Exploiting Self-Similarity For Change-Detection

Giacomo Boracchi and Manuel Roveri

Dipartimento di Elettronica, Informazione,

e Bioingegneria, Politecnico di Milano

giacomo.boracchi@polimi.it

IJCNN 2014, Beijing

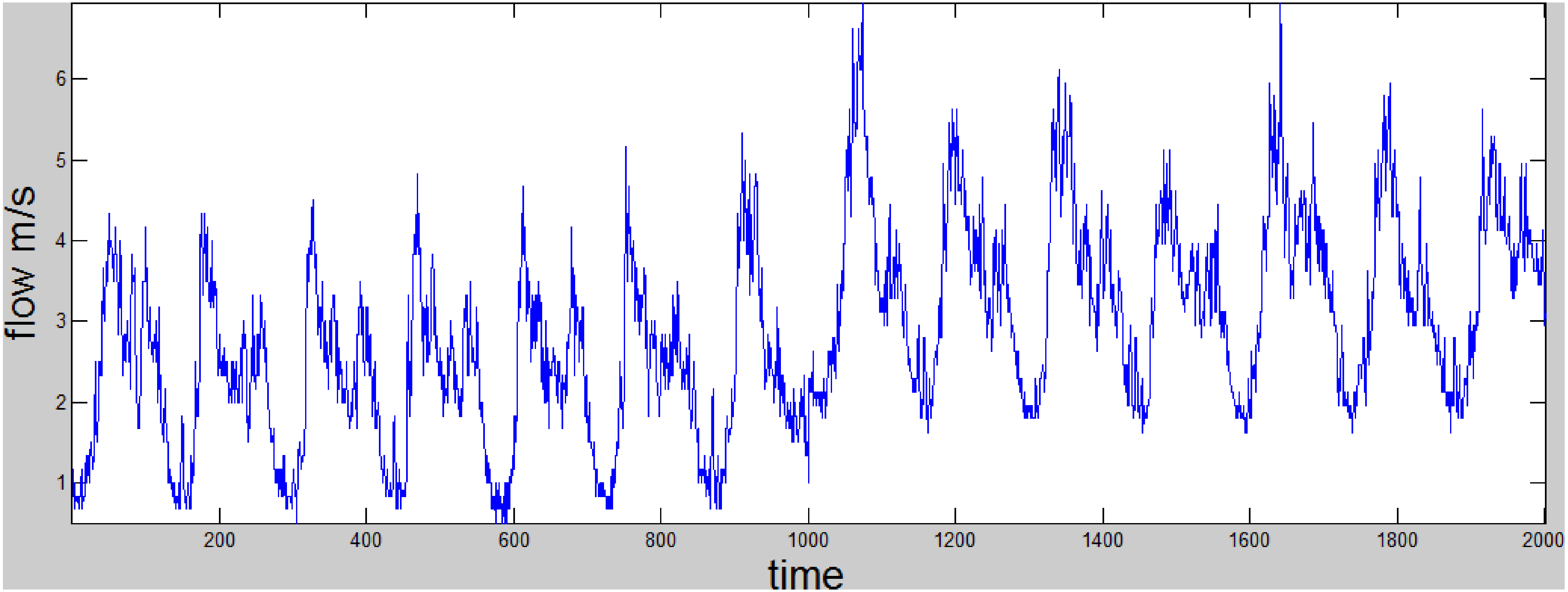
11 July 2014

THE MOTIVATING IDEA

…and our contribution

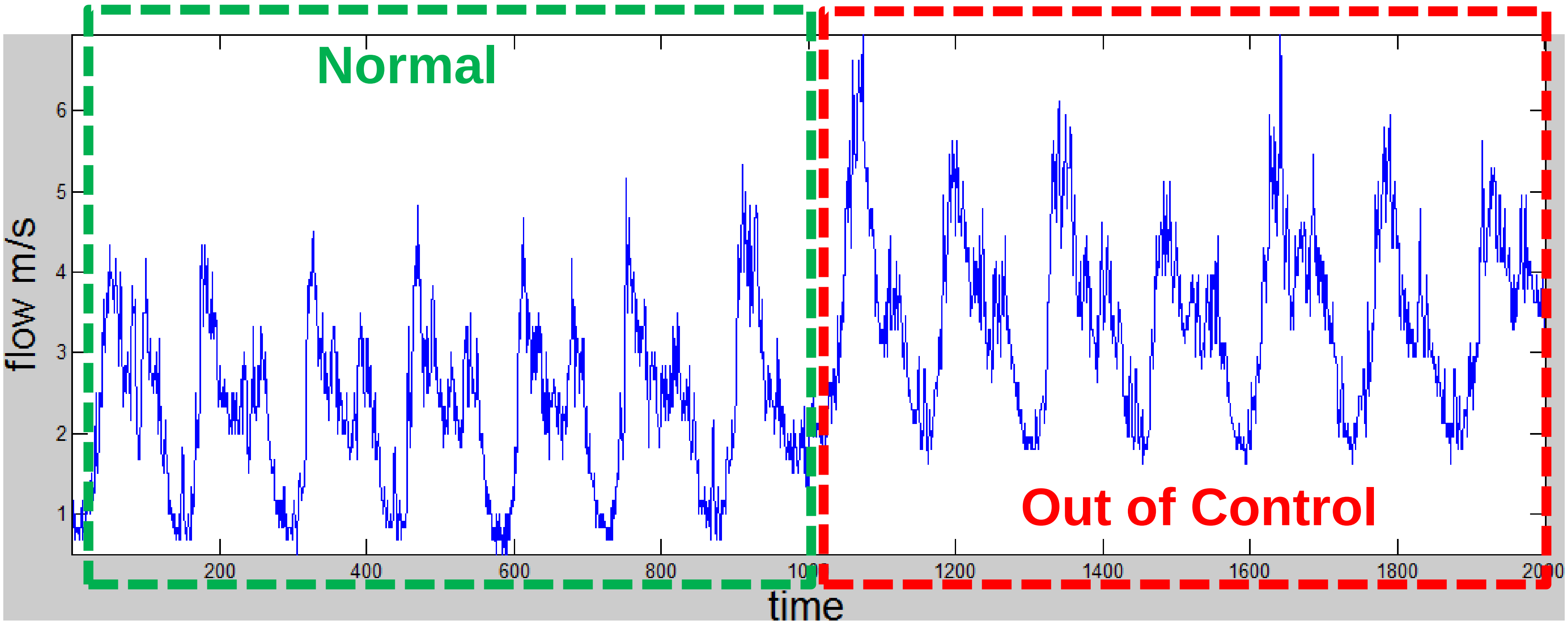
Motivating Idea

- Detecting changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes



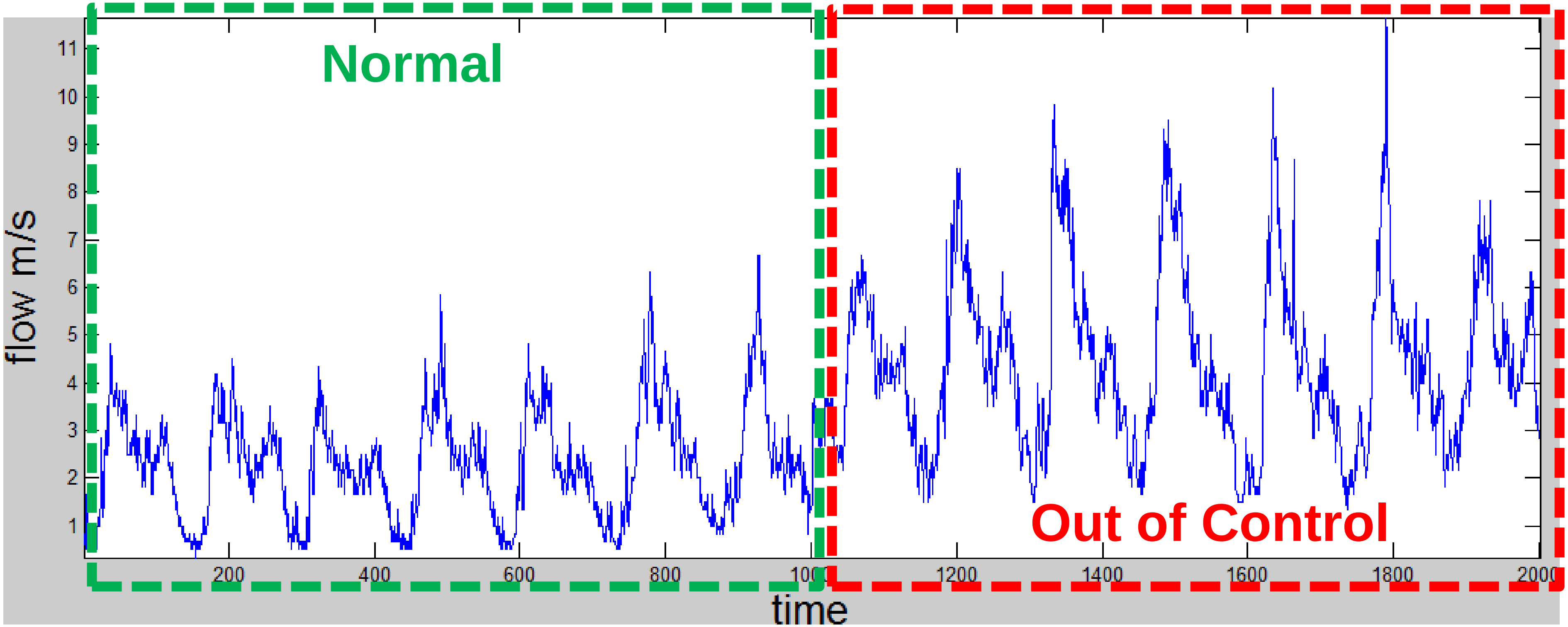
Motivating Idea

- Identifying changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes



Motivating Idea

- Identifying changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes



Motivating Idea (cnt)

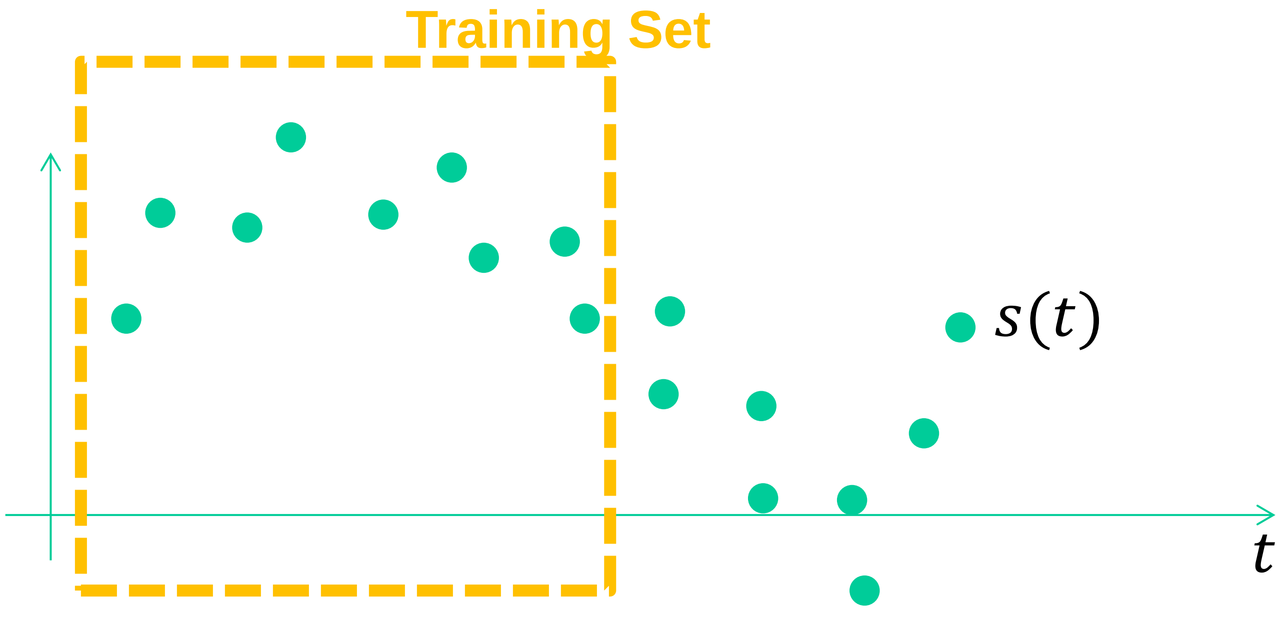
1. Fit a model / build a predictor for the time series

$$f_{\hat{\theta}}(t)$$

2. For each incoming samples compute the residuals

$$e(t) = s(t) - f_{\hat{\theta}}(t)$$

3. Monitor the stationarity of the residuals



Motivating Idea (cnt)

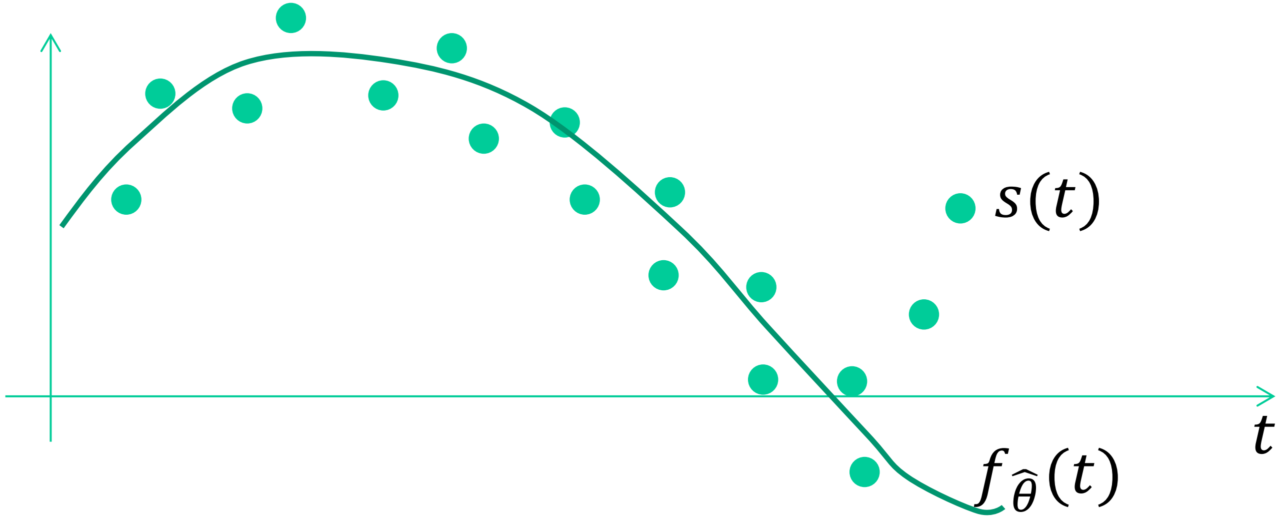
1. Fit a model / build a predictor for the time series

$$f_{\hat{\theta}}(t)$$

2. For each incoming samples compute the residuals

$$e(t) = s(t) - f_{\hat{\theta}}(t)$$

3. Monitor the stationarity of the residuals



Motivating Idea (cnt)

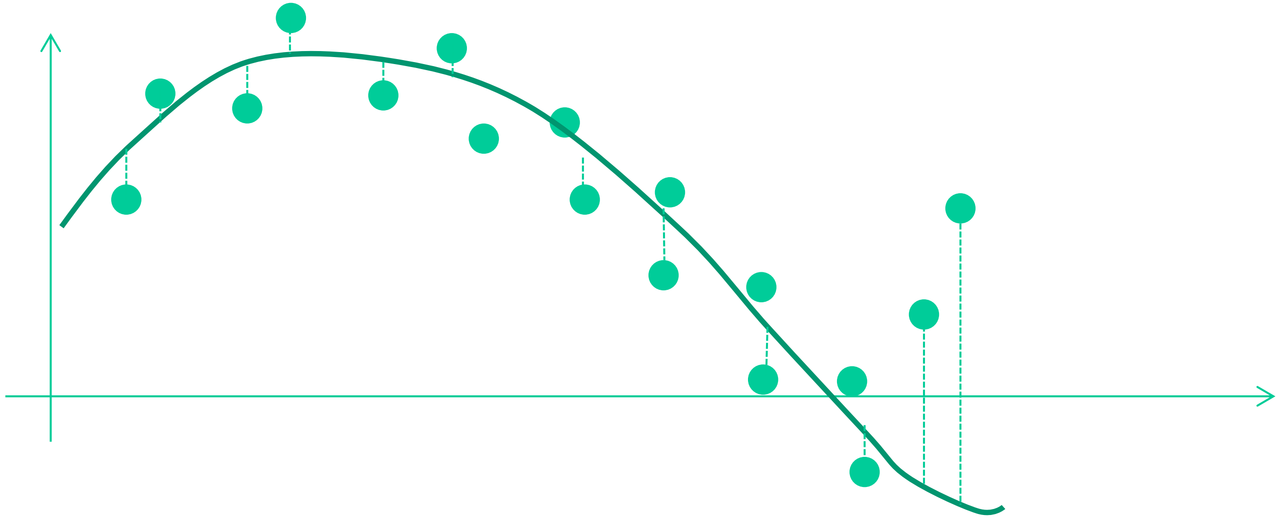
1. Fit a model / build a predictor for the time series

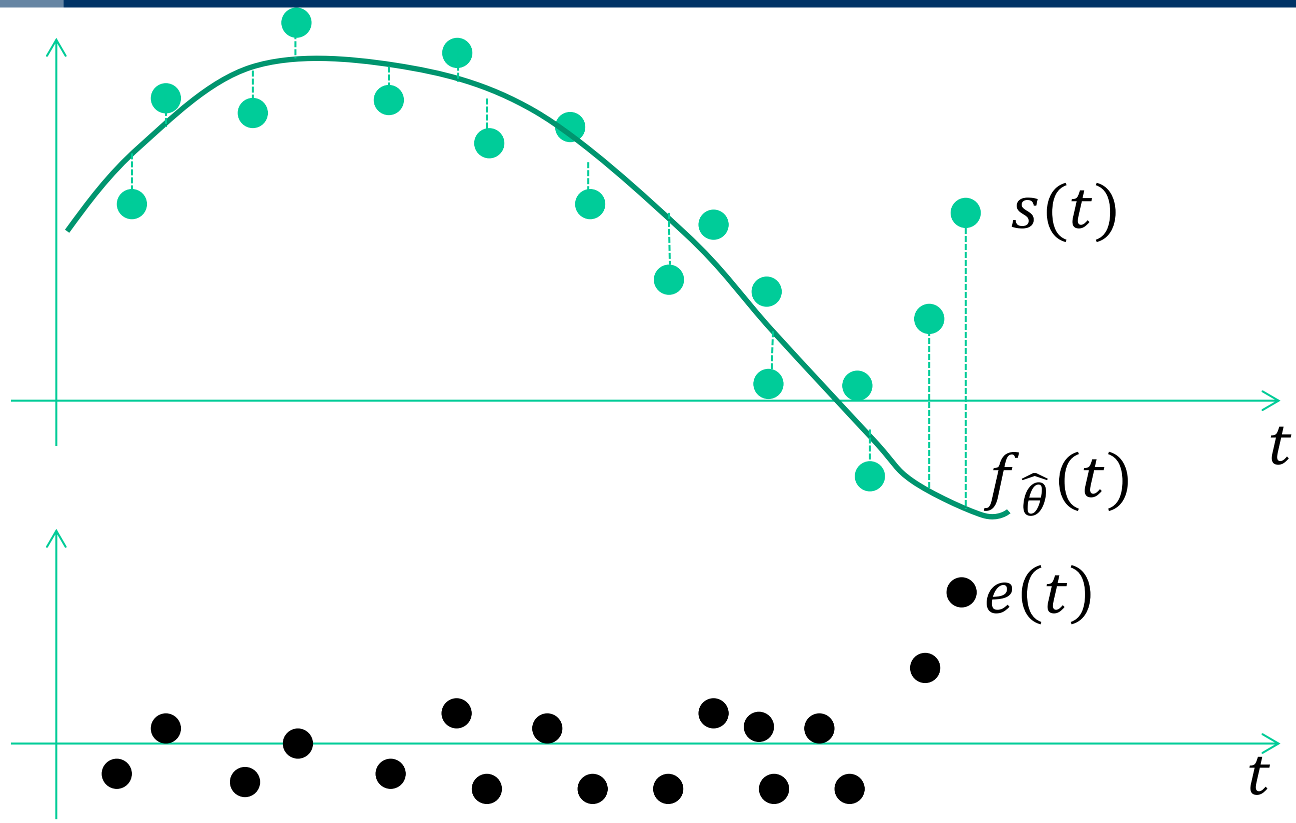
$$f_{\hat{\theta}}(t)$$

2. For each incoming samples compute the residuals

$$e(t) = s(t) - f_{\hat{\theta}}(t)$$

3. Monitor the stationarity of the residuals

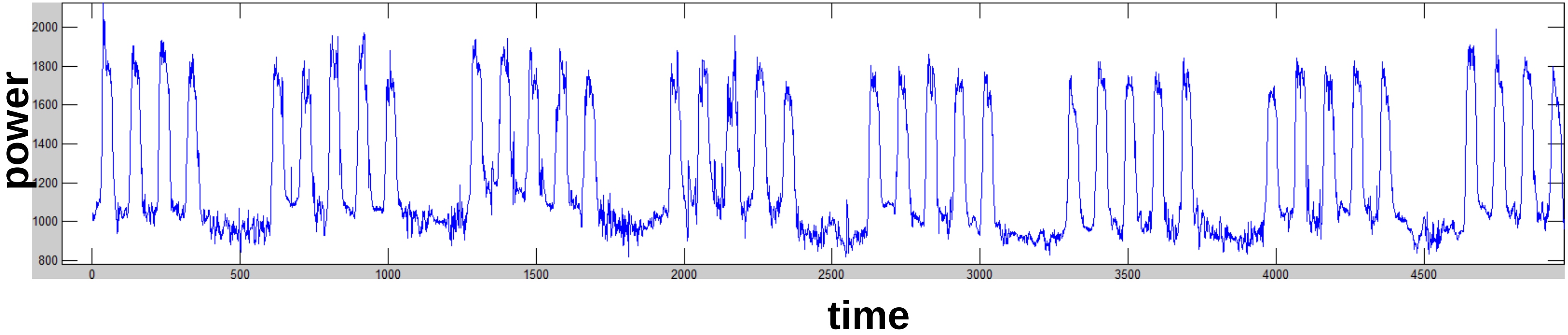




- Unfortunately, it is sometimes difficult to:
  - find good model family (i.e., f)
  - reliably fit this model (i.e., estimating $\hat{\theta}$)

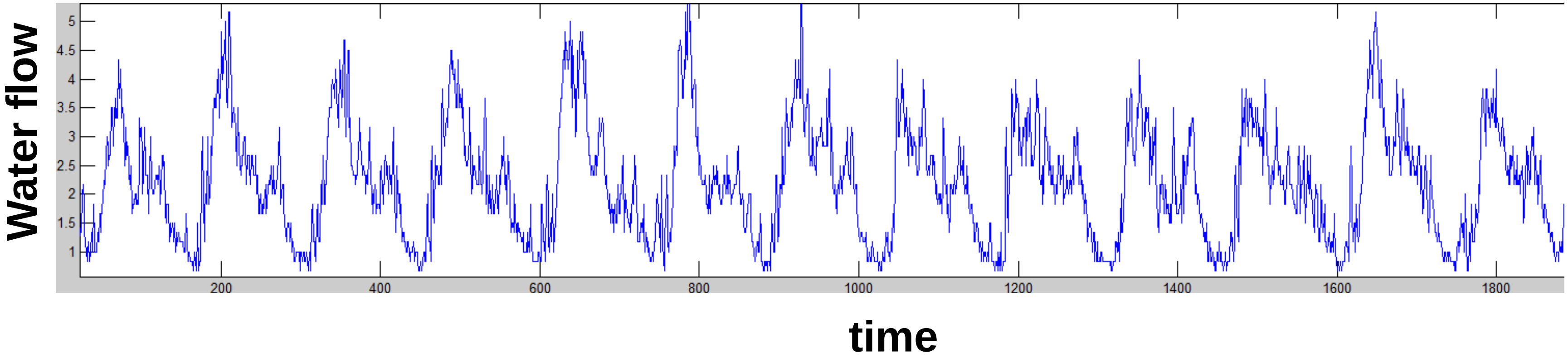
Motivating Idea (cnt.)

- Often, signals and time series are redundant and exhibit self-similarity

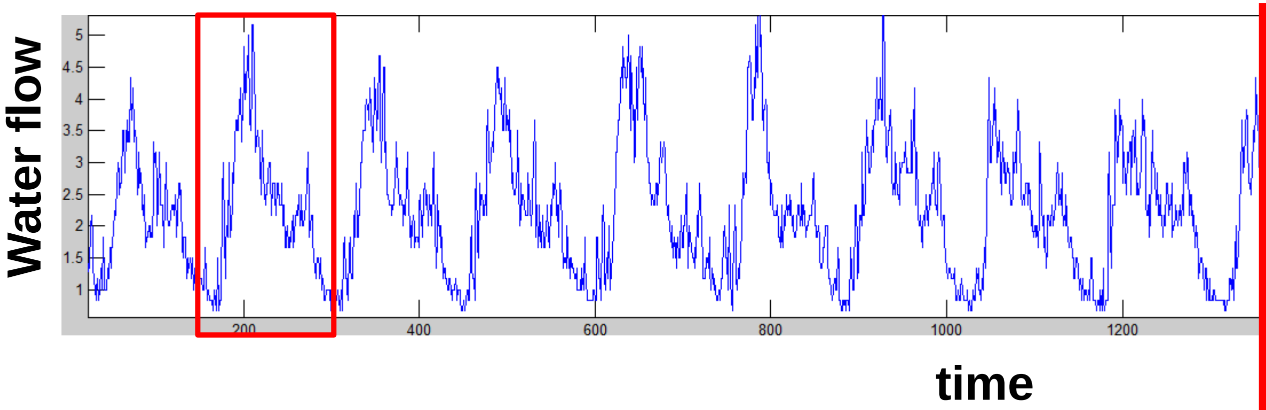


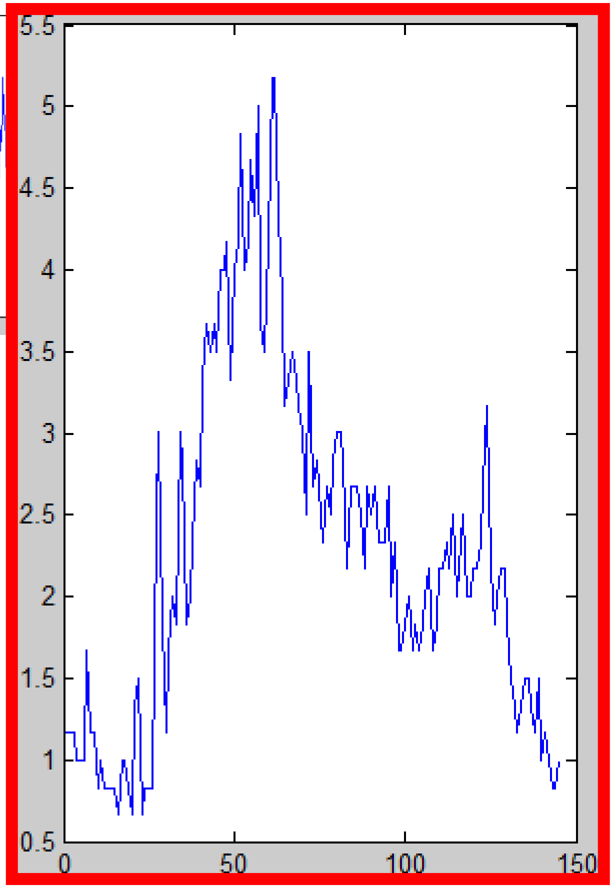
Motivating Idea (cnt.)

- Often, signals and time series are redundant and exhibit self-similarity



- Often, signals and time series are redundant and exhibit self-similarity

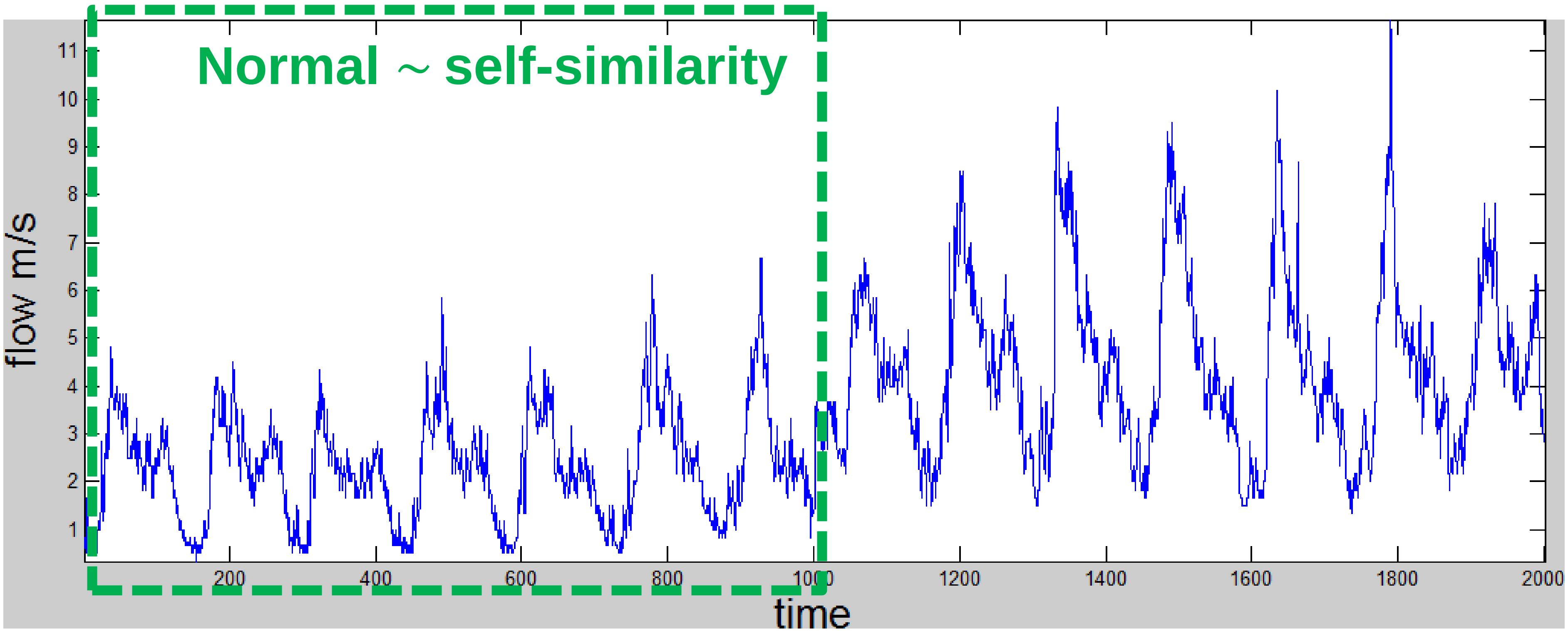




In case of water consumption, the periodicity is due to inhabitants' customary habits

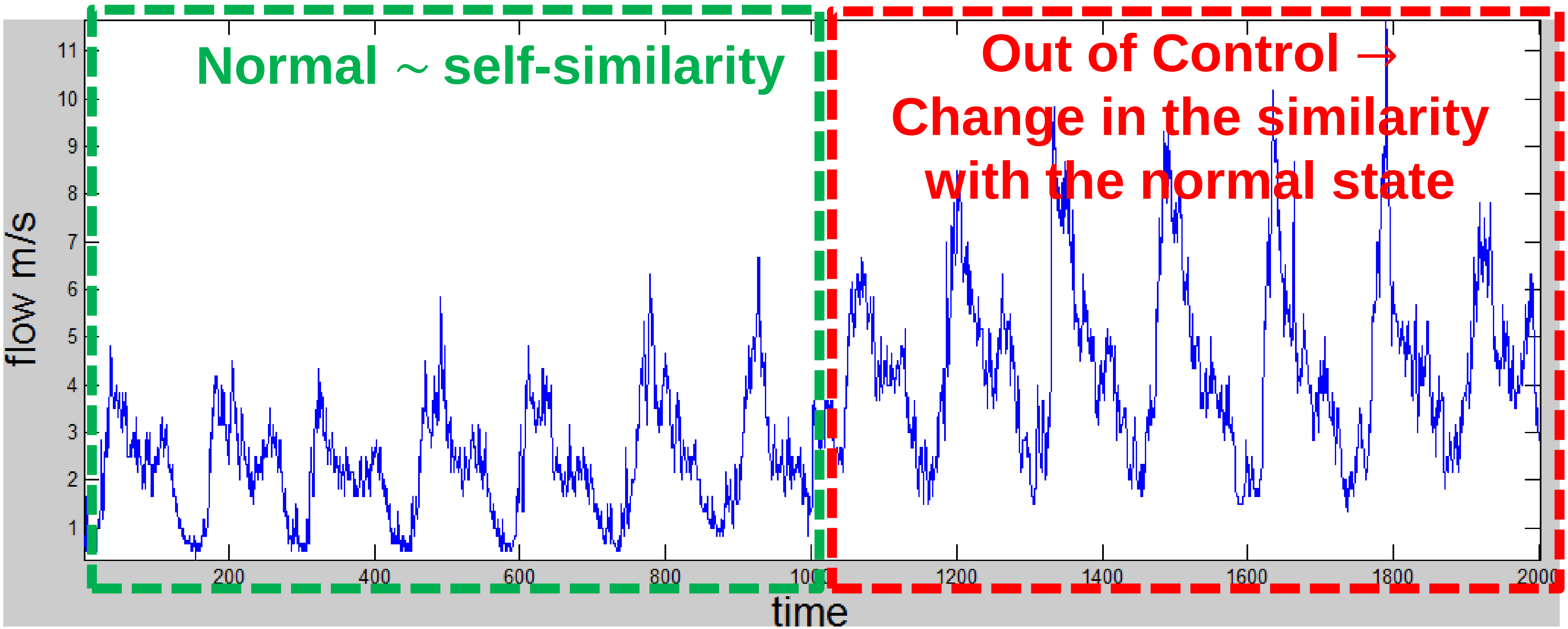
Motivating Idea (cnt.)

- The process, in its **normal state**, exhibits some structure which is **redundant**, **self similar**, repeated



Motivating Idea (cnt.)

- The process, in its **normal state**, exhibits some structure which is **redundant, self similar**, repeated
- **Out of of control states** instead exhibit **patterns** that are instead **different** from the normal state.
  - **Degree of similarity with the normal state changes**



Our Contribution

- We present a Change Detection Test (CDT) to sequentially monitor time series that uses self-similarity to
  - Characterize normal state of the process
  - Detect any departure from normal condition

Outline

- Self similarity as a powerful prior
- Problem Formulation
- Proposed Solution
  - Change Indicator
  - Search Regions
  - The Algorithm
- Experiments
- Dicussion and Conclusions

SELF SIMILARITY

A powerful prior in signal-image processing

Self similarity is a powerful prior

- Texture completion
- Denoising (Regression)
- Inpainting (Reconstruction)

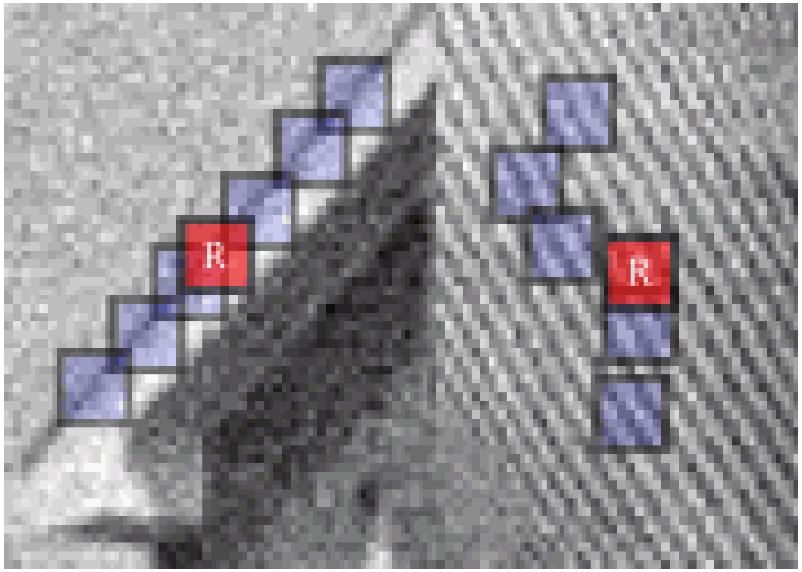


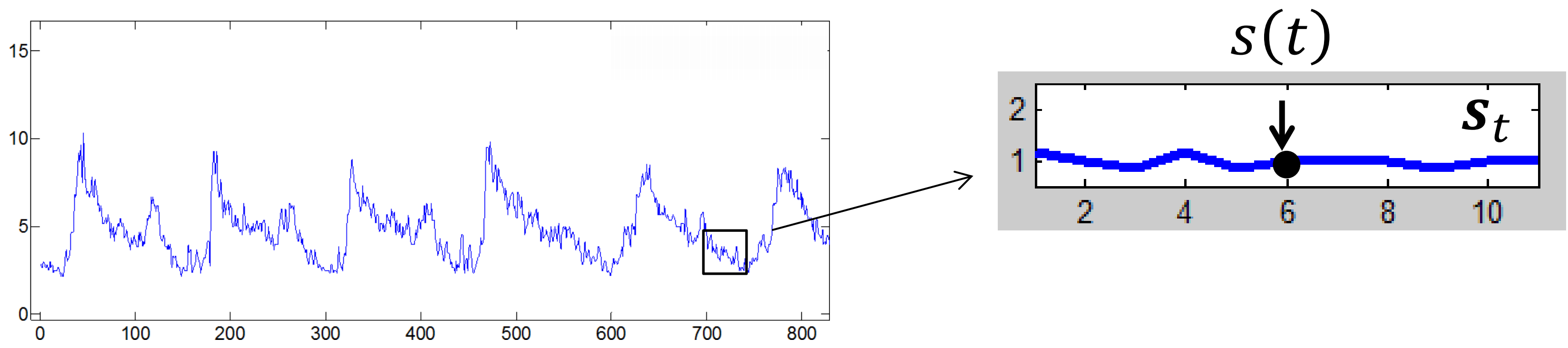
Image courtesy of Alessandro Foi
http://www.cs.tut.fi/~foi/

- Never used for discriminative purposes in a sequential detection task

Self similarity as a powerful prior

- Self-similarity is measured patch-wise
- We consider 1D datastreams $\{s(\tau), \tau = 1, \dots\},\ s(\tau) \in \mathbb{R}$
- We define a patch centered at t having size ν as

$$\boldsymbol{s}_t = \{s(t-\nu), \dots, s(t), \dots, s(t+\nu)\}$$



- The distance between two patches is the ℓ_2 norm of their difference $\|\boldsymbol{s}_t - \boldsymbol{s}_\tau\|_2 = \sqrt{\sum_{i=-\nu}^{\nu} \big(s(t+i) - s(\tau+i)\big)^2}$

PROBLEM FORMULATION

Problem Formulation

- Let us assume that a **process** S generates a **datastream** $\{s(\tau), \tau = 1, \ldots\},\ s(\tau) \in \mathbb{R}$
  - S has to exhibit self similarity in the normal state
- We say that there is a **change** at T^* if S **permenently shifts** from the **normal** state into an **out of control** state.
- We consider out of control states that **modifies self-similarity** of S
  - the **patches** from $\{s(\tau), \tau = 1, \ldots, T^*\}$ are **not similar to patches** from $\{s(\tau), \tau = T^* + 1, \ldots\}$.
- **Goal**: Given a *normal training sequence* TS, detect changes analyzing, in a sequential and online manner $\{s(\tau), \tau = L + 1, \ldots\}$

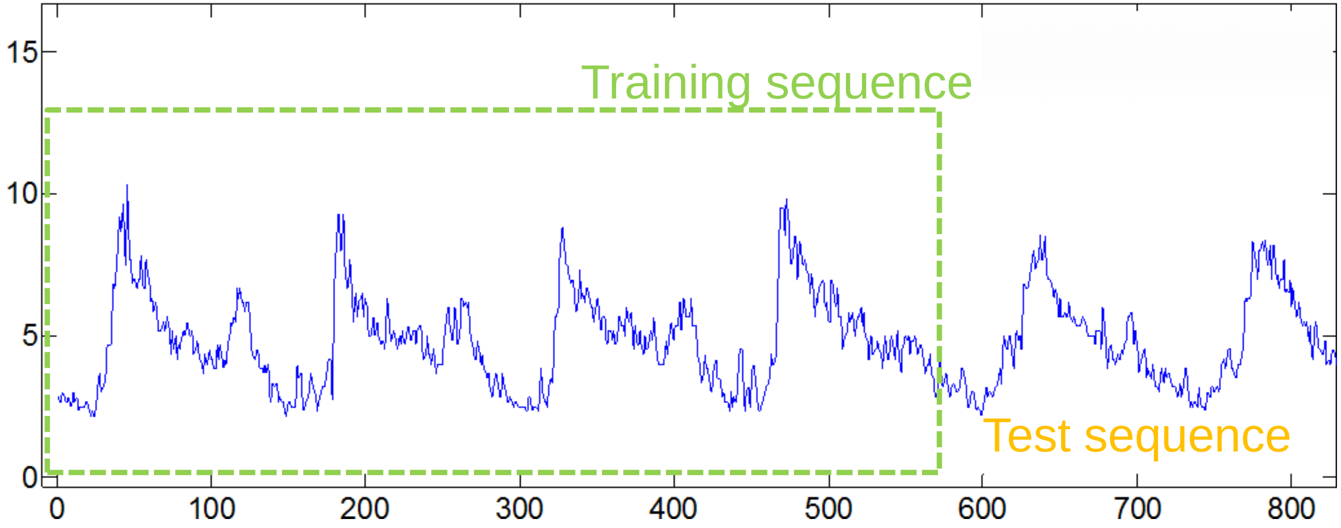
PROPOSED SOLUTION

Exploiting self similarity for performing change-detection

The Proposed Solution

- We build a training set for *normal patches*

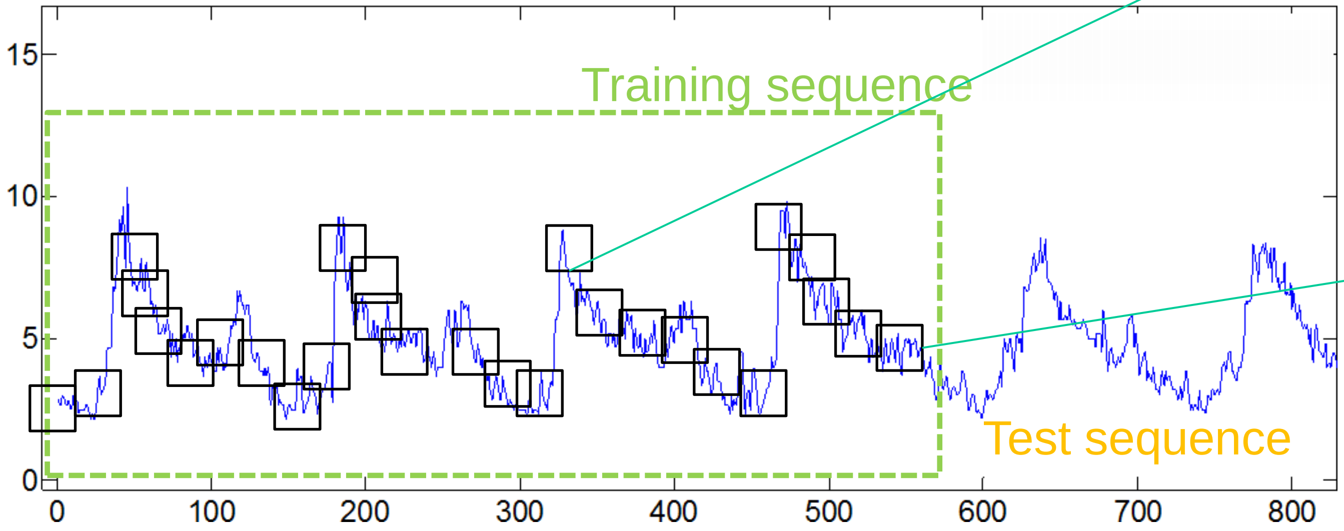
$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \dots, M - \nu\}$$

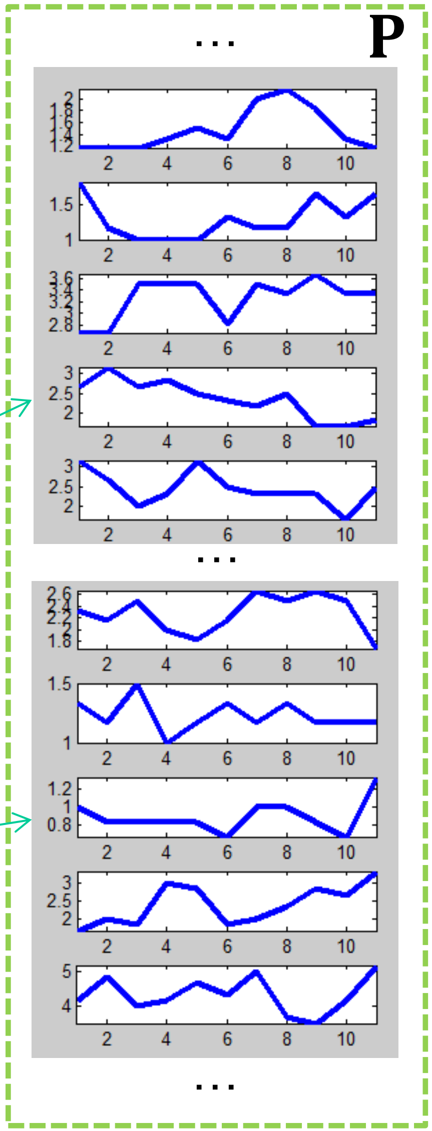


The Proposed Solution

- We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \dots, M - \nu\}$$





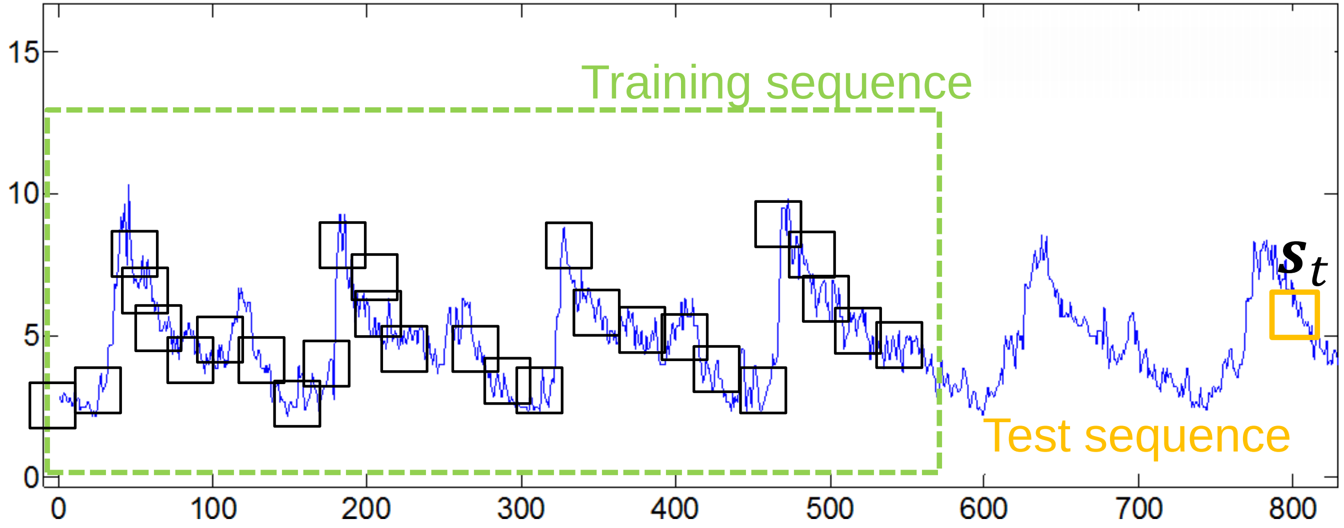
The Proposed Solution

- We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \ldots, M - \nu\}$$

- **Intuition:**

$$\begin{cases} \exists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t < T^* & \textbf{Normal} \\ \nexists\ \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t \geq T^* & \textbf{Out of Control} \end{cases}$$

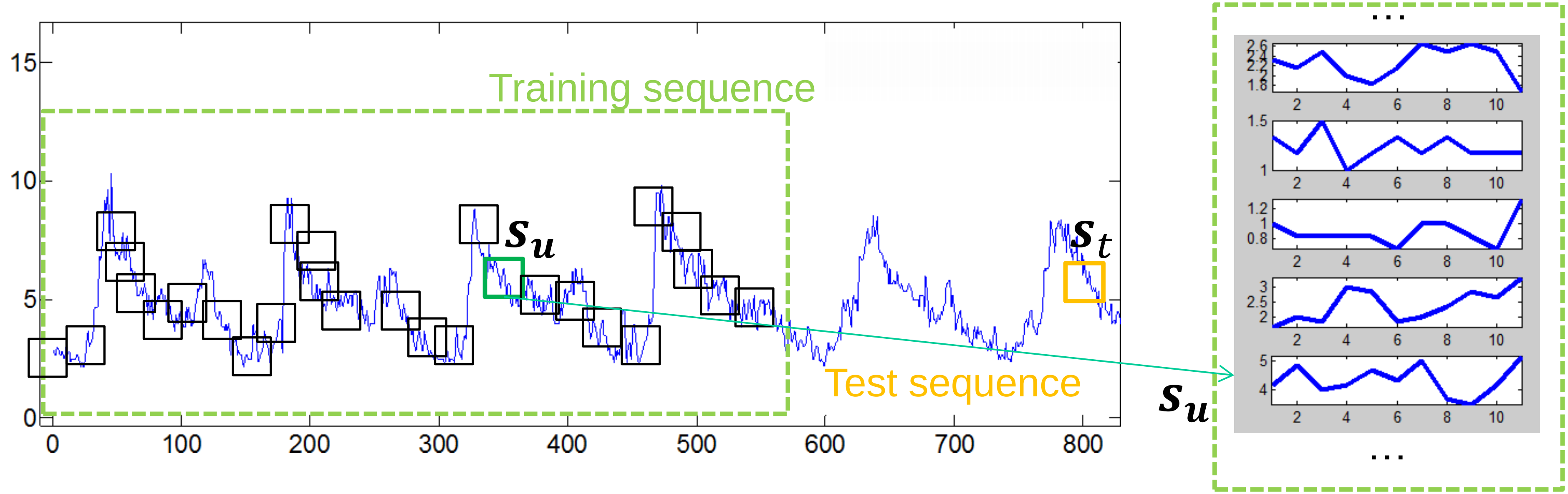


- We build a training set for *normal patches*

$$\mathbf{P} = \{\boldsymbol{s}_t, t = \nu, \dots, M - \nu\}$$

- **Intuition:**

$$\begin{cases} \exists\, \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t < T^* & \textbf{Normal} \\ \nexists\, \boldsymbol{s}_u \in \mathbf{P} \text{ similar to } \boldsymbol{s}_t, \forall t \geq T^* & \textbf{Out of Control} \end{cases}$$



The Change Indicator

- We need to construct a **change indicator** $x(t)$ to **quantitatively assess** our intuition
- We expect the change indicator $x(t)$ to satisfy
  - $\{x(t), t < T^*\}$ should be **i.i.d**. realizations of an **unknown random variable**
  - $\{x(t), t \geq T^*\}$ should come from a **different distribution**, not necessarily being i.i.d.
- Out of **control states** can be **detected** as **changes in the distribution of x**
  - We can use any statistical process control technique

The Change Indicator (cnt.)

- The compute the change indicator $x(t)$ we first identify the most similar patch in $\mathbf{P}$ to $\boldsymbol{s}_t$.
- We define $\pi(\cdot)$ as the map that associate to t the location $\pi(t)$ of the patch $\mathbf{P}$ of that is most similar to $\boldsymbol{s}_t$

$$\pi(t) = \underset{\tau=\nu,\dots,M-\nu}{\operatorname{argmin}} ||\boldsymbol{s}_t - \boldsymbol{s}_\tau||_2$$

the values of $\pi(\cdot)$ can be com

- $x(t)$ is the difference between the centers of $\boldsymbol{s}_t$ and $\boldsymbol{s}_{\pi(t)}$

$$x(t) = s(t) - s(\pi(t))$$

The Change Indicator (cnt.)

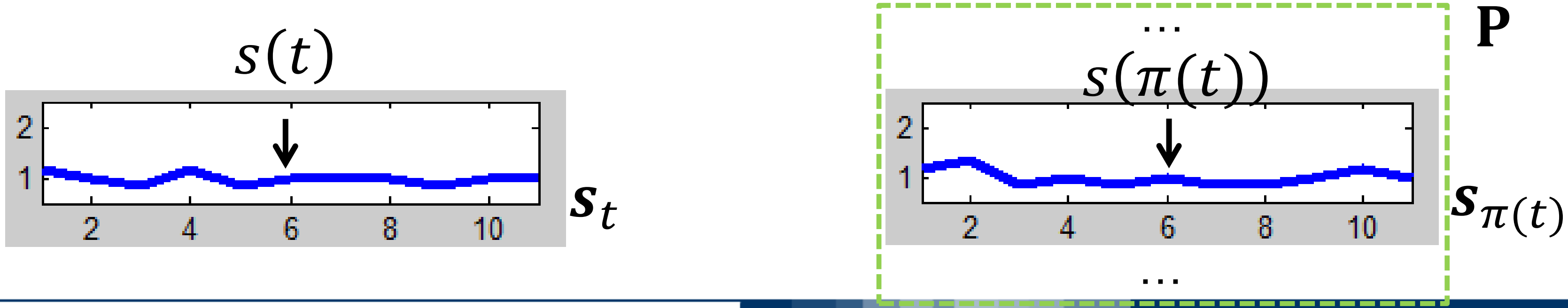
- The compute the change indicator $x(t)$ we first identify the most similar patch in $\mathbf{P}$ to $\boldsymbol{s}_t$.
- We define $\pi(\cdot)$ as the map that associate to t the location $\pi(t)$ of the patch $\mathbf{P}$ of that is most similar to $\boldsymbol{s}_t$

$$\pi(t) = \underset{\tau=\nu,\ldots,M-\nu}{\operatorname{argmin}} ||\boldsymbol{s}_t - \boldsymbol{s}_\tau||_2$$

the values of $\pi(\cdot)$ can be com

- $x(t)$ is the difference between the centers of $\boldsymbol{s}_t$ and $\boldsymbol{s}_{\pi(t)}$

$$x(t) = s(t) - s(\pi(t))$$



In Ideal Conditions

- Assume perfect matches in normal conditions, i.e., $\boldsymbol{s}_t$ and $\boldsymbol{s}_{\pi(t)}$ differ only because of noise
- Then, $\forall t < T^*$

$$x(t) = s(t) - s\big(\pi(\tau)\big) = \eta$$

  i.i.d random variable and $\mathrm{E}[\eta] = 0$
- While $\forall t > T^*$, we do not expect perfect matches: some bias appears in $x(t)$, namely $\mathrm{E}[x(t)] \neq 0$
- In this case, it is possible to detect changes in $x(t)$ by means of any sequential CDT.

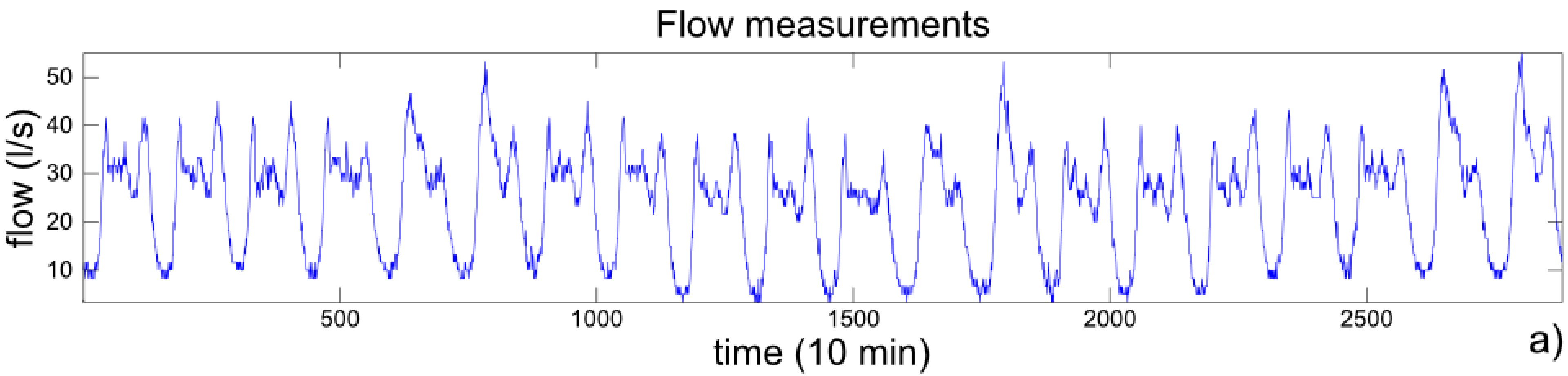
In the Real Life

- In the real life, perfect matches are rare
  - Patches do not differ only because of noise
  - Noise affects also the association function $\pi(\cdot)$
- However, there is an **experimental evidence** that **patch similarity** well correlates with the **similarity between** their **central pixels**
  - This is the idea behind *Non Local Means filter* [Buades et al 2005], which introduced a well established paradigm in signal/image processing

[Buades et al 2005] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," Multiscale Modeling Simulation, vol. 4, no. 2, p. 490, 2005.

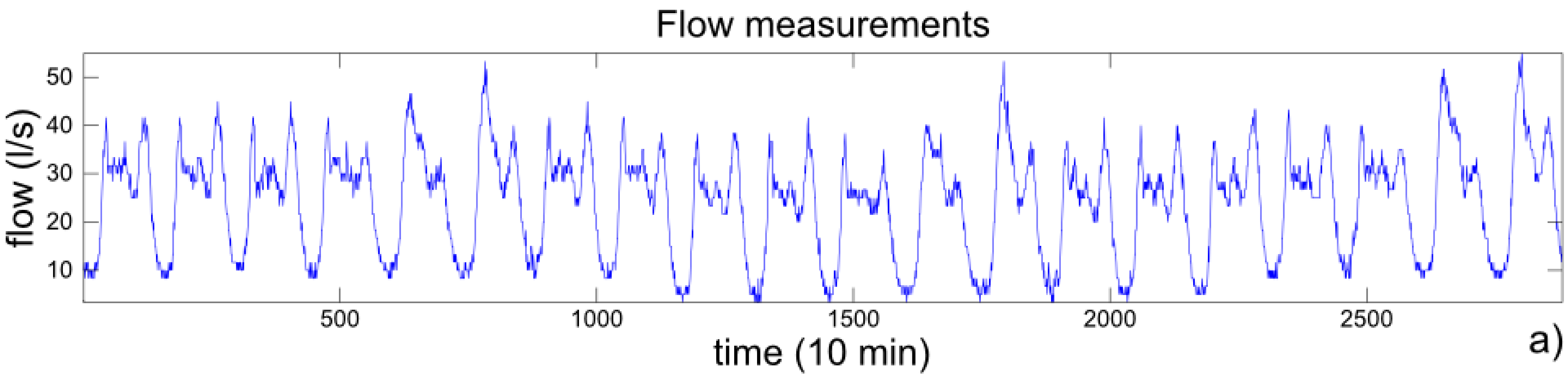
The Search Region

- Often, self similarity is due to **periodic or cyclic** nature of the phenomenon under monitoring



The Search Region

- Often, self similarity is due to **periodic or cyclic** nature of the phenomenon under monitoring



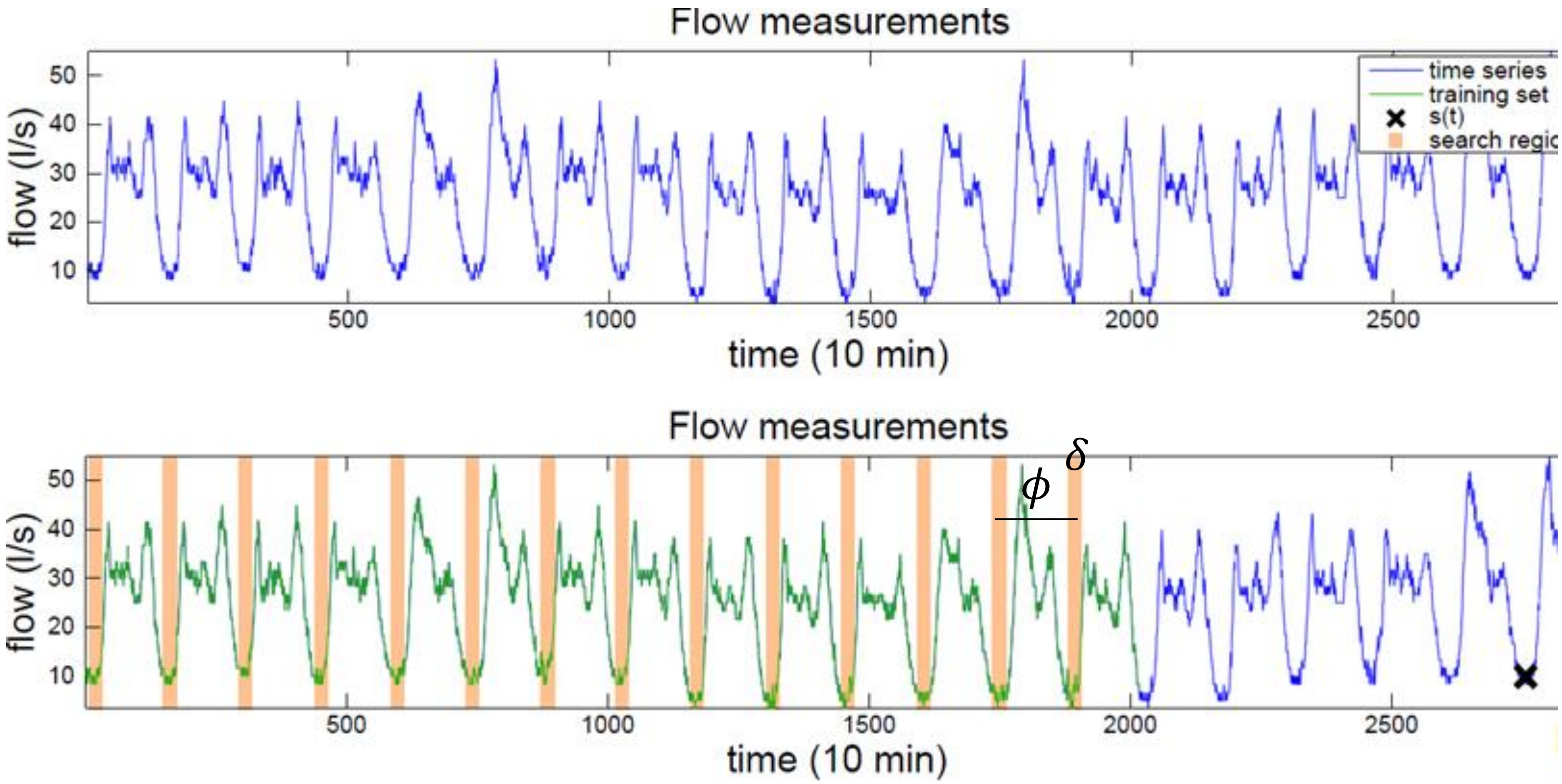
- Search for similar patches should be constrained to the same time instants accross the periods
  - This determines what «out of control states» are
  - This improves computational complexity

The Search Region

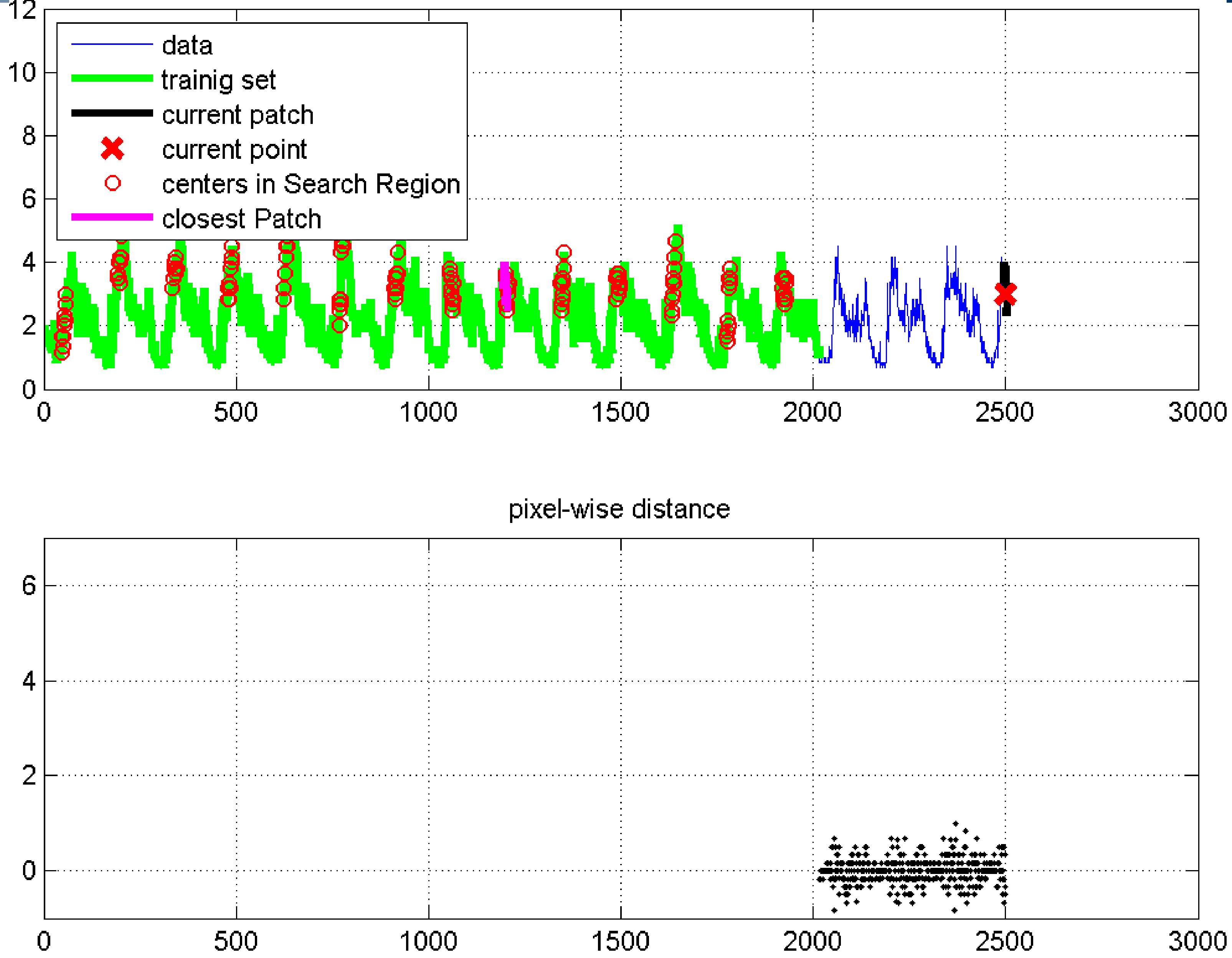
- The function $\pi(\cdot)$ is thus defined

$$\pi(t) = \underset{\tau \in R_{\phi,t,\delta}}{\operatorname{argmin}} || \boldsymbol{s}_t - \boldsymbol{s}_\tau ||_2$$

- Being, $R_{\phi,t,\delta} = \bigcup_i \{\tau, \ \text{s.t.} \ |t_0 + i\phi - \tau| < \delta \}$

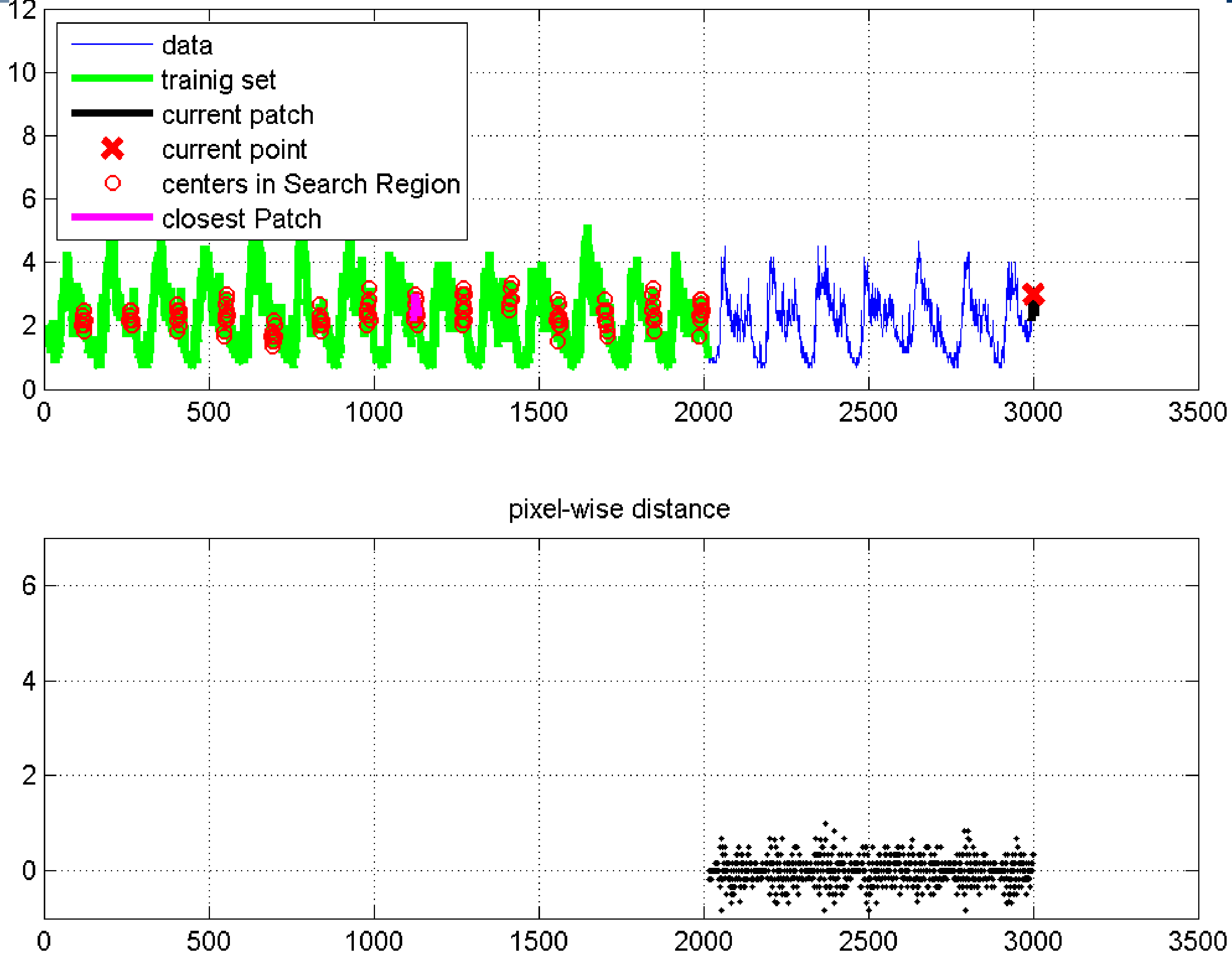


An example

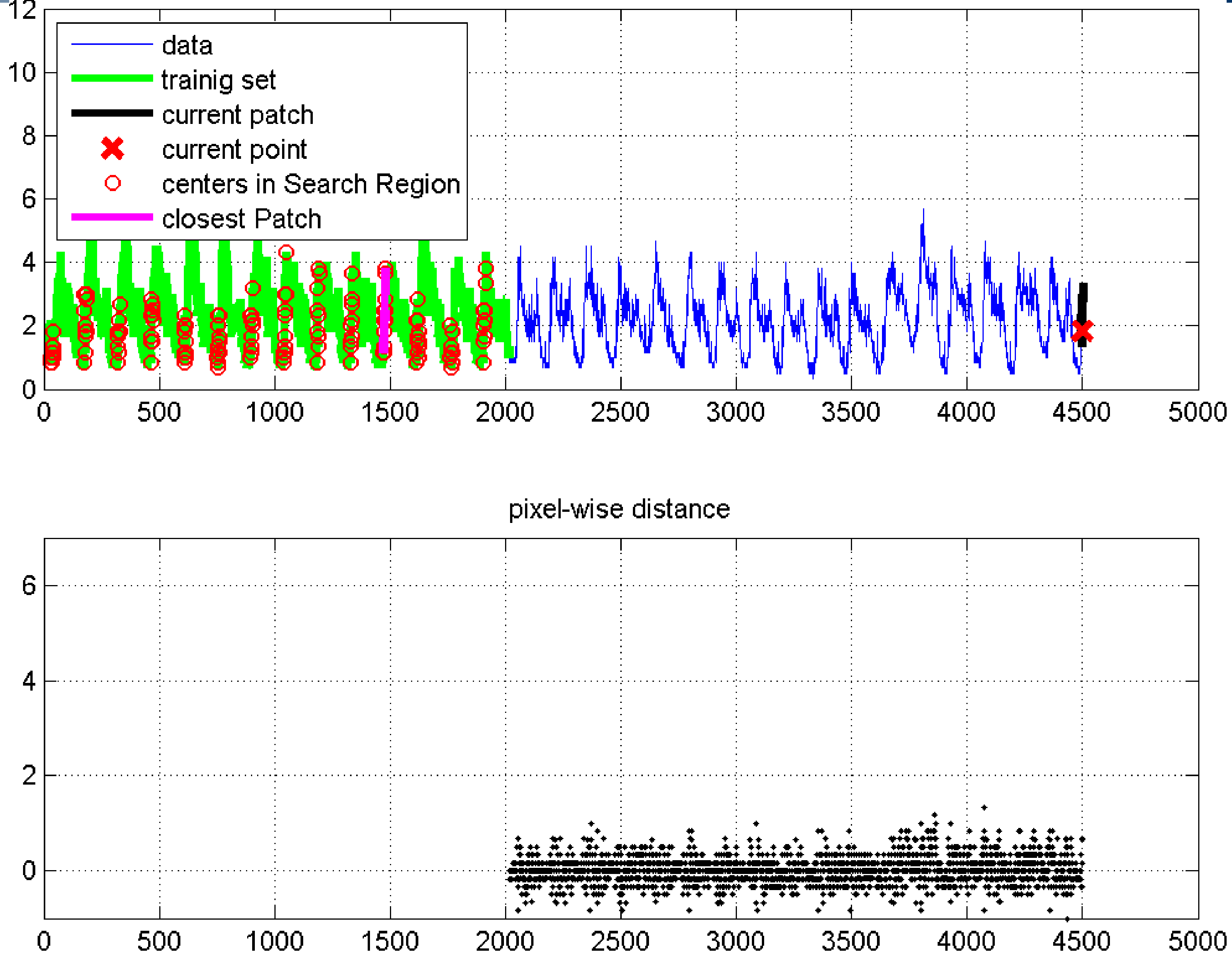


data
trainig set
current patch
current point
centers in Search Region
closest Patch
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
pixel-wise distance
0
2
4
6
0
500
1000
1500
2000
2500
3000

An example

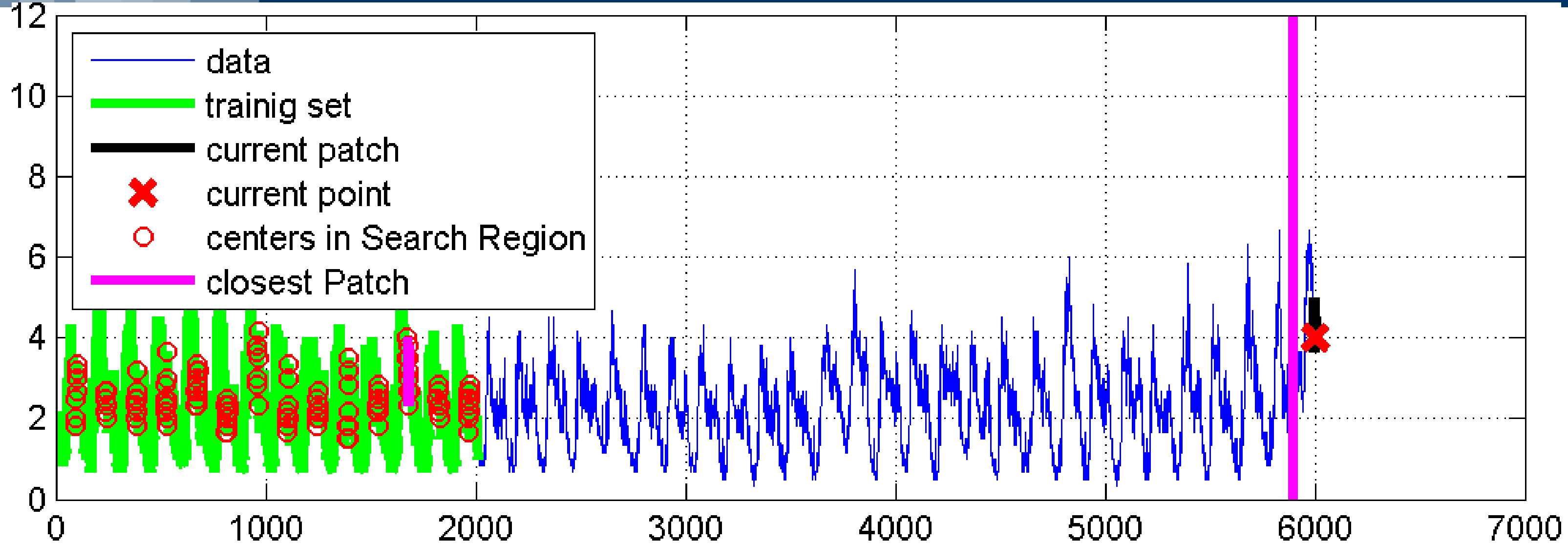

data
trainig set
current patch
current point
centers in Search Region
closest Patch
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
3500
pixel-wise distance
0
2
4
6
0
500
1000
1500
2000
2500
3000
3500

An example



data
trainig set
current patch
current point
centers in Search Region
closest Patch
0
2
4
6
8
10
12
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
pixel-wise distance
0
2
4
6

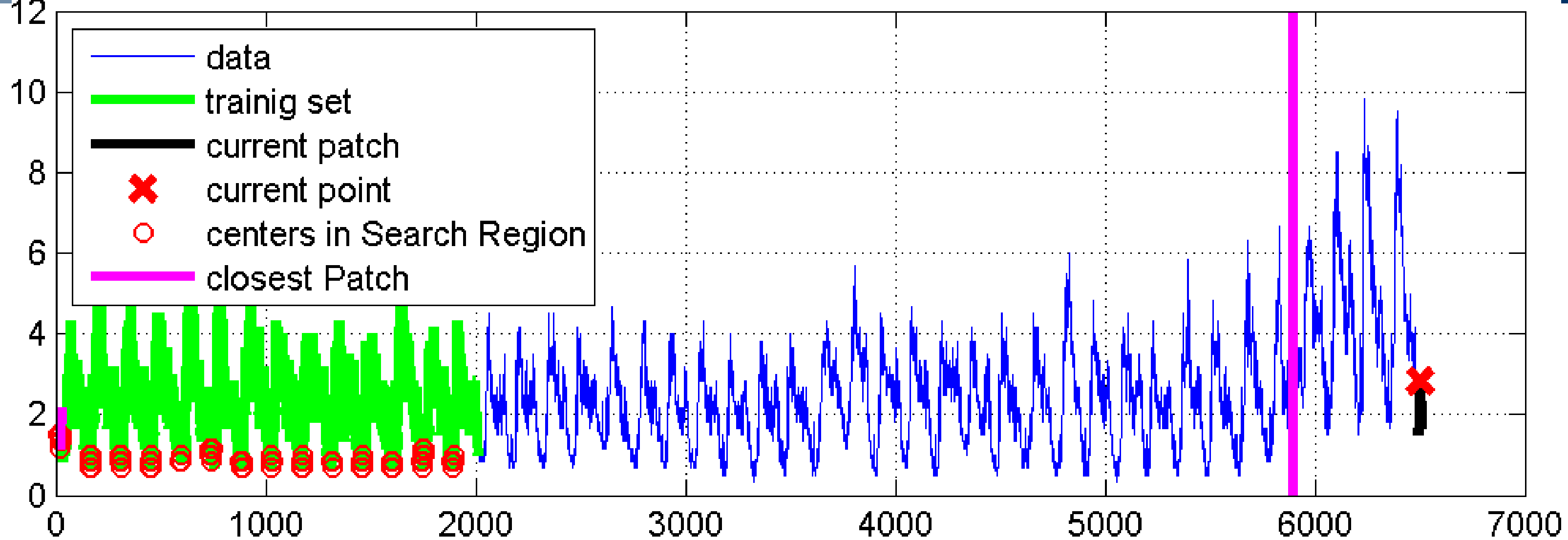
An example



data
trainig set
current patch
current point
centers in Search Region
closest Patch
12
10
8
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000

pixel-wise distance
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000

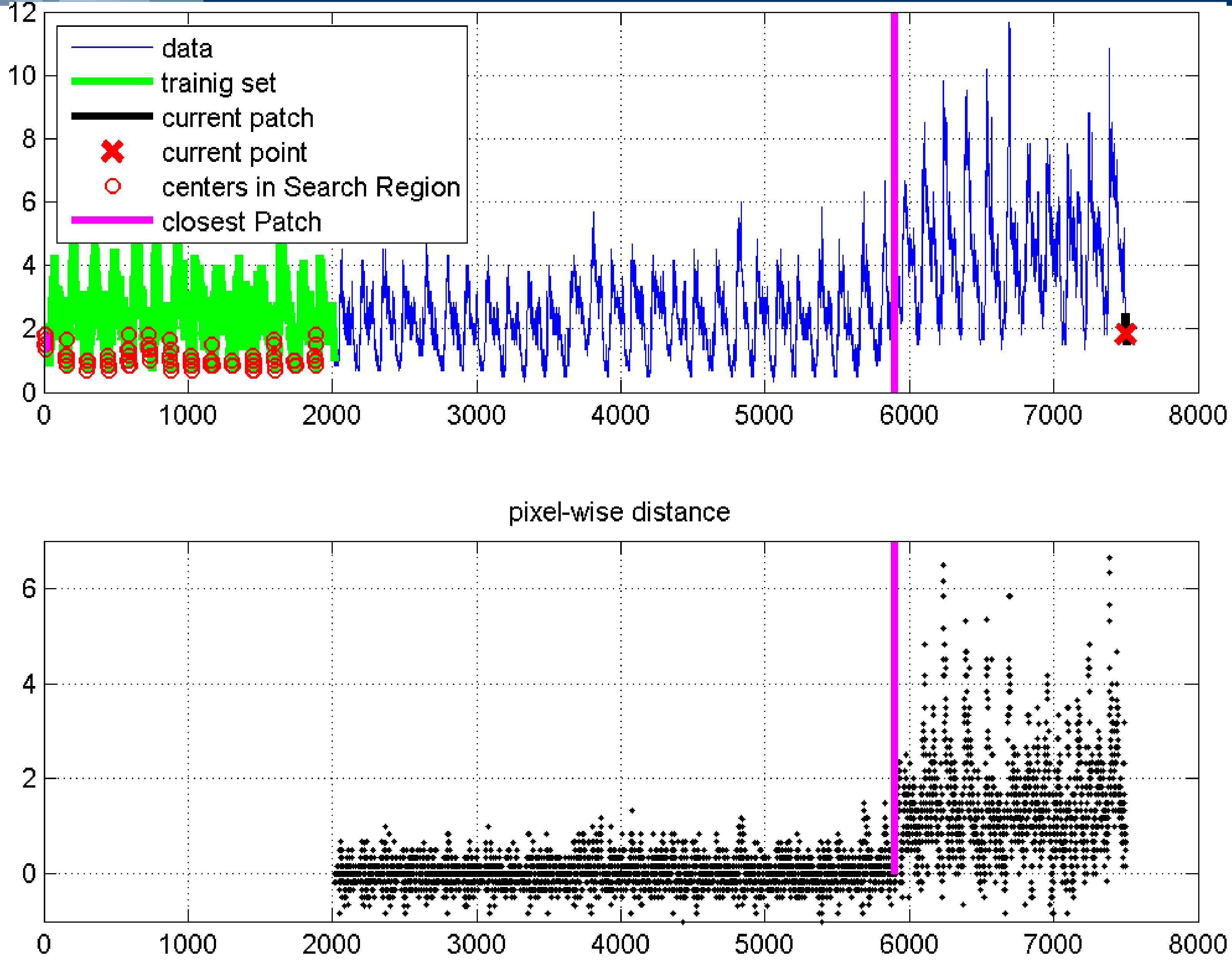
An example



data
trainig set
current patch
current point
centers in Search Region
closest Patch
12
10
8
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000

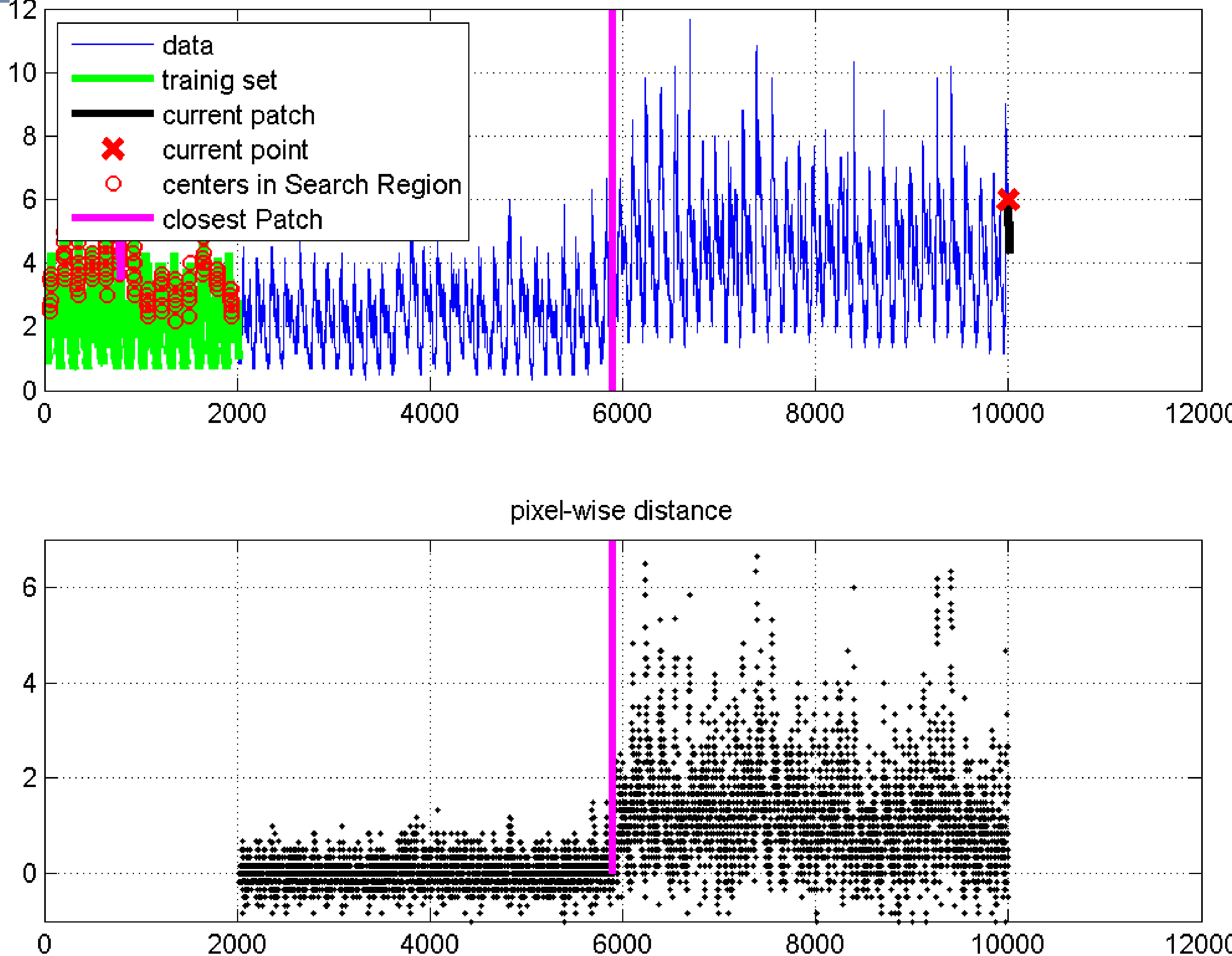
pixel-wise distance
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000

An example



data
trainig set
current patch
current point
centers in Search Region
closest Patch
12
10
8
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000
8000
pixel-wise distance
6
4
2
0
0
1000
2000
3000
4000
5000
6000
7000
8000

An example


data
trainig set
current patch
current point
centers in Search Region
closest Patch
0
2
4
6
8
10
12
0
2000
4000
6000
8000
10000
12000
pixel-wise distance
0
2
4
6
0
2000
4000
6000
8000
10000
12000

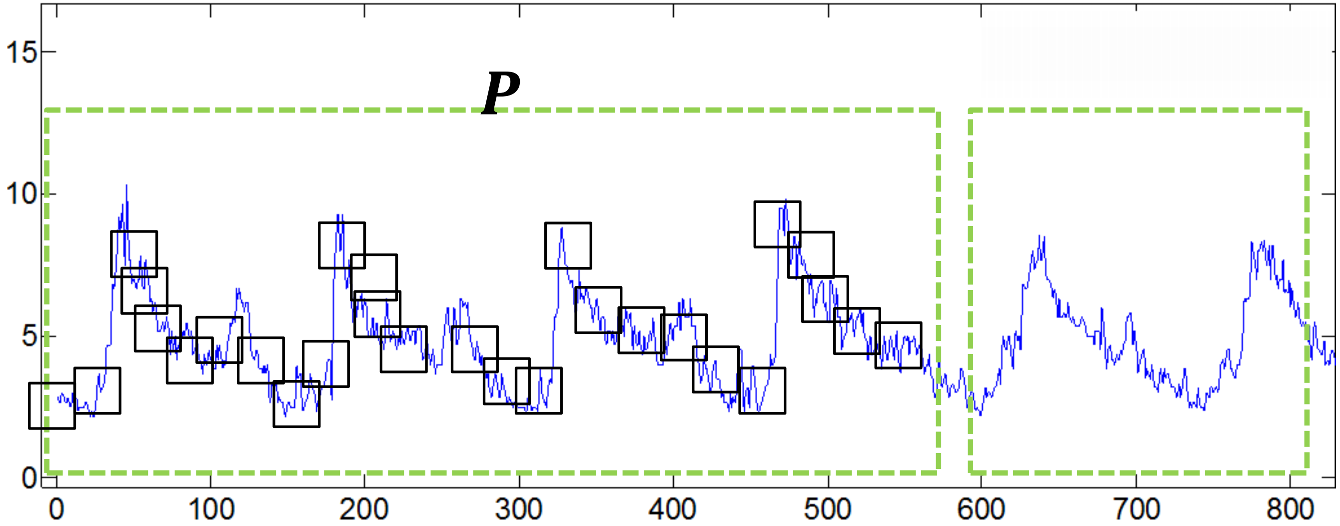
The CDT on the change indicators

- A CDT can be used to detect online and sequentially, changes in the distribution of x.
- CDTs often require a training sequence containing values of x that have been computed when S is in the normal state
- Change indicators are monitored by a CDT
  - We used the ICI-based CDT [Alippi et al 2010]

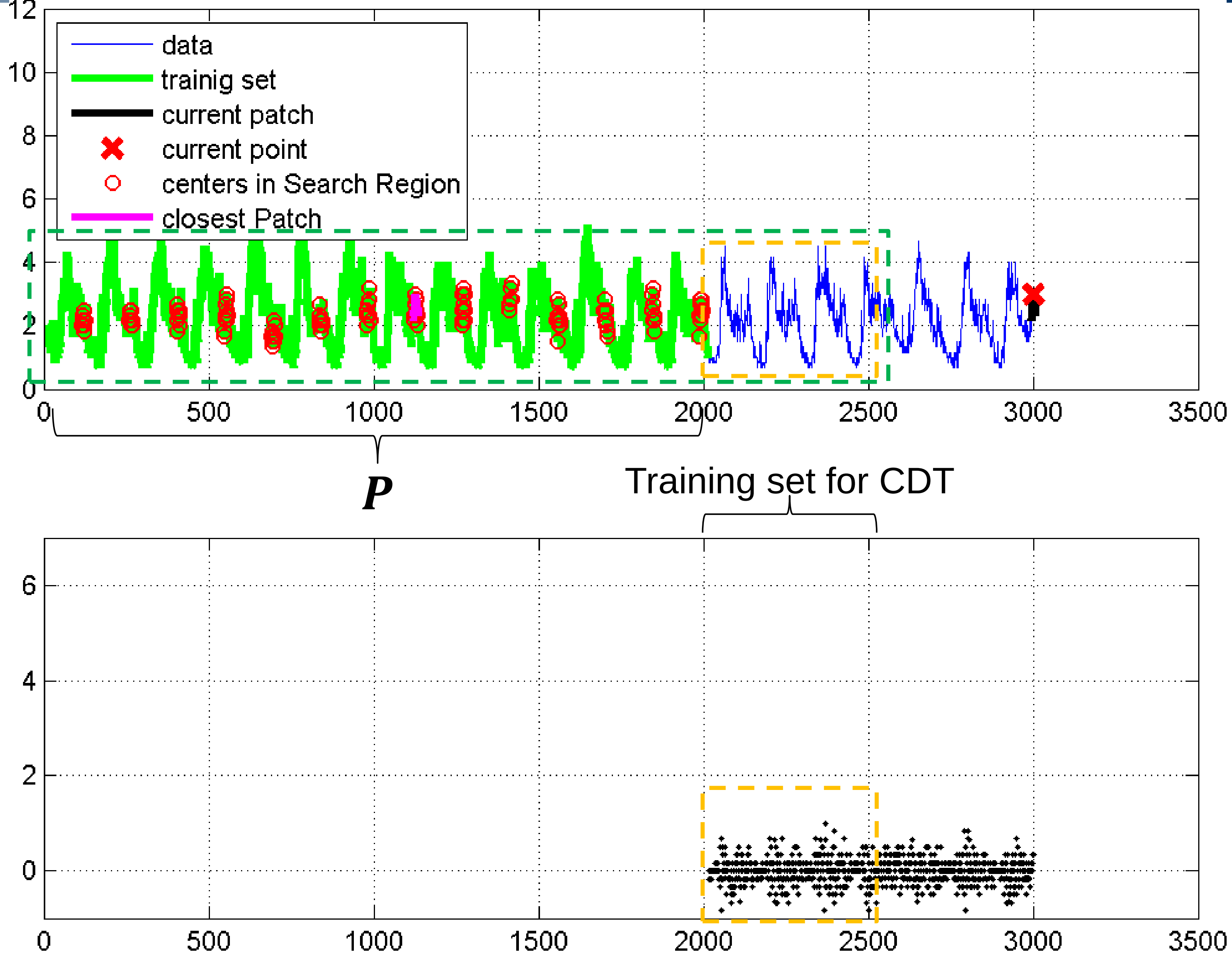
[Alippi et al 2010] C. Alippi, G. Boracchi, and M. Roveri, "A Just-In-Time adaptive classification system based on the Intersection of Confidence Intervals rule," Neural Networks, vol. 24, no. 8, pp. 791 – 800, 2011.

The Algorithm: The Training Phase

- CDTs often require a training sequence of values of x, computed when S is in the normal state
- Change indicators are monitored by a CDT
  - We used the ICI-based CDT [Alippi et al 2010]
- The initial training set TS is divided in two parts



The CDT Training Set



The Algorithm

Build the set of training patches ***P***

1- **input**: $\{s(\tau), \tau = 1, \ldots, L\}$, ν, δ, ϕ, M

2- define $\mathbf{P}$ from $TS = \{s(\tau), \tau = 1, \ldots, M\}$ as in (3),

3- **for** $(t = M + 1;\ t \leq L;\ t{+}{+})$ **do**

4- extract the patch $\mathbf{s}_t$ as in (1),

5- define the search region $R_{t,\phi,\delta}$ as in (8),

6- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7

7- compute the change indicator $x(t)$ as in (6),

end

8- configure the CDT on $\{x(t), t = M + 1, \ldots, L\}$,

9- wait for the next ν samples,

10- **while** $(s(t + \nu)$ *arrives*$)$ **do**

11- extract the patch $\mathbf{s}_t$ as in (1),

12- define the search region $R_{t,\phi,\delta}$ as in (8),

13- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7

14- compute the change indicator $x(t)$ as in (6),

15- **if** $(CDT\big(\{x(\tau), \tau = M, \ldots, t\}\big) == 1)$ **then**

16- detect a structural change in $\mathcal{S}$ at $\widehat{T} = t$.

17- **return**.

end

18- $t = t + 1$;

end

The Algorithm

Training Phase

Compute the change indicators over normal data

1- **input**: $\{s(\tau), \tau = 1, \dots, L\}$, ν, δ, ϕ, M
2- define $\mathbf{P}$ from $TS = \{s(\tau), \tau = 1, \dots, M\}$ as in (3),
3- **for** $(t = M + 1;\ t \leq L;\ t{+}{+})$ **do**
4- extract the patch $\mathbf{s}_t$ as in (1),
5- define the search region $R_{t,\phi,\delta}$ as in (8),
6- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7
7- compute the change indicator $x(t)$ as in (6),
end
8- configure the CDT on $\{x(t), t = M + 1, \dots, L\}$,
9- wait for the next ν samples,
10- **while** $(s(t + \nu)$ *arrives*$)$ **do**
11- extract the patch $\mathbf{s}_t$ as in (1),
12- define the search region $R_{t,\phi,\delta}$ as in (8),
13- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7
14- compute the change indicator $x(t)$ as in (6),
15- **if** $(CDT\big(\{x(\tau), \tau = M, \dots, t\}\big) == 1)$ **then**
16- detect a structural change in $\mathcal{S}$ at $\widehat{T} = t$.
17- **return**.
end
18- $t = t + 1$;
end

The Algorithm

Training Phase

Configure the ICI-based CDT on these change indicators

1- **input**: $\{s(\tau), \tau = 1, \dots, L\}$, ν, δ, ϕ, M
2- define $\mathbf{P}$ from $TS = \{s(\tau), \tau = 1, \dots, M\}$ as in (3),
3- **for** $(t = M+1;\ t \leq L;\ t{+}{+})$ **do**
4- extract the patch $\mathbf{s}_t$ as in (1),
5- define the search region $R_{t,\phi,\delta}$ as in (8),
6- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7)
7- compute the change indicator $x(t)$ as in (6),
end
8- configure the CDT on $\{x(t), t = M+1, \dots, L\}$,
9- wait for the next ν samples,
10- **while** $(s(t+\nu)$ *arrives*$)$ **do**
11- extract the patch $\mathbf{s}_t$ as in (1),
12- define the search region $R_{t,\phi,\delta}$ as in (8),
13- compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7)
14- compute the change indicator $x(t)$ as in (6),
15- **if** $(CDT(\{x(\tau), \tau = M, \dots, t\}) == 1)$ **then**
16- detect a structural change in $\mathcal{S}$ at $\widehat{T} = t$.
17- **return**.
end
18- $t = t + 1$;
end

Operational Life

Crop a patch around $s(t)$

```
input: {s(τ), τ = 1, ..., L}, ν, δ, φ, M
define P from TS = {s(τ), τ = 1, ..., M} as in (3),
for (t = M + 1; t ≤ L; t++) do
    extract the patch s_t as in (1),
    define the search region R_{t,φ,δ} as in (8),
    compute the patch most similar to s_t in R_{t,φ,δ}, (7
    compute the change indicator x(t) as in (6),
end
configure the CDT on {x(t), t = M + 1, ..., L},
wait for the next ν samples,
while (s(t + ν) arrives) do
    extract the patch s_t as in (1),
    define the search region R_{t,φ,δ} as in (8),
    compute the patch most similar to s_t in R_{t,φ,δ}, (7
    compute the change indicator x(t) as in (6),
    if (CDT({x(τ), τ = M, ..., t}) == 1) then
        detect a structural change in S at T̂ = t.
        return.
    end
    t = t + 1;
end
```

The Algorithm

Operational Life

Compute the change indicator $x(t)$

```
input: {s(τ), τ = 1, ..., L}, ν, δ, φ, M
define P from TS = {s(τ), τ = 1, ..., M} as in (3),
for (t = M + 1; t ≤ L; t++) do
    extract the patch s_t as in (1),
    define the search region R_{t,φ,δ} as in (8),
    compute the patch most similar to s_t in R_{t,φ,δ}, (7)
    compute the change indicator x(t) as in (6),
end
configure the CDT on {x(t), t = M + 1, ..., L},
wait for the next ν samples,
while (s(t + ν) arrives) do
    extract the patch s_t as in (1),
    define the search region R_{t,φ,δ} as in (8),
    compute the patch most similar to s_t in R_{t,φ,δ}, (7)
    compute the change indicator x(t) as in (6),
    if (CDT({x(τ), τ = M, ..., t}) == 1) then
        detect a structural change in S at T̂ = t.
        return.
    end
    t = t + 1;
end
```

The Algorithm

input: $\{s(\tau), \tau = 1, \dots, L\}$, ν, δ, ϕ, M

define $\mathbf{P}$ from $TS = \{s(\tau), \tau = 1, \dots, M\}$ as in (3),

for $(t = M + 1;\ t \leq L;\ t{+}{+})$ **do**

  extract the patch $\mathbf{s}_t$ as in (1),

  define the search region $R_{t,\phi,\delta}$ as in (8),

  compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7

  compute the change indicator $x(t)$ as in (6),

end

configure the CDT on $\{x(t), t = M + 1, \dots, L\}$,

wait for the next ν samples,

while $(s(t + \nu)$ *arrives*$)$ **do**

  extract the patch $\mathbf{s}_t$ as in (1),

  define the search region $R_{t,\phi,\delta}$ as in (8),

  compute the patch most similar to $\mathbf{s}_t$ in $R_{t,\phi,\delta}$, (7

  compute the change indicator $x(t)$ as in (6),

  if $(CDT\big(\{x(\tau), \tau = M, \dots, t\}\big) == 1)$ **then**

    detect a structural change in $\mathcal{S}$ at $\widehat{T} = t$.

    return.

  end

  $t = t + 1;$

end

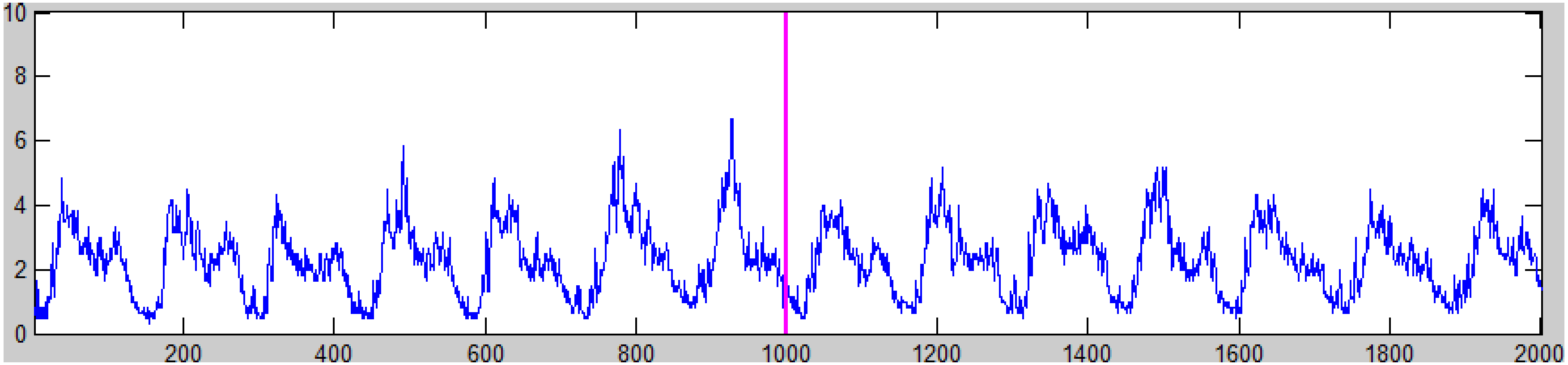
Operational Life

Run the CDT until a change is detected

EXPERIMENTS

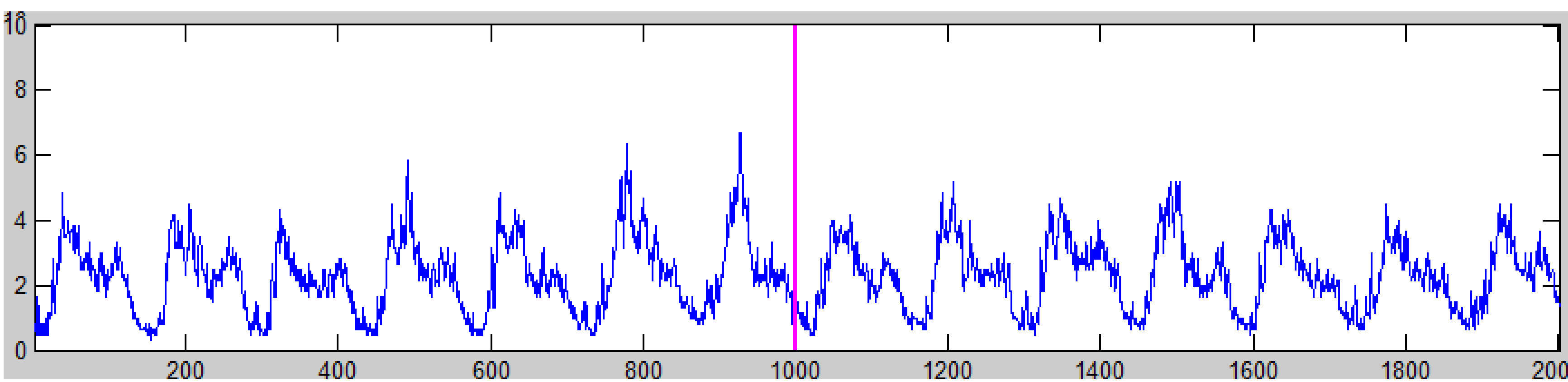
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period



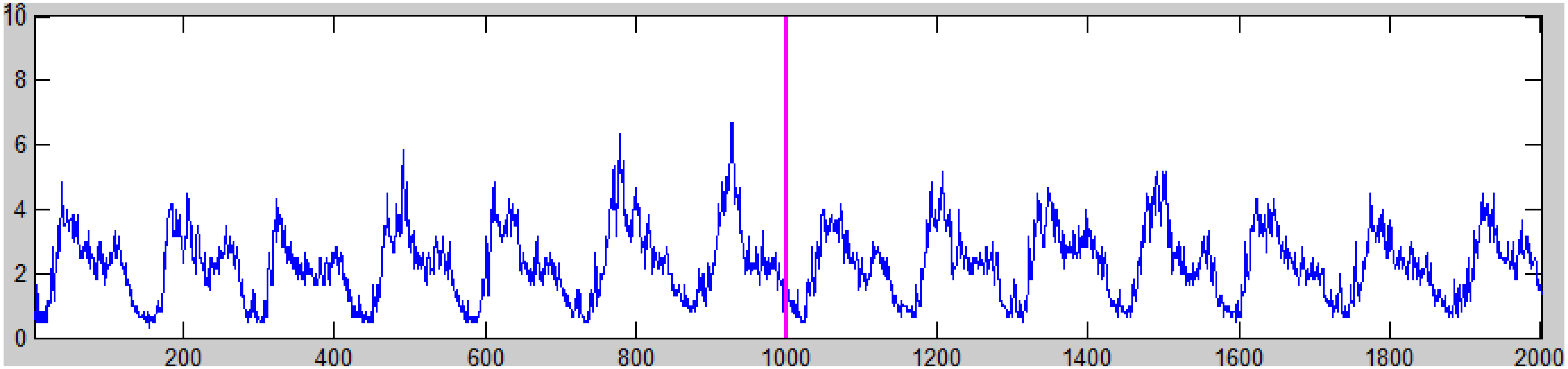
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** $s(t) = s(t) + o,\ t > T^*,\ o \in \{0.25a, 0.5a\}$



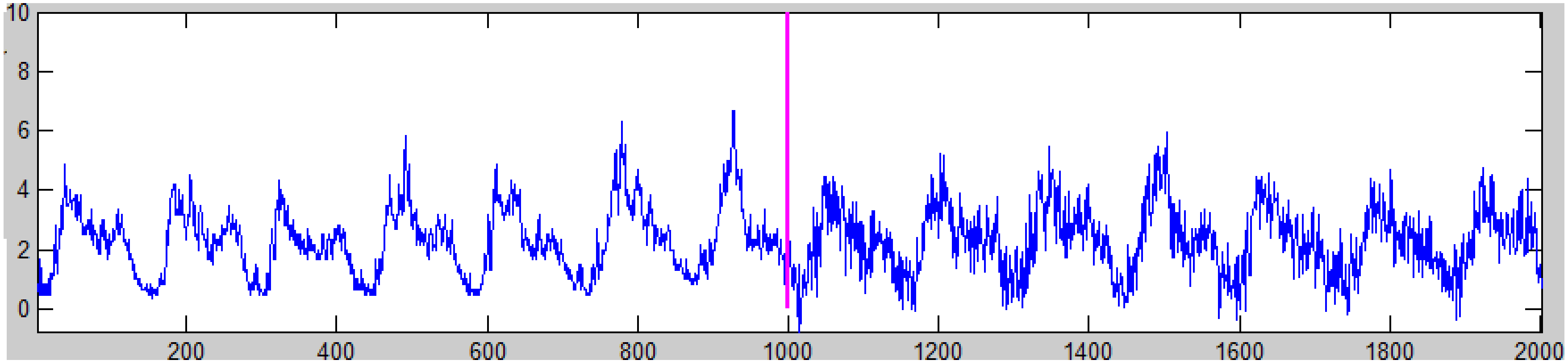
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** $s(t) = s(t) + o,\ t > T^*,\ o \in \{0.25a, 0.5a\}$



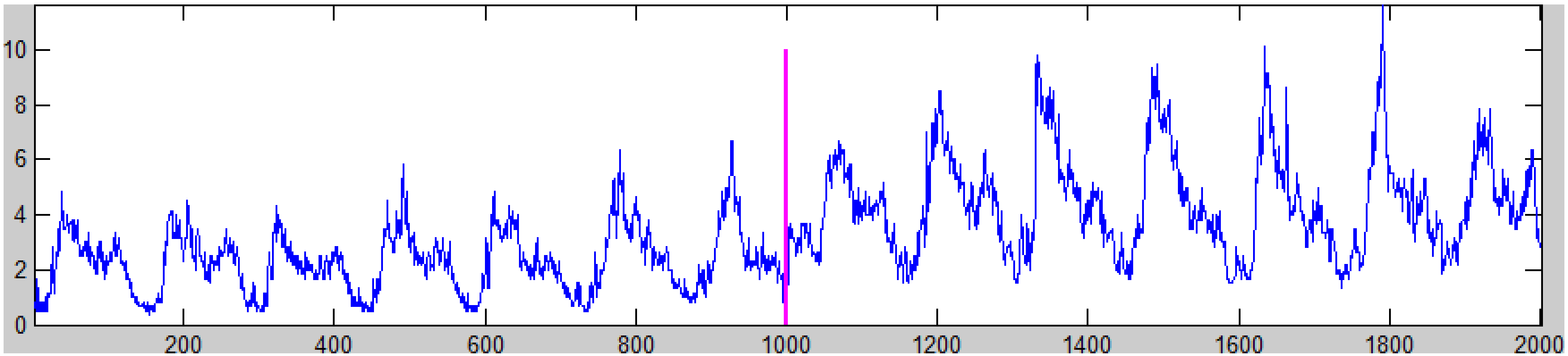
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** $s(t) = s(t) + o,\ t > T^*,\ o \in \{0.25a, 0.5a\}$
  - **Sensor Degradation:** $s(t) = s(t) + \eta(t),\ t > T^*, \eta \sim N(0, \sigma^2)$



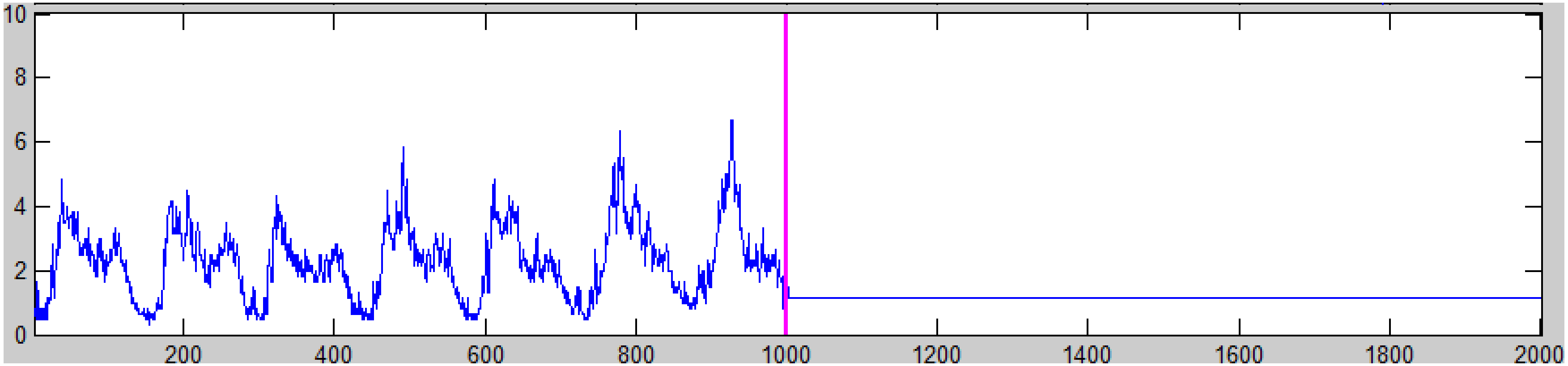
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** $s(t) = s(t) + o$, $t > T^*$, $o \in \{0.25a, 0.5a\}$
  - **Sensor Degradation:** $s(t) = s(t) + \eta(t)$, $t > T^*, \eta \sim N(0, \sigma^2)$
  - **Source Change** $s(t) = s_1(t), t > T^*$ from a different DMA



The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** $s(t) = s(t) + o,\ t > T^*,\ o \in \{0.25a, 0.5a\}$
  - **Sensor Degradation:** $s(t) = s(t) + \eta(t),\ t > T^*, \eta \sim N(0, \sigma^2)$
  - **Source Change** $s(t) = s_1(t), t > T^*$ from a different DMA
  - **Stack-at:** $s(t) = k, t > T^*$



The Considered CDTs

- **Residual-based:** a predictive model $f_{\hat{\theta}}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\hat{\theta}}(t)$$

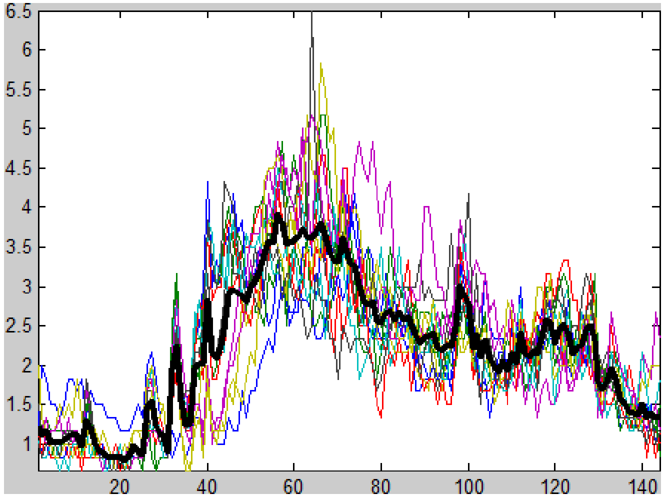
The Considered CDTs

- **Residual-based:** a predictive model $f_{\hat{\theta}}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\hat{\theta}}(t)$$

- **Template-based:** compute the difference w.t.r template, i.i.e the average flow profile in the n periods in the TS

$$p(t) = s(t) - \frac{1}{n}\sum_{i=1}^{n} s(t_0 + i\phi)$$



The Considered CDTs

- **Residual-based:** a predictive model $f_{\hat{\theta}}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\hat{\theta}}(t)$$

- **Template-based:** compute the difference w.t.r template, i.i.e the average flow profile in the n periods in the TS

$$p(t) = s(t) - \frac{1}{n}\sum_{i=1}^{n} s(t_0 + i\phi)$$

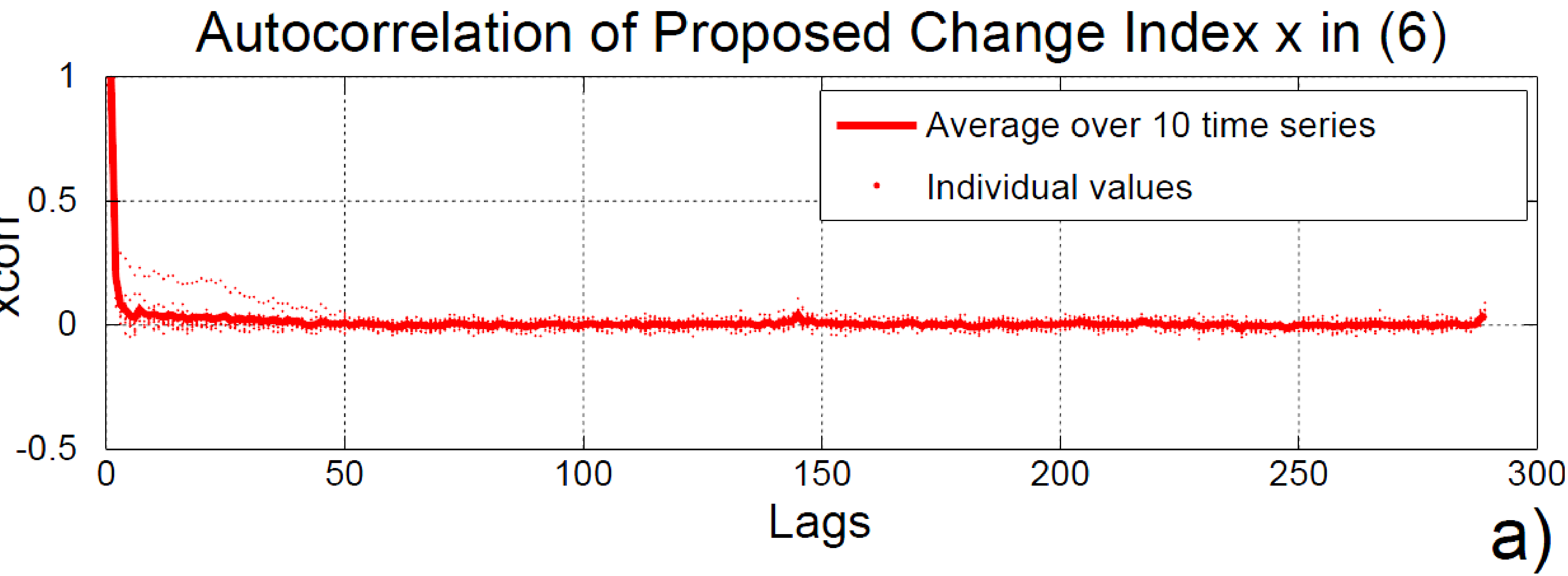
Self-similarity: the proposed solution, $\nu = 5,\ \delta = 5$

Details:

2 weeks of recordings used for bulding $\mathbf{P}$ / model fitting / template estimation, 400 samples for CDT configuration

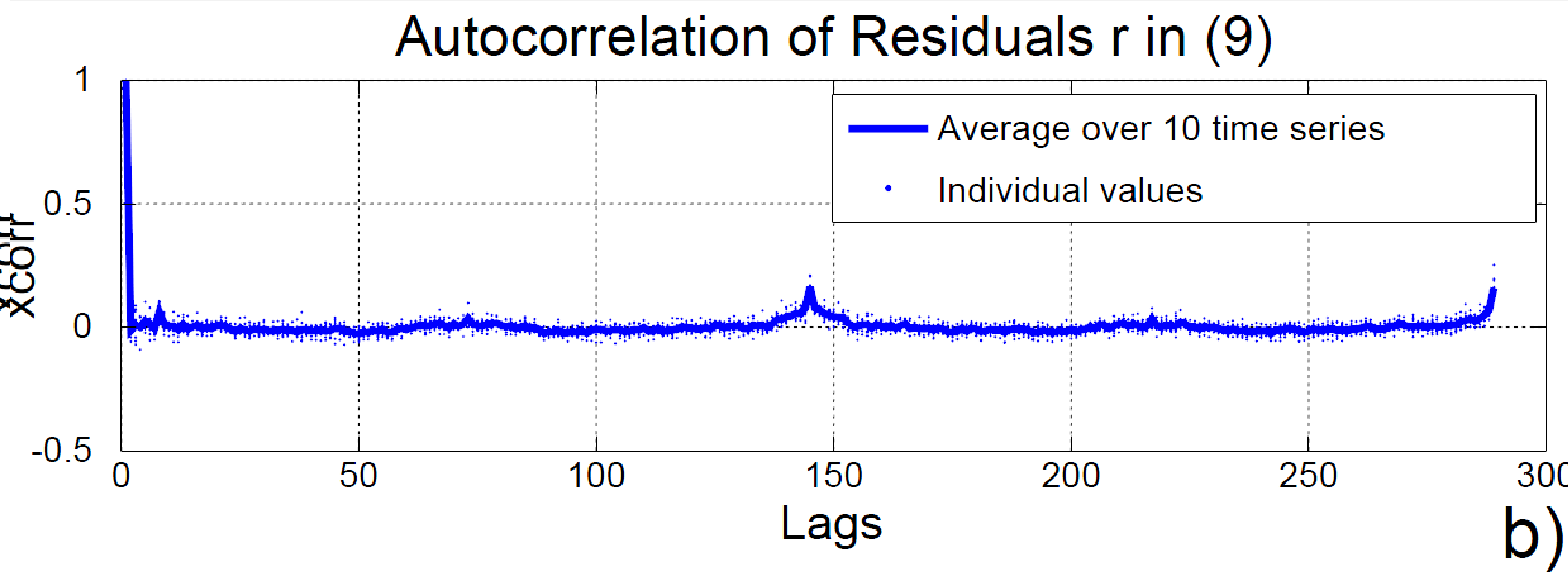
Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions



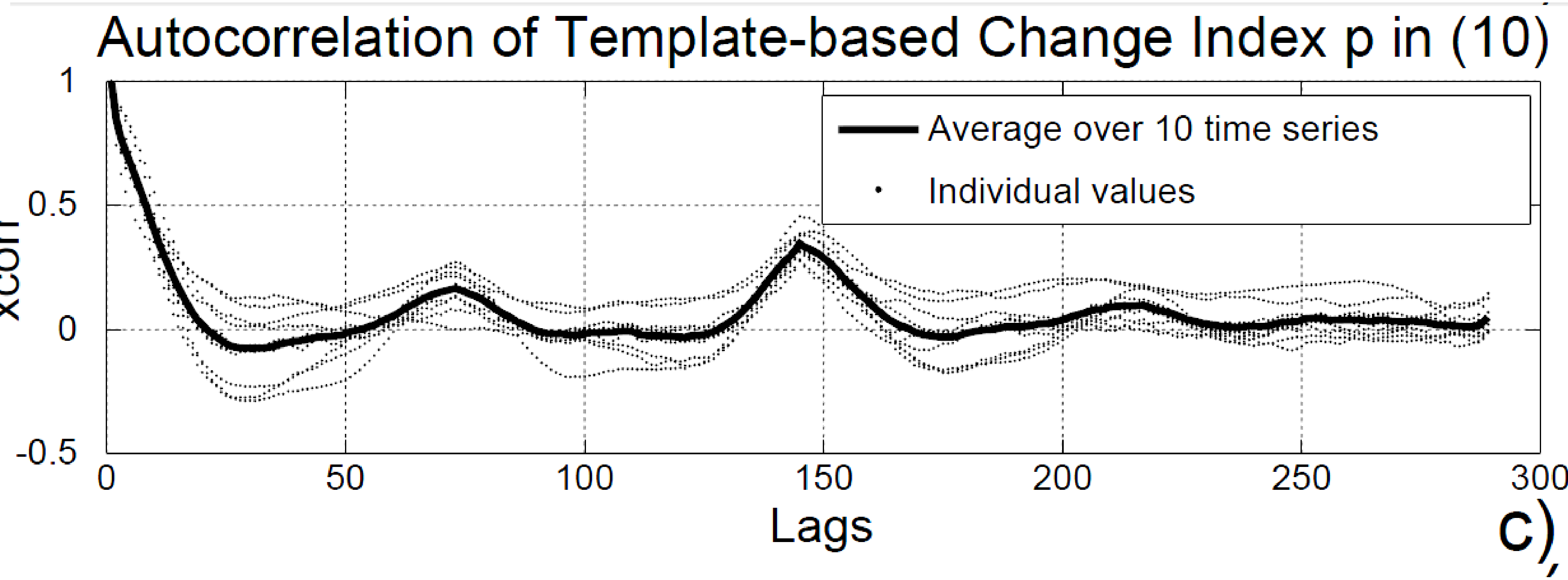
Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions



Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions

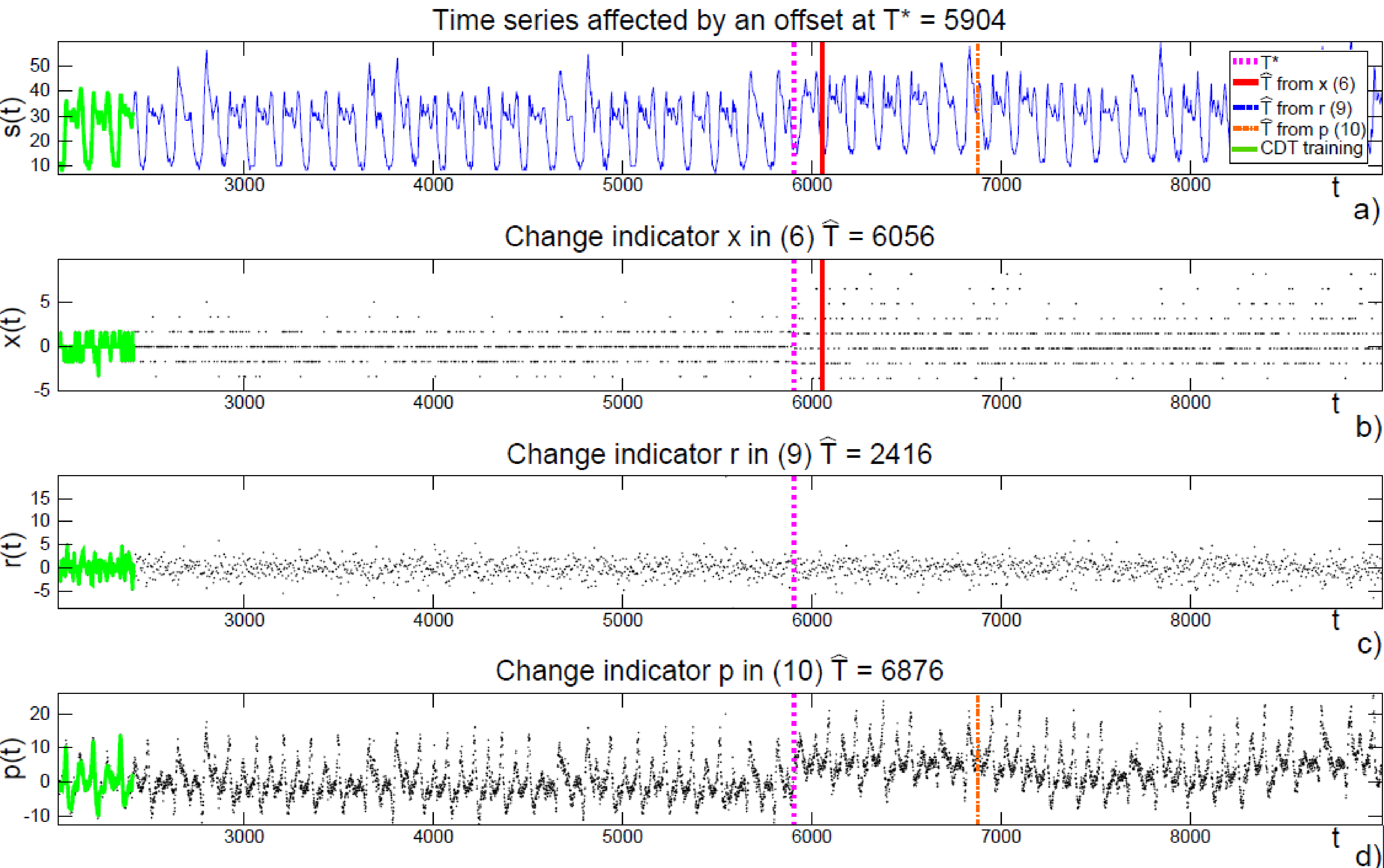


Change Detection Performance

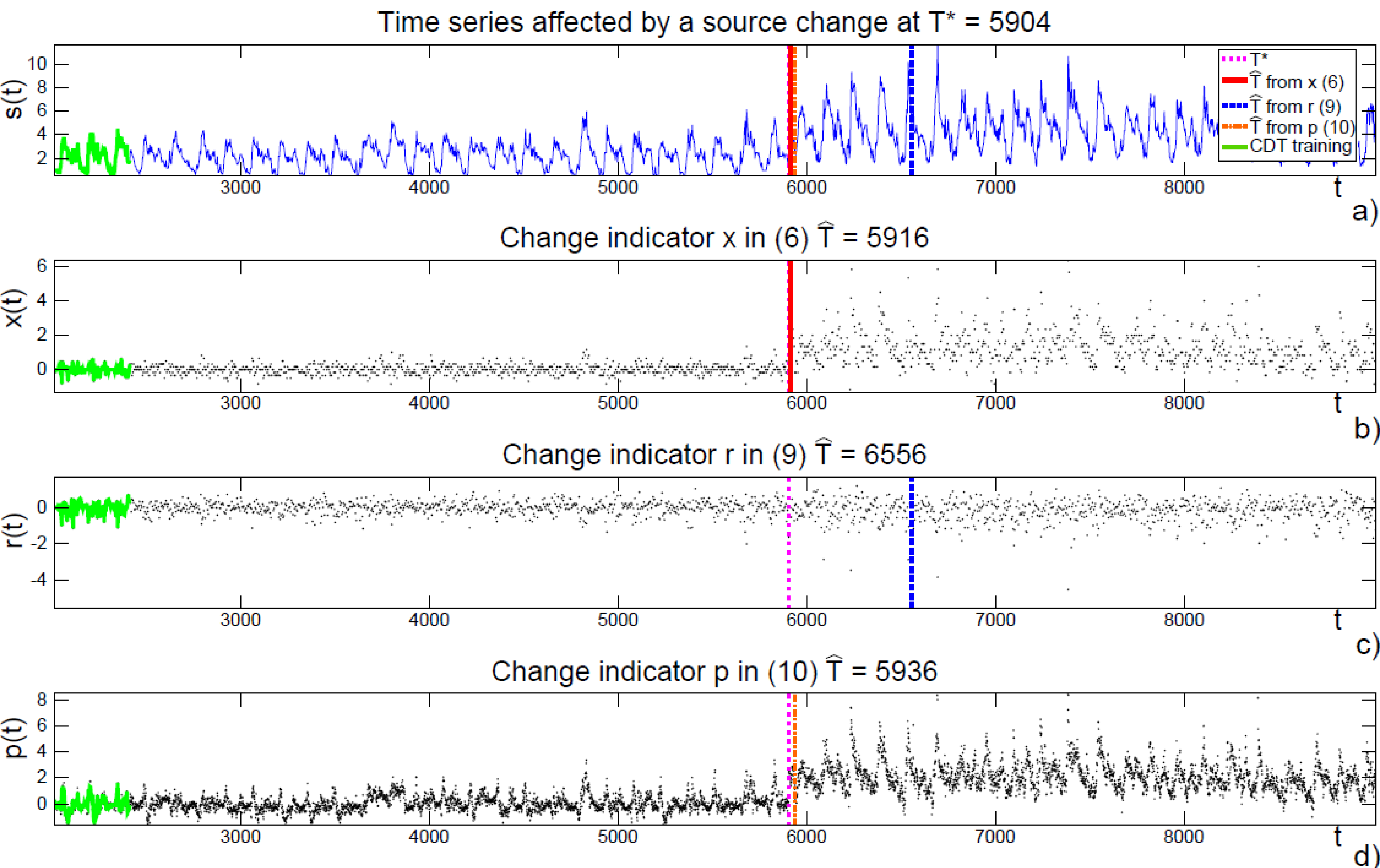
- FPR: False Positive Rate
- FNR: False Negative Rate
- DD: Expected Detection Delay

	Self-Similarity based			Residuals-based			Template-based		
	FPR	FNR	DD	FPR	FNR	DD	FPR	FNR	DD
offset 0.5	0.1	0.0	156.4	0.0	0.0	1554.0	0.2	0.0	332.0
offset 0.25	0.1	0.0	914.2	0.0	0.3	2803.4	0.2	0.0	747.0
sensor degradation 0.5	0.1	0.0	174.2	0.0	0.0	170.0	0.2	0.0	269.5
sensor degradation 0.25	0.1	0.0	336.4	0.0	0.0	288.0	0.2	0.0	652.0
source change	0.1	0.0	103.1	0.0	0.0	800.0	0.2	0.0	219.5
stack-at	0.1	0.0	169.8	0.0	0.0	160.0	0.2	0.0	534.5

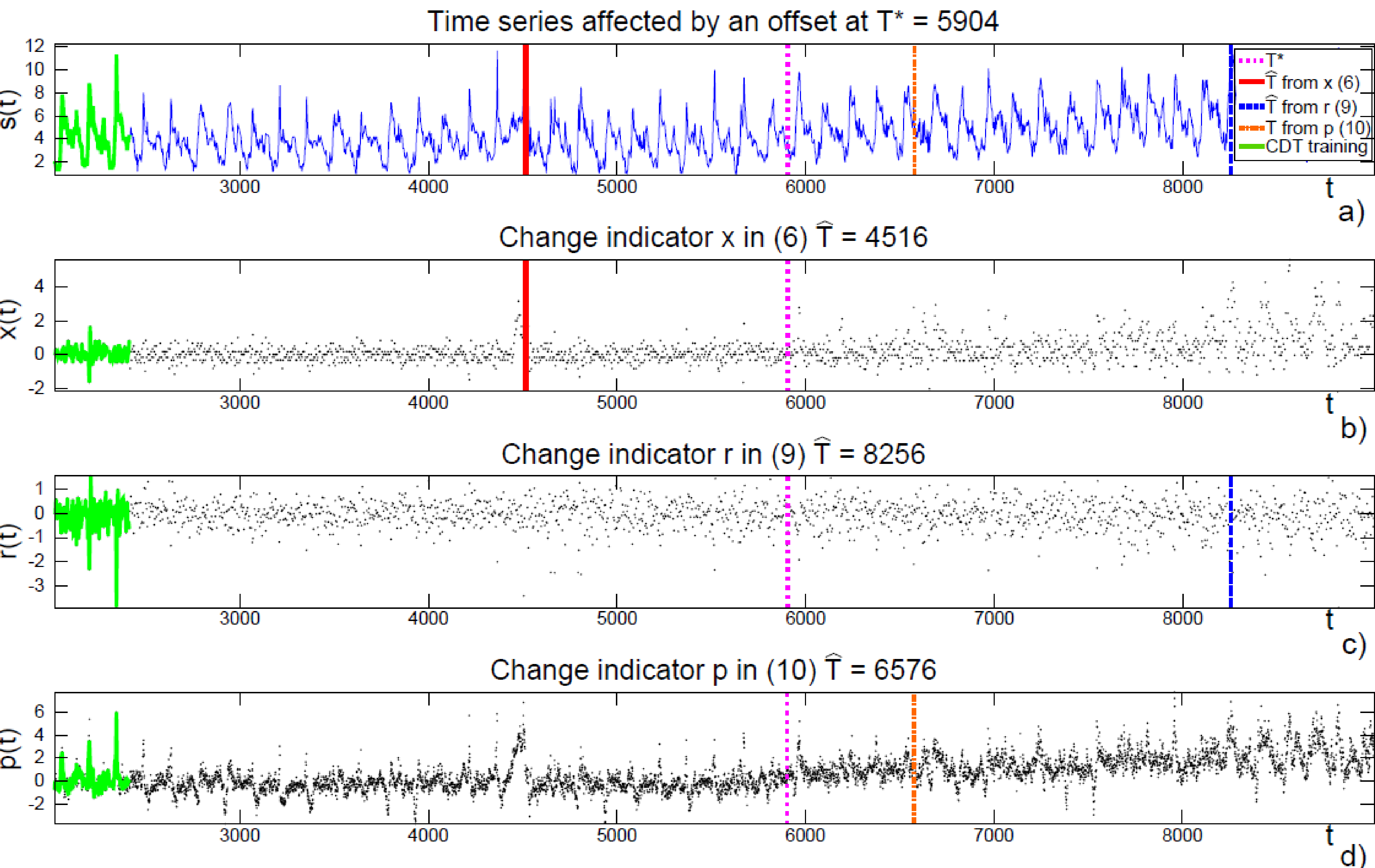
Offset of +50% the average flow value



Source Change



Offset of +25% the mean flow (False Positive)



CONCLUDING REMARKS

Concluding Remarks

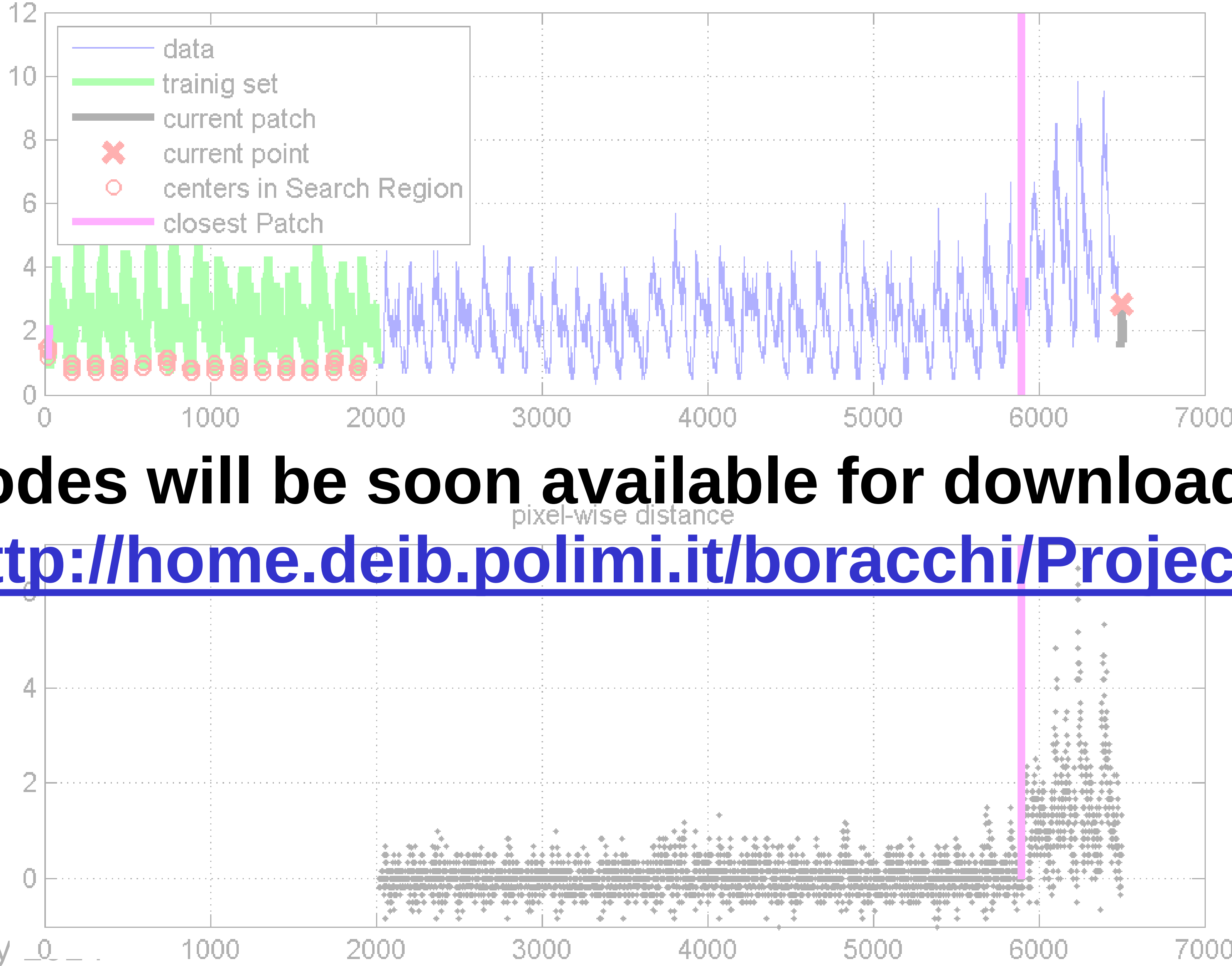
- Self similarity seems a promising approach for detecting changes in the structure of a self-similar datastream
  - Detection performance and autocorrelation show that x is very good at assessimg self similarity
  - Detection performance indicates that x reliably reacts to changes
- Ongoing Works
  - Investigating different change indicators for assessing self similarity.
  - Exolploiting self similarity in a collaborative manner (multichannel observations)
  - Self similarity when data are not periodic
  - Automatic criteria to identify the best patch size

Acknowldegments

Many thanks to Prof. Vicenç Puig from Universitat Politécnica de Catalunya, Barcelona, Spain, for providing us the datasets and for the meaningful discussions.

This work was supported by the FP7 EU project i-Sense, *Making Sense of Nonsense*, Contract No: INSFO-ICT-270428.

Questions?



Codes will be soon available for download at

http://home.deib.polimi.it/boracchi/Projects/