Decentralized Sequential Hypothesis Testing & Change Detection

Giorgos Fellouris, Columbia University, NY, USA
George V. Moustakides, University of Patras, Greece

Outline

- Sequential hypothesis testing and SPRT
- Sequential change detection and CUSUM
- Decentralized detection and corresponding models
- Centralized schemes (points of reference)
- Decentralized detection using asynchronous random sampling
- Simulation comparisons

Sequential hypothesis testing and SPRT

Conventional binary hypothesis testing (fixed sample size): Collection of observations $\xi_1,\ldots,\xi_K$

$$\mathbb{H}_0\colon \xi_1,\ldots,\xi_K \sim f_0(\xi_1,\ldots,\xi_K);$$

$$\mathbb{H}_1\colon \xi_1,\ldots,\xi_K \sim f_1(\xi_1,\ldots,\xi_K);$$

Decision rule $D(\xi_1,\ldots,\xi_K)\in\{0,1\}$

$\mathbb{P}(D=1 \mid \mathbb{H}_1)$ (Correct decision)

$\mathbb{P}(D=1 \mid \mathbb{H}_0)$ (Type I error)

$\mathbb{P}(D=0 \mid \mathbb{H}_1)$ (Type II error)

$\mathbb{P}(D=0 \mid \mathbb{H}_0)$ (Correct decision)

Bayes and Neyman-Pearson formulation

Likelihood ratio test:

$$\frac{f_1(\xi_1,\ldots,\xi_K)}{f_0(\xi_1,\ldots,\xi_K)} \mathop{\gtreqless}_{\mathbb{H}_0}^{\mathbb{H}_1} \gamma$$

For i.i.d.:

$$u_K = \sum_{n=1}^{K} \log\left(\frac{f_1(\xi_n)}{f_0(\xi_n)}\right) \mathop{\gtreqless}_{\mathbb{H}_0}^{\mathbb{H}_1} \log(\gamma) = \gamma'$$

WAIT until K samples become available, **THEN** decide

Sequential binary hypothesis testing

Observations $\xi_1,\ldots,\xi_n,\ldots$ are supplied **sequentially**.

$$\mathbb{H}_0:\ \xi_1,\ldots,\xi_n,\ldots \quad \sim \quad f_0(\xi_n)$$

$$\mathbb{H}_1:\ \xi_1,\ldots,\xi_n,\ldots \quad \sim \quad f_1(\xi_n)$$

Time	Observations
1	ξ_1
2	ξ_1,ξ_2
$\ldots$	$\ldots$
$\mathcal{N}$	$\xi_1,\ldots,\xi_{\mathcal{N}}$

Can ξ_1 make a reliable decision?

$$\mathcal{N}(\xi_1,\ldots,\xi_n) = \{\text{stop},\text{continue}\}$$

We **stop** receiving observations

Time $\mathcal{N}$ is **RANDOM**

Decision Rule $\quad D(\xi_1,\ldots,\xi_{\mathcal{N}}) \in \{0,1\}$

WHY sequential?

For the same level of confidence with a sequential test we need, in the average, **(significantly) less samples** than a fixed sample size test, to reach a decision.

The Sequential Probability Ratio Test (SPRT)

(Wald 1947)

Changes with time

$$u_n = \sum_{k=1}^{n} \log \left(\frac{f_1(\xi_k)}{f_0(\xi_k)} \right)$$

$$u_n = u_{n-1} + \log \left(\frac{f_1(\xi_n)}{f_0(\xi_n)} \right)$$

We define **two** thresholds $A < 0 < B$

Stopping rule: $\mathcal{N} = \inf_n \{n : u_n \notin (A, B)\}$

Decision rule: $D(\xi_1, \ldots, \xi_{\mathcal{N}}) = \begin{cases} 1 & \text{if } u_{\mathcal{N}} \geq B \\ 0 & \text{if } u_{\mathcal{N}} \leq A \end{cases}$

Remarkable optimality property of SPRT

$$\min_{\mathcal{N},D} \mathbb{E}[\mathcal{N}|\mathbb{H}_0]$$

$$\mathbb{P}[D_{\mathcal{N}} = 1|\mathbb{H}_0] \leq \alpha; \qquad \mathbb{P}[D_{\mathcal{N}} = 0|\mathbb{H}_1] \leq \beta$$

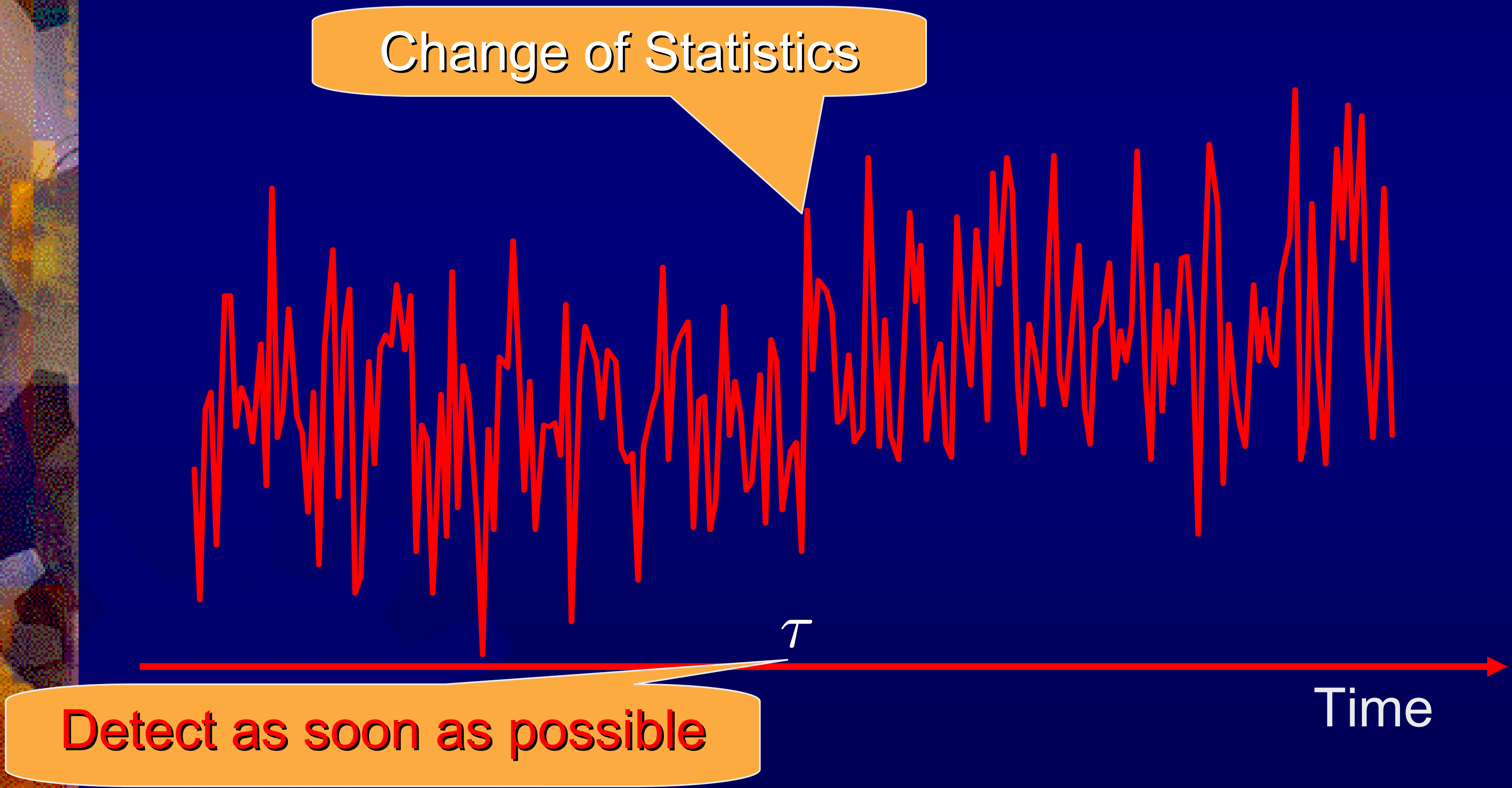
$$\min_{\mathcal{N},D} \mathbb{E}[\mathcal{N}|\mathbb{H}_1]$$

SPRT solves **BOTH** problems **simultaneously**

- Proved by Wald and Wolfowitz in 1948.

The Sequential change detection problem

Also known as the **Disorder problem** or the **Change-Point problem** or the **Quickest Detection problem.**



Applications

- Monitoring of quality of manufacturing process (1930's)
- Biomedical Engineering
- Electronic Communications
- Econometrics
- Seismology
- Speech & Image Processing
- Vibration monitoring
- Security monitoring (fraud detection)
- Spectrum monitoring
- Scene monitoring
- Network monitoring and diagnostics (router failures, intruder detection)
- Databases .....

Mathematical setup

We are observing sequentially a process $\{\xi_n\}$ with the following statistics:

$$\xi_n \quad \sim f_0 \quad \text{for } 0 < n \leqslant \tau$$
$$\quad \sim f_1 \quad \text{for } \tau < n$$

Goal: Detect the change time τ **"as soon as possible"**

- Change time τ : unknown
- Densities f_0 , f_1 : known
- At every time instant n we perform a test and decide whether there was a change or not. In the former case we stop in the latter we continue sampling.
- The test at time n must be based on the **available information up to time n** (and not on any future information).

Cumulative Sum (CUSUM) test

We recall the running log-likelihood:

$$u_n = \sum_{k=1}^{n} \log\left(\frac{f_1(\xi_k)}{f_0(\xi_k)}\right)$$

The running minimum: $m_n = \inf_{0 \leqslant s \leqslant n} u_s$.

Define the CUSUM process y_n:

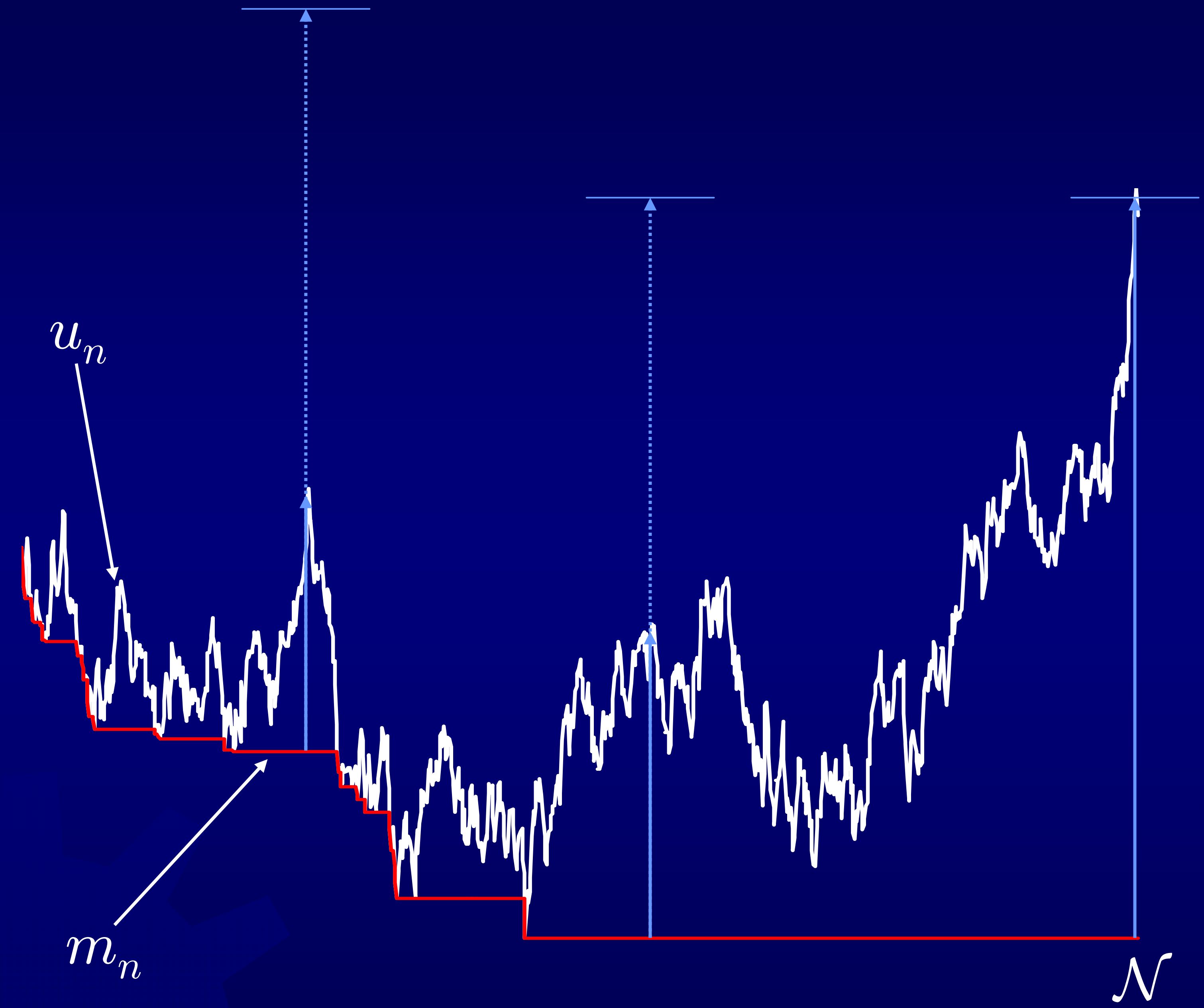
$$y_n = u_n - m_n$$

The CUSUM stopping rule:

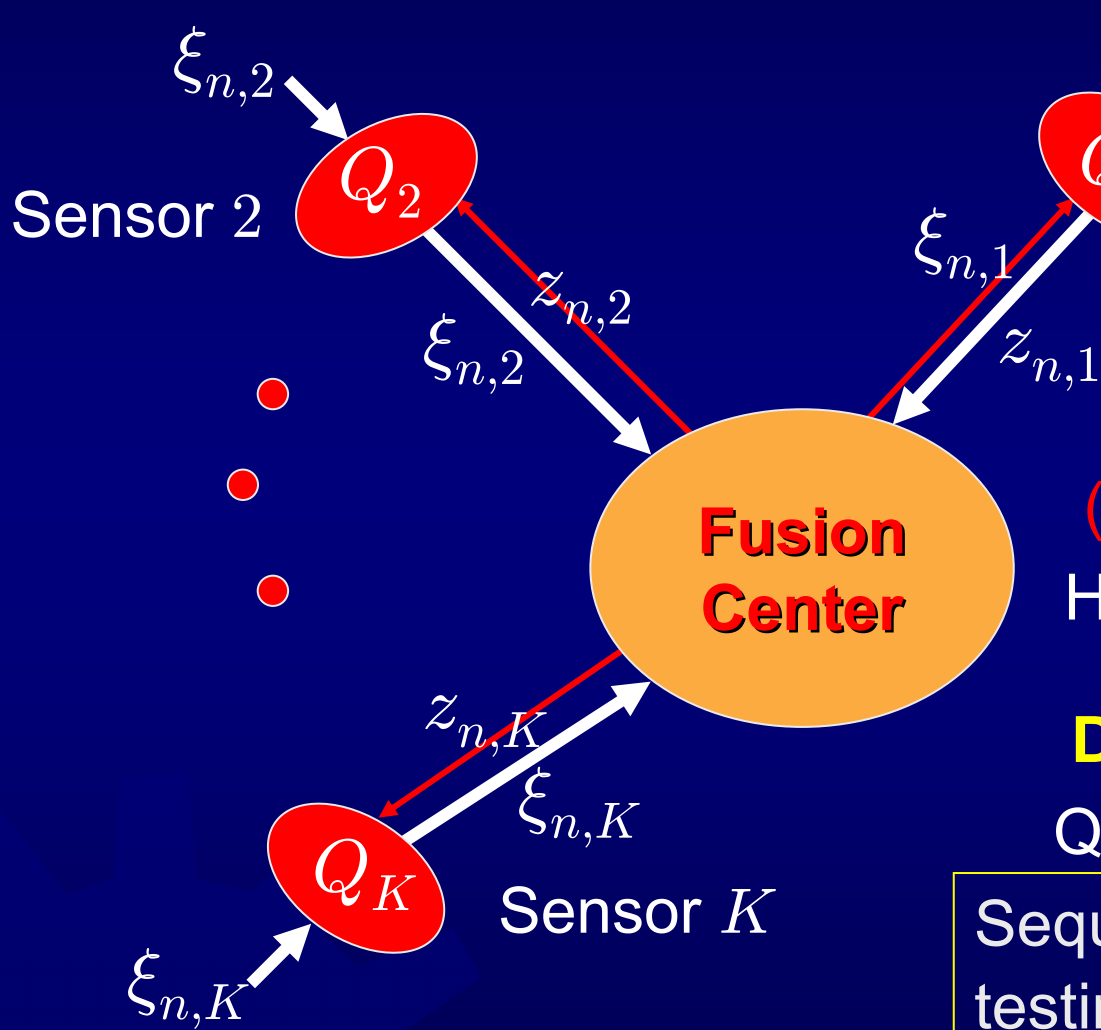
$$\mathcal{N} = \inf_n \{ \, n \colon y_n \geqslant \nu \, \}$$

We have a convenient recursion:

$$y_n = \max\left\{ y_{n-1} + \log\left(\frac{f_1(\xi_n)}{f_0(\xi_n)}\right), \; 0 \right\}$$


u_n
m_n
$\mathcal{N}$

Decentralized detection and corresponding models



Centralized Test

(point of reference)

High communication load

Decentralized Test

Quantization scheme

Sequential hypothesis testing between $f_{0,i}$ and $f_{1,i}$.

- No Local Memory:

$$z_{n,i}=Q_i(\xi_{n,i})$$

- Full Local Memory:

$$z_{n,i}=Q_i(\xi_{n,i},\xi_{n-1,i},\ldots,\xi_{1,i})$$ Mei (2008)

- Feedback with No Local Memory:

$$z_{n,i}=Q_i(\xi_{n,i},[z_{n-1,1},\ldots,z_{n-1,K}])$$

- Feedback with Partial Local Memory:

$$z_{n,i}=Q_i(\xi_{n,i},[z_{n-1,1},\ldots,z_{n-1,K}],\ldots,[z_{1,1},\ldots,z_{1,K}])$$

- Feedback with Full Local Memory:

$$z_{n,i}=Q_i(\xi_{n,i},\ldots,\xi_{1,i},[z_{n-1,1},\ldots,z_{n-1,K}],\ldots,[z_{1,1},\ldots,z_{1,K}])$$

Centralized tests

We recall that in this case the sensors send the observations $\xi_{n,i}$ to the Fusion center.

At the Fusion center we form the running log-likelihood ratio

$$\begin{aligned} u_n &= u_{n,1} + u_{n,2} + \cdots + u_{n,K} \\ u_{n,i} &= u_{n-1,i} + \log\left(\frac{f_{1,i}(\xi_{n,i})}{f_{0,i}(\xi_{n,i})}\right) \end{aligned}$$

and apply an SPRT:

Stopping rule: $\mathcal{N}_c = \inf_n \{n : u_n \notin (A_c, B_c)\}$

Decision rule: $D_c(u_1, \ldots, u_{\mathcal{N}_c}) = \begin{cases} 1 & \text{if } u_{\mathcal{N}_c} \geq B_c \\ 0 & \text{if } u_{\mathcal{N}_c} \leq A_c \end{cases}$

Remark 1:

In ALL previous detection structures it is assumed the existence of a **GLOBAL CLOCK**.
Synchronization of distant sensors with the fusion center is practically difficult (especially in sensor networks).

Remark 2:

In most practical applications the observation samples $\xi_{n,i}$ come from **canonical sampling of a continuous time process** $\xi_{t,i}$ where

$$\xi_{n,i} = \xi_{nT,i}$$

i.e. we sample $\xi_{t,i}$ at the time instances $t_n = nT$.

An even better centralized scheme

The fusion center instead of receiving the samples $\xi_{n,i}$ it can receive the **CONTINUOUS TIME PROCESSES** $\xi_{t,i}$ to form an SPRT.

$$\begin{aligned} u_t &= u_{t,1} + u_{t,2} + \cdots + u_{t,K} \\ u_{t,i} &: \text{time evolution of } u_{t,i} \text{ (sde)} \end{aligned}$$

Stopping rule: $\mathcal{T} = \inf_t \{t : u_t \notin (A, B)\}$

Decision rule: $D(u_t, 0 \le t \le \mathcal{T}) = \begin{cases} 1 & \text{if } u_{\mathcal{T}} \ge B \\ 0 & \text{if } u_{\mathcal{T}} \le A \end{cases}$

The continuous time SPRT is **better** than the discrete time SPRT due to infinite time resolution. **It constitutes the ultimate point of reference!**

Asynchronous random sampling

Let t_n^i be increasing sequence of sampling times **NOT necessarily canonical**. At these times we sample the local log-likelihood $u_{t,i}$ in the form $u_{t_n^i,i}$.

Instead of

$$u_t = u_{t,1} + u_{t,2} + \cdots + u_{t,K}$$

we propose the use of the following expression:

$$v_t = u_{t_n^1,1} + u_{t_n^2,2} + \cdots + u_{t_n^K,K}$$

Canonical sampling corresponds to: $t_n^i = nT$

Stopping rule: $\mathcal{T} = \inf_t \{t : v_t \notin (A, B)\}$

Decision rule: $D(v_1, \ldots, v_\mathcal{T}) = \begin{cases} 1 & \text{if } v_\mathcal{T} \geq B \\ 0 & \text{if } v_\mathcal{T} \leq A \end{cases}$

How do we transmit the local log-likelihoods $u_{t_n^i,i}$ from the sensors to the Fusion center ?

We observe

$$u_{t_n^i,i} = [u_{t_n^i,i} - u_{t_{n-1}^i,i}] + \cdots + [u_{t_1^i,i} - u_{t_0^i,i}]$$

To form the local log-likelihood $u_{t_n^i,i}$ at the fusion center, Sensor i needs to transmit the differences

$$[u_{t_n^i,i} - u_{t_{n-1}^i,i}]$$

We select t_n^i so that the difference $[u_{t_n^i,i} - u_{t_{n-1}^i,i}]$ takes the value A_i or B_i which are specified before hand.

What is the sampling strategy at Sensor i ?

$$t_n^i = \inf_{t > t_{n-1}^i} \left\{ t : [u_{t,i} - u_{t_{n-1}^i,i}] \text{ hits } A_i \text{ or } B_i \right\}$$

- Repeated SPRTs at each sensor.

Communication with the Fusion center

- If the difference is equal to B_i send 1 if it is equal to A_i send 0.
- The Fusion center knows that 1 corresponds to B_i and 0 to A_i and can therefore update the partial log-likelihood then compute v_t and perform the test by comparing v_t to A and B.

- Every time new information arrives at the Fusion center (even from one sensor) the Fusion center updates

$$v_t = u_{t_n^1,1} + u_{t_n^2,2} + \cdots + u_{t_n^K,K}$$

and performs the test.

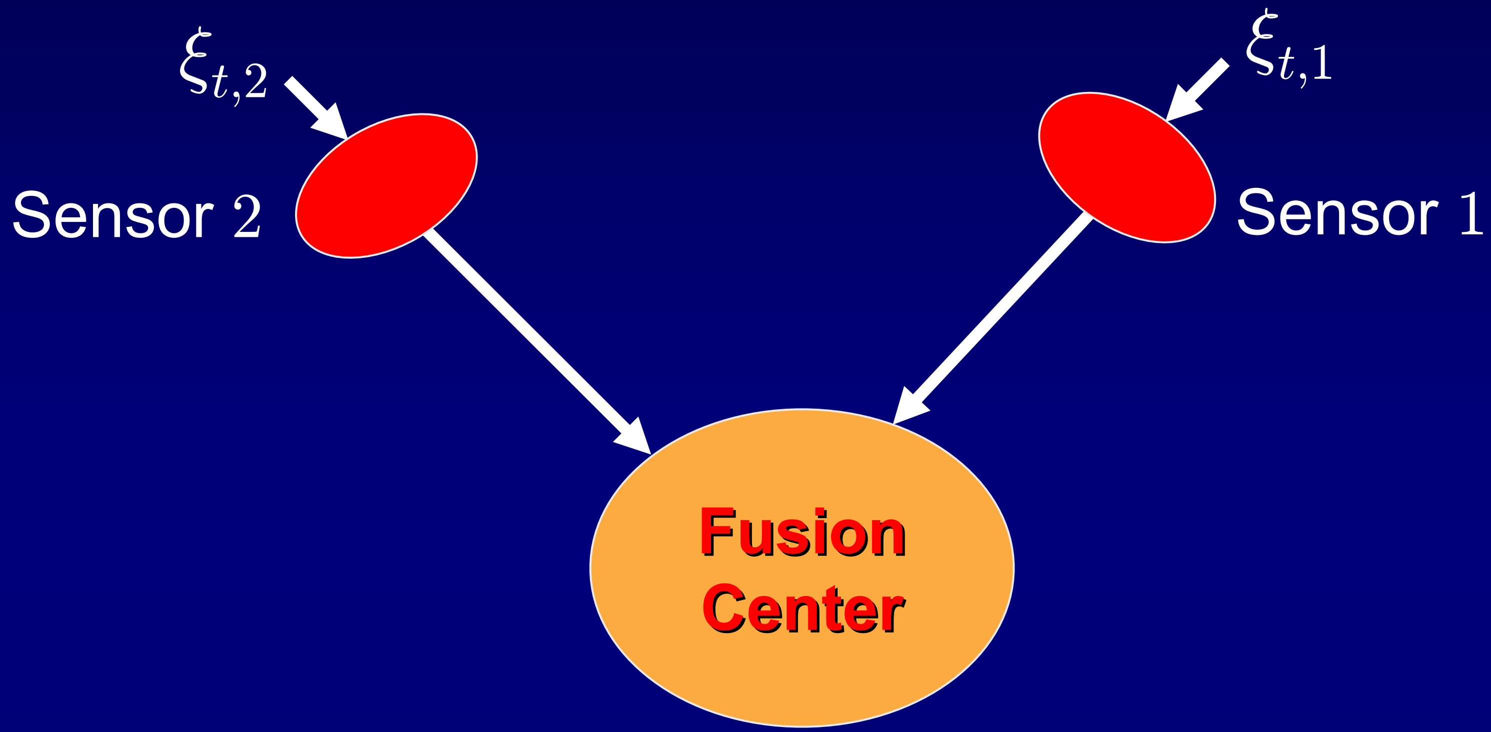
- Communication is Asynchronous and Random!!!

How do we select the local thresholds A_i , B_i ?

We can specify a communication rate between sensors and Fusion center.

If the sensors must communicate, **in the average**, every T time units, then this condition specifies completely the thresholds. We must select the thresholds so that the "average detection delay" of the local SPRTs is equal to T.

Simulations

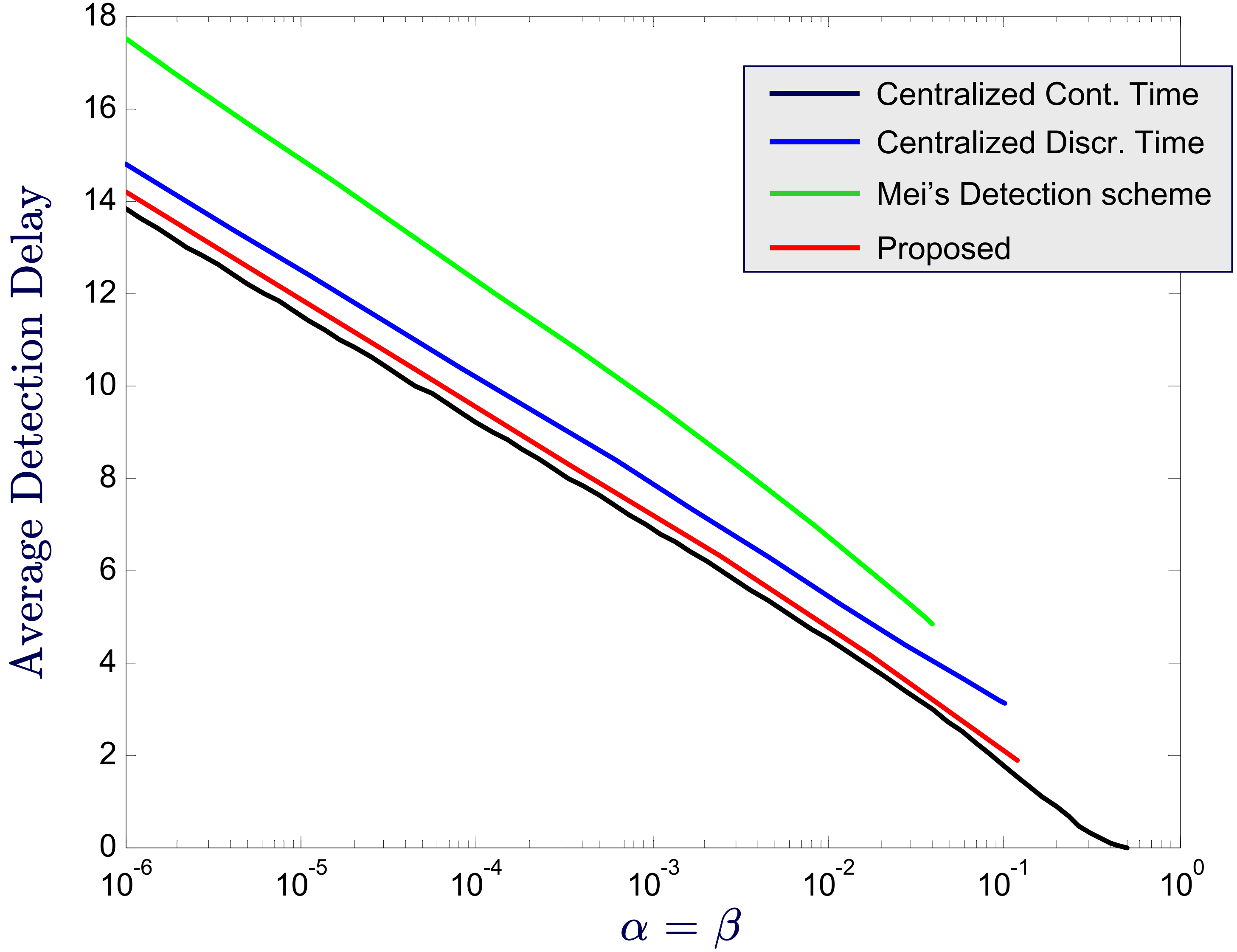


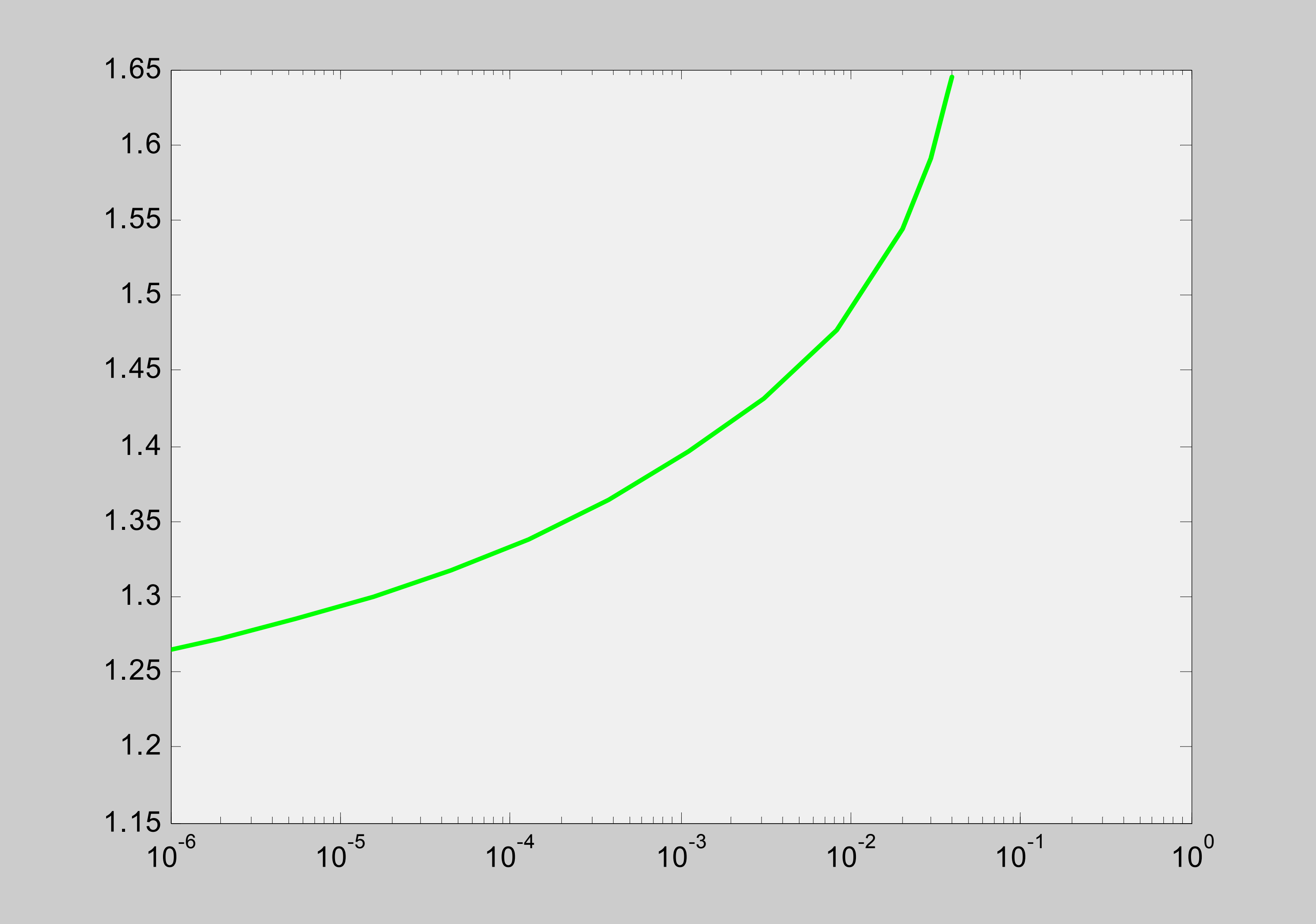
$\xi_{t,i}$: both standard Brownian Motions
$\mathbb{H}_0$: drift $= 0$; $\mathbb{H}_1$: drift $= 1$.

Average communication period $T=3$.

Canonical sampling generates:

$\mathbb{H}_0$: Gaussian$(0,3)$; $\mathbb{H}_1$: Gaussian$(3,3)$


Average Detection Delay
18
16
14
12
10
8
6
4
2
0
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
$\alpha = \beta$
Centralized Cont. Time
Centralized Discr. Time
Mei's Detection scheme
Proposed


1.65
1.6
1.55
1.5
1.45
1.4
1.35
1.3
1.25
1.2
1.15
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}

FIN

Merci de votre attention