mu_{X_{1,2}}'(t)}{\sqrt{2}\sigma_{1,2}}\right)
$$

$$
\mu_{X_{1,2}}(t) \approx \tanh\left(C\mu_{X_{1,2}}'(t)\right)
$$
Demonstration System (3)

PC with a PCI-card running Labview

\[
H_2(s) \xrightarrow{\times} H_1(s) \xrightarrow{\times} \text{AWGN} \xrightarrow{X(t)} Y(t)
\]

\[
\hat{X}(t) \xrightarrow{\times} \mu_B(t) \xrightarrow{\hat{X}(t)} \mu_A(t) \xrightarrow{Y(t)}
\]

Analog Discrete Hardware
Demonstration System (4)

The Filters

- $H_1(s)$: Butterworth Lowpass Filter, 5th order, $f_c = 1.6$ kHz
- $H_2(s)$: $4 \times H_1(s)$ in series (or $6 \times H_1(s)$ in series)
Demonstration System (5)

The Soft-Threshold Function, AWGN-Channel Estimation

Differential pair with the gain $A$ as an adjustable parameter

$$\log \frac{\mu_A(0)(t)}{\mu_A(1)(t)} = \log \frac{e^{-\frac{(Y(t)-1)^2}{2\sigma^2}}}{e^{-\frac{(Y(t)+1)^2}{2\sigma^2}}} = \frac{2Y(t)}{\sigma^2}$$
Demonstration System (6)

The Equality Constraint Gate

Forward-only EQU-Softgate

\[
\begin{align*}
I_{in1} &+ I_{in2} - H_1(s) \mu(t) &\quad \text{H1(s)} \\
I_{in1} &- I_{in2} - H_2(s) \mu_B(t) &\quad \text{H2(s)}
\end{align*}
\]

\[
\hat{X}(t) \quad \mu_B(t) \quad \mu_A(t) \quad Y(t)
\]
Demonstration System (7)

The Soft-XOR Gate

Forward-only XOR-Softgate corresponds to a “Gilbert multiplier”
Demonstration System (8)
Example of a sequence at SNR = 0 dB

Sampling rate: 50 kHz, 500 samples shown

\[ H_2(s): 4 \times H_1(s) \]
sequence-length: 361 samples

\[ H_2(s): 6 \times H_1(s) \]
1'707 samples
Measurement results

MSE vs. SNR
Results

Results for $\sigma^2_W$ ($\sigma^2_U = 0.1; \sigma^2_W = 0.001, 0.01, 0.1$)

Estimation algorithm by Sascha Korl.

Does the algorithm perform well?
Results for $\sigma_w^2$ ($\sigma_u^2 = 0.1$; $\sigma_w^2 = 0.001, 0.01, 0.1$)

Estimation algorithm by Sascha Korl.
Results for $a$ ($\sigma^2_U = 0.1$);

Standard CRB with unknown $\sigma^2_U$ and $\sigma^2_W$ (solid) for $\sigma^2_W = 0.1/0.01/0.001$;
MSE of algorithm by S. Korl (dashed);
Standard CRB for $a$ with known $\sigma^2_W = 0$. 
BCRB from information matrix of joint pdf

- The BCRBs of estimation in cycle-free graphical models can be computed efficiently by message passing.
- Messages are matrices.
- Messages are updated at each node according to specific update rules.
- The BCRBs are computed by combining those messages.
BCRB from information matrix of joint pdf (2)

Differentiable node function

\[
J_{f \mapsto y}(\mathbf{Y}) = \begin{pmatrix}
J_{x_1 \mapsto f}(X_1) + E[-\Delta_{x_1} \log f] & \ldots & E[-\Delta_{x_1} \log f] & \ldots & E[-\Delta_{x_1} \log f] \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E[-\Delta_{x_N} \log f] & \ldots & J_{x_N \mapsto f}(X_N) + E[-\Delta_{x_N} \log f] & \ldots & E[-\Delta_{x_N} \log f] \\
E[-\Delta_{y_1} \log f] & \ldots & E[-\Delta_{y_1} \log f] & \ldots & E[-\Delta_{y_N} \log f] \\
\end{pmatrix}^{-1}
\]

with \( \Delta_{x_i} \triangleq \nabla_{x_i} \nabla_{x_j}^T \)

Remarks

- Expectations \( E[\Delta_{x_i} \log f] \) supposed to be well-defined.
- They can easily be computed numerically.
- Rows and corresponding columns can be exchanged.
BCRB from information matrix of joint pdf (3)

Equality constraint node

\[ J_{f \rightarrow y} = \sum_{i=1}^{N} J_{x_i \rightarrow f} \]

Terminal node

\[ J_{f \rightarrow x} = -E[\Delta_x^x \log f] \]

PCRB

\[ J_{\text{tot}} = J_{f \rightarrow x} + J_{g \rightarrow x}. \]
Kernels from probability measures (2)

Probabilistic kernel

\[ \kappa(\hat{y}_i, \hat{y}_j) \triangleq \sum_x p(\hat{y}_i, x|\hat{\theta}) p(\hat{y}_j, x|\hat{\theta}), \]

with the parameters \( \hat{\theta} \) are obtained from the whole data set \( \mathcal{D} \), e.g., by ML-estimation:

\[ \hat{\theta}^{\text{ML, tot}} \triangleq \arg\max_\theta \prod_{i=1}^{N} p(\hat{y}_i|\theta) \]
\[ = \arg\max_\theta \prod_{i=1}^{N} \sum_x p(\hat{y}_i, x|\theta). \]
Kernels from probability measures (3)

**Product kernel [Jebara et al., 2004]**

The product-kernel is computed as follows:

\[ \kappa(\hat{y}_i, \hat{y}_j) \triangleq \sum_y p(y|\hat{y}_i)p(y|\hat{y}_j), \]

with

\[ p(y|\hat{y}_i) \triangleq \sum_x p(y|x, \hat{\theta})p(x|\hat{\theta}, \hat{y}_i), \]

where the parameters \( \hat{\theta} \) is estimated by means of the sample \( \hat{y}_i \), e.g., by ML estimation:

\[ \hat{\theta}^{ML} \triangleq \arg\max_{\theta} p(\hat{y}_i|\theta) = \arg\max_{\theta} \sum_x p(\hat{y}_i, x|\theta). \]
Feed-forward neural network
Feed-forward neural network
Feed-forward neural network (2)

\[
p(x, \xi | y, w) \triangleq \prod_{\ell=1}^{N} f(\xi^{(\ell)}, y^{(\ell)}, w) \prod_{k=1}^{m} \tilde{p}(x_k^{(\ell)} | \xi_k^{(\ell)}) \\
= \prod_{\ell=1}^{N} \delta(\xi^{(\ell)} - \xi(y^{(\ell)}, w)) \prod_{k=1}^{m} \tilde{p}(x_k^{(\ell)} | \xi_k^{(\ell)}).
\]
Feed-forward neural network: additional nodes

\[
p(x, \xi, z, y, w) \supseteq p_z(z)p_w(w) \prod_{i=1}^{N} \left[ \delta(\xi^{(i)} - \xi(z^{(i)}, w)) \right. \\
\left. \cdot \left( \prod_{j=1}^{m} \tilde{p}(x_j^{(i)}|x_j^{(i)})p(x_j^{(i)}|\xi_j^{(i)}) \right) \left( \prod_{j=1}^{n} p(y_j^{(i)}|z_j^{(i)}) \right) \right].
\]
Feed-forward neural network: pre-processing

\[
p(x, \xi, z, y, w) \triangleq p_z(z) p_w(w) \prod_{i=1}^{N} \left[ \delta(\xi^{(i)} - \xi(z^{(i)}, w)) \right. \\
\left. \cdot \left( \prod_{j=1}^{m} \tilde{p}(x_j^{(i)}|x_j^{(i)}) p(x_j^{(i)}|\xi_j^{(i)}) \right) \left( \prod_{j=1}^{n} p(y_j^{(i)}|z_j^{(i)}) \right) \right].
\]
Numerical results

Random-walk phase model with i.u.d. 4-PSK input symbols $X$

SNR = 10dB and $\sigma_W = 0.5$
Capacity of memoryless channel

**Definition**

\[
C \triangleq \sup_{p(x)} \int_x \int_y p(x)p(y|x) \log \frac{p(y|x)}{p(y)} \, dx \, dy \triangleq \sup_{p(x)} I(X; Y)
\]

with \( p(y) \triangleq \int_x p(x)p(y|x) \, dx \).

**Channel Coding Theorem**

\( C = \text{highest rate} \) at which information can be sent over the channel \( p(y|x) \) with \( \text{arbitrarily low } P_e \).
Two Blahut-Arimoto-type algorithms

Accelerated Blahut-Arimoto algorithm

\[ p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x) \exp \left( \mu^{(k)} D \left( p(y|x) \parallel p^{(k-1)}(y) \right) \right) \]

Natural-gradient based algorithm

\[ p^{(k)}(x) = p^{(k-1)}(x) \left[ 1 + \mu^{(k)} \left( D \left( p(y|x) \parallel p^{(k-1)}(y) \right) - I^{(k-1)} \right) \right] \]
Results: Gaussian channel with $E[X^2] \leq P = 1$

$$C = \frac{1}{2} \log_2 \left( 1 + \frac{P}{\sigma_0^2} \right)$$
Results: Gaussian channel with $0 \leq X \leq 1$

Model for **free-space optical** communications channel

- Transmitter: light emitting diode (LED) or laser diode (LD)
- Signal modulated on optical intensity (ON/OFF keying)
- Direct line-of-sight path is dominant
- Noise source = ambient light
- Peak power constraint due to eye safety and potential thermal skin damage
Verification

**Theorem (Karush-Kuhn-Tucker condition)**

For memoryless channel with average-power constraint \( E[X^2] \leq P \), \( p(x) \) achieves capacity \( C \) iff there exists \( \gamma \geq 0 \) such that

\[
\gamma(x^2 - P) + C - D(p(y|x)\|p(y)) \geq 0,
\]

for all \( x \), with equality for all \( x \in \text{supp}[p(x)] \).

If no average-power constraint: \( \gamma = 0 \).