National and Kapodistrian
UNIVERSITY OF ATHENS

Context Compression: using Principal Component Analysis for Efficient Wireless Communications

Christos Anagnostopoulos

Pervasive Computing Research Group,
Department of Informatics & Telecommunications
National and Kapodistrian University of Athens, Greece

IEEE MDM 2011

Objective

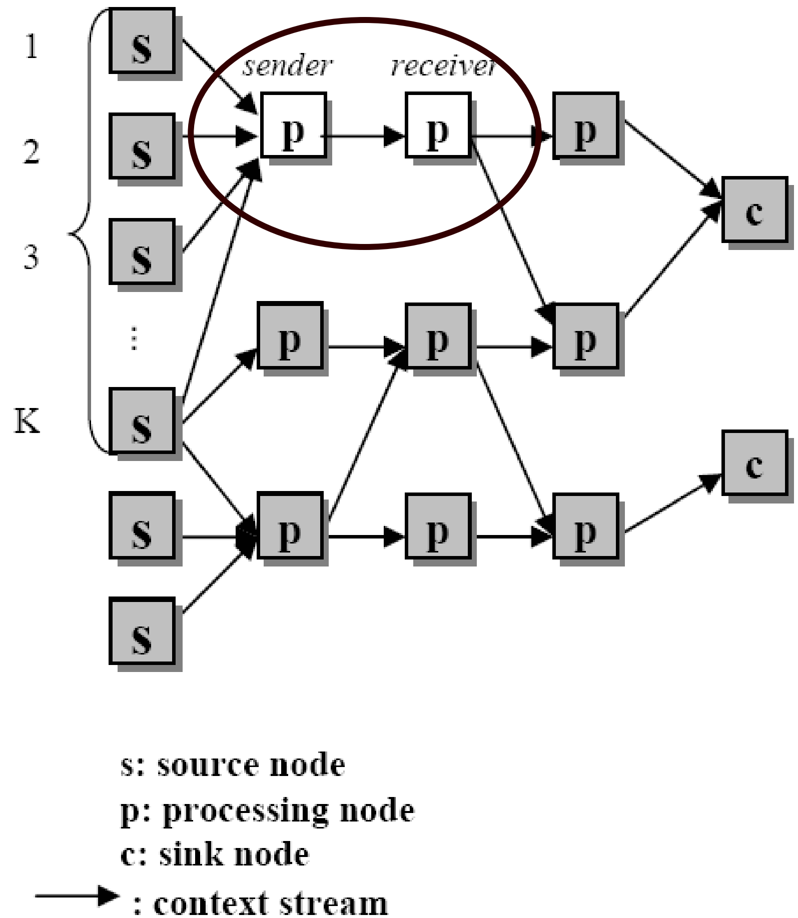
- **Improving** energy efficiency in Wireless Sensor Networks
- **Compressing** contextual information **prior** to transmission based on the *current* **Principal Components** of the sampled data.
- **Exploitation of the natural characteristics** of the pieces of context

Network model

- A set of **sensor** nodes (sources), **processing** nodes (relays), and **sink** nodes (consumers).

- Paths leading from the sources to sinks through processing nodes.

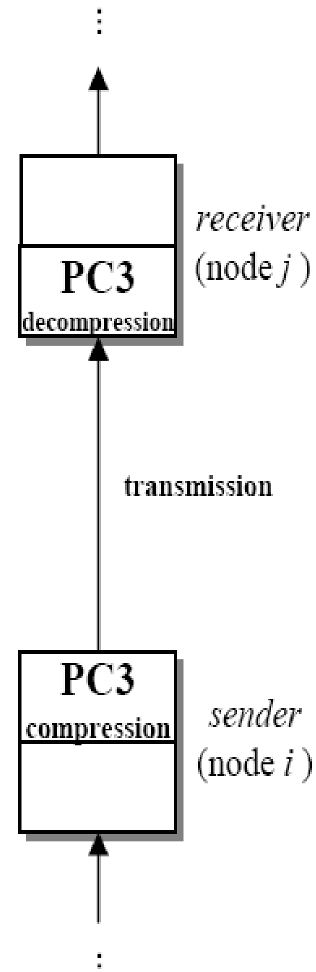
- A node is battery-powered (energy-constrained).



National and Kapodistrian
UNIVERSITY OF ATHENS

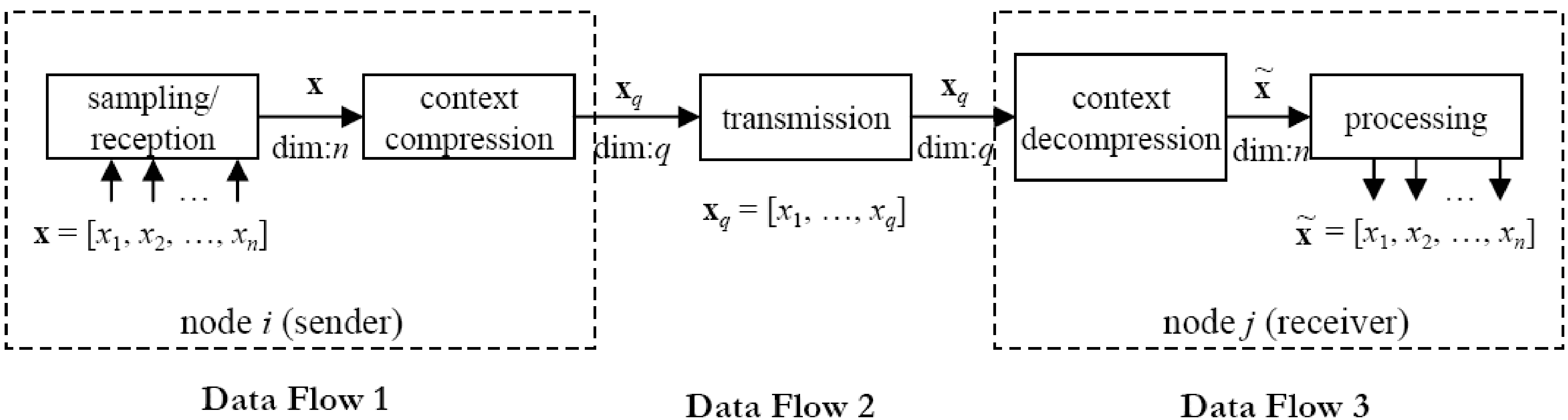
Sender and Receiver node

- Consider a **distributed compression algorithm** between sender *i* and receiver *j*:
  - Node *i* performs context compression and forwards the compressed contextual data (stream) to node *j*,
  - Node *j* performs context decompression in order to reproduce the original contextual data.
- Subsequently, node *j* can further act as a sender node for the upstream nodes and so on…



Sender and Receiver node (focus)

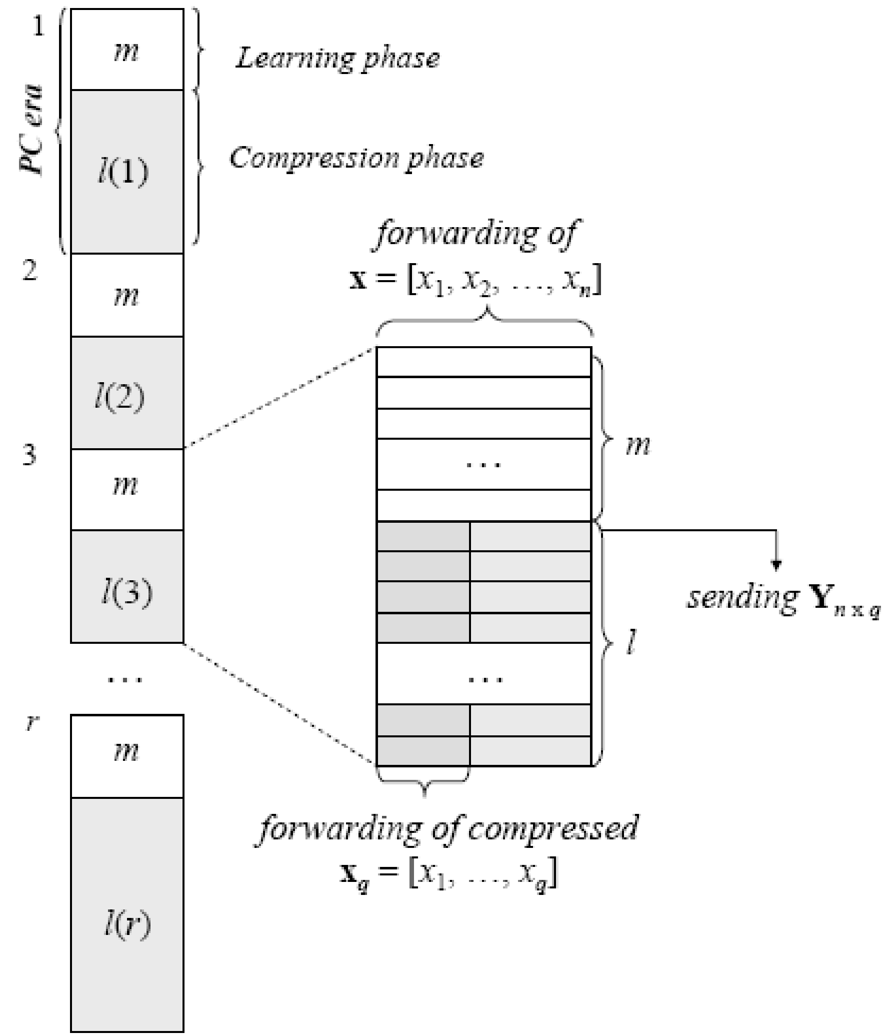
- Node *i* captures the *n*-dimensional context vector (CV) $\mathbf{x}$;
- Node *i* compresses $\mathbf{x}$ to a *q*-dimensional CV ($q < n$), $\mathbf{x}_q$ and forwards it to upstream node *j*.
- Node *j* reproduces the *n*-dimensional context vector, $\tilde{\mathbf{x}}(t)$ for further processing (or forwarding to upstream nodes).



National and Kapodistrian UNIVERSITY OF ATHENS

Sender node i (focus)

- t : discretized time domain
- $\mathbf{x}_i(t) = [x_{ik}(t)]$, $k = 1,\ldots, n$ be the CV of n measurements collected by node i at time t
- $x_{ik}(t)$: the kth contextual component (e.g., temperature, humidity, wind speed).
- Node i gathers the last $m > 0$ received CVs $\mathbf{x}(t - m)$, $\mathbf{x}(t - m+1)$, …, $\mathbf{x}(t)$, thus, forming a $m \times n$ matrix $\mathbf{X}$ consisting of m CVs.
- Based on $\mathbf{X}$, node i obtains the corresponding **principal components** (PCs) and, thus, reduces (compresses) a $\mathbf{x}(t)$ to $\mathbf{x}_q(t)$.



National and Kapodistrian UNIVERSITY OF ATHENS

Principal Component Analysis (PCA)

- A mechanism that performs *lossy* data compression
- PCA discovers a **linear** relationship among the contextual components x_k.
- PCA keeps the components that **better** describe the variance of the sample **X**.
  - The trade-off here is between compression (**count of principal contextual components retained**) and compression fidelity (**the variance preserved**).

National and Kapodistrian UNIVERSITY OF ATHENS

Principal Component Analysis (PCA)

- Given a set of *m* multivariate CVs of dimension *n*, $\mathbf{x}(t)$, $1 \leq t \leq m$, the PC basis is obtained by minimizing the optimization function:

$$J_q(\mathbf{x}(t), \mathbf{y_k}) = \frac{1}{m} \sum_{t=1}^{m} \| \mathbf{x}(t) - \sum_{k=1}^{q} \mathbf{y_k} \mathbf{y_k}^\top \mathbf{x}(t) \|^2$$

National and Kapodistrian UNIVERSITY OF ATHENS

Compression through PCA

- Dimensionality reduction from n to q with accuracy d%, i.e., the minimum number of PCs, q^*, that describe at least the d% of the variance of the projection of CVs on the PC basis expressed by the eigenvalues λ_k.
- Then:
  - we obtain a **compression** of $\mathbf{x}(t)$ of dimension $q < n$:

$$\mathbf{x}_q(t) = \mathbf{Y}^\top \mathbf{x}(t)$$

  - we obtain an **approximation** of $\mathbf{x}(t)$:

$$\tilde{\mathbf{x}}(t) = \mathbf{Y}\mathbf{x}_q^\top(t)$$

Context compression scheme

- **Learning phase (lasts for m):**
  - Node i learns the PCs of the last m measurements. Node i gathers the most recent m CVs. During this history window, i.e., for $1 \leq t \leq m$, node i forwards **each** received $\mathbf{x}(t)$ to the peer node j.
  - Once m CVs are received at node i then node i can determine the q principal components forming the $\mathbf{Y}$ matrix w.r.t. the recent m measurements and $a = 90\%$.

Context compression scheme

- **Compression phase (lasts for l):**
  - Node i forwards to node j the $\mathbf{Y}$ ($n \times q$) matrix only **once**.
  - The *trained* node i forwards to node j only the values of the q principal contextual parameters for a finite time period $l > 0$

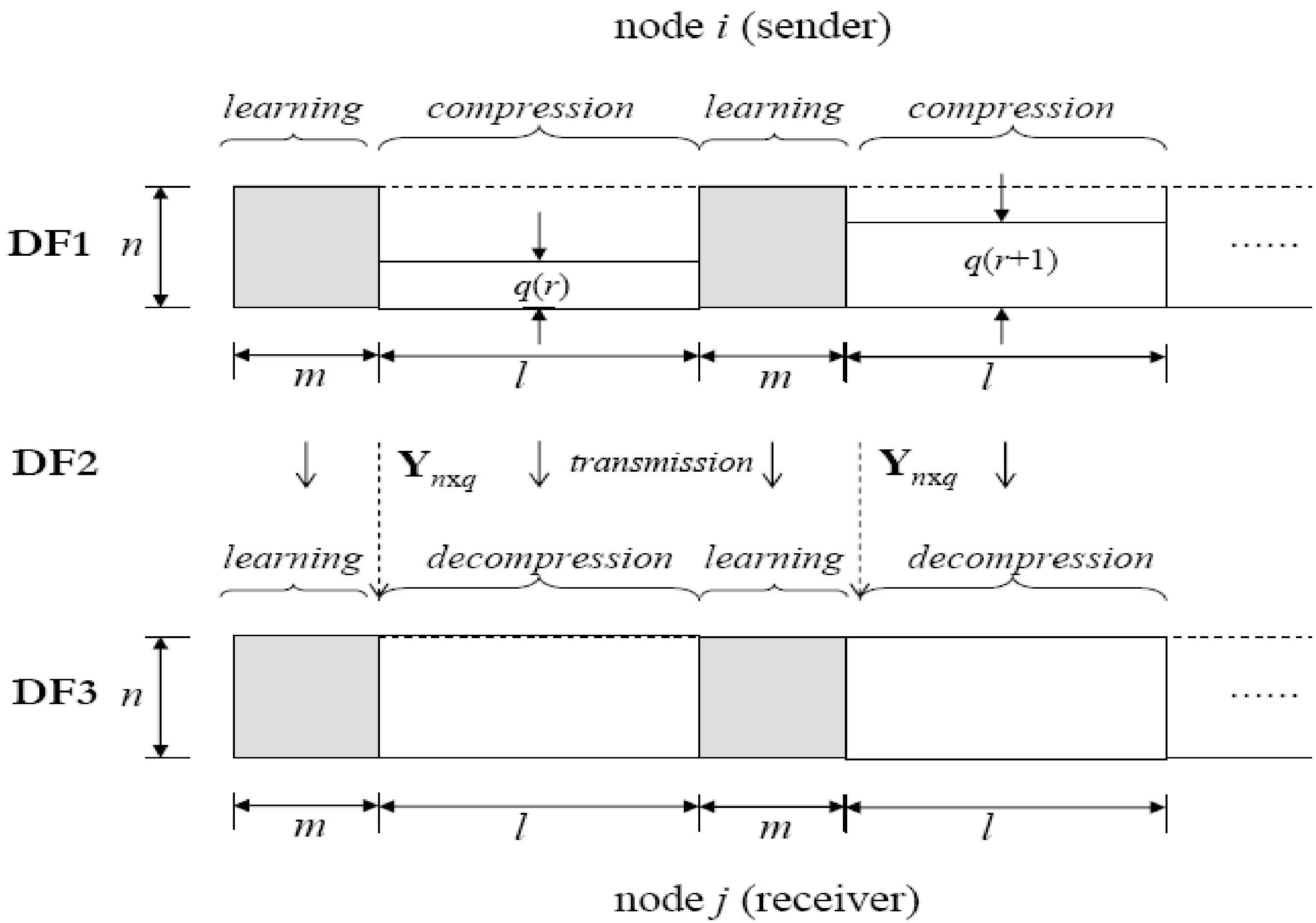
$$\mathbf{x}_q(t) = \mathbf{Y}^\top \mathbf{x}(t)$$

  - Node i forwards the compressed CV to node j.
  - On the other side, in the (de)-compression phase, node j has the required information (i.e., the $\mathbf{Y}$ ($n \times q$) matrix) to re-produce / approximate

$$\tilde{\mathbf{x}}(t) = \mathbf{Y}\mathbf{x}_q^\top(t)$$

National and Kapodistrian UNIVERSITY OF ATHENS

The Data Flow



National and Kapodistrian UNIVERSITY OF ATHENS

Error…

- For the finite period l we assume that
  - (i) the **number** of the PCs remain unchanged and
  - (ii) the **order** of the corresponding eigenvalues does not change.
- The node j re-produces the CVs during the l period inducing the re-production / reconstruction error:

$$e^R(t) = \| \tilde{\mathbf{x}}(t) - \mathbf{x}(t) \|$$

Adaptive mechanism

- The value of $l\,(r)$ for the r-th compression phase **is not a-priori known** and, additionally, there is no knowledge about the underlying data distribution w.r.t. the PCs.
- A controller $A(l\,)$ adjusts the period $l(r + 1)$ of the $(r + 1)$-th compression phase based on the error $e(l\,)$ and the $l(r)$ value of the r-th compression phase.
- Adaptation rule

$$l(r + 1) = l(r) + a(r) : a(r) \in \{-1, 0, 1\}$$

National and Kapodistrian UNIVERSITY OF ATHENS

Periodic error

- Periodic error $e(l)$: the average value of the relative error $e^*(t)$ within a period l

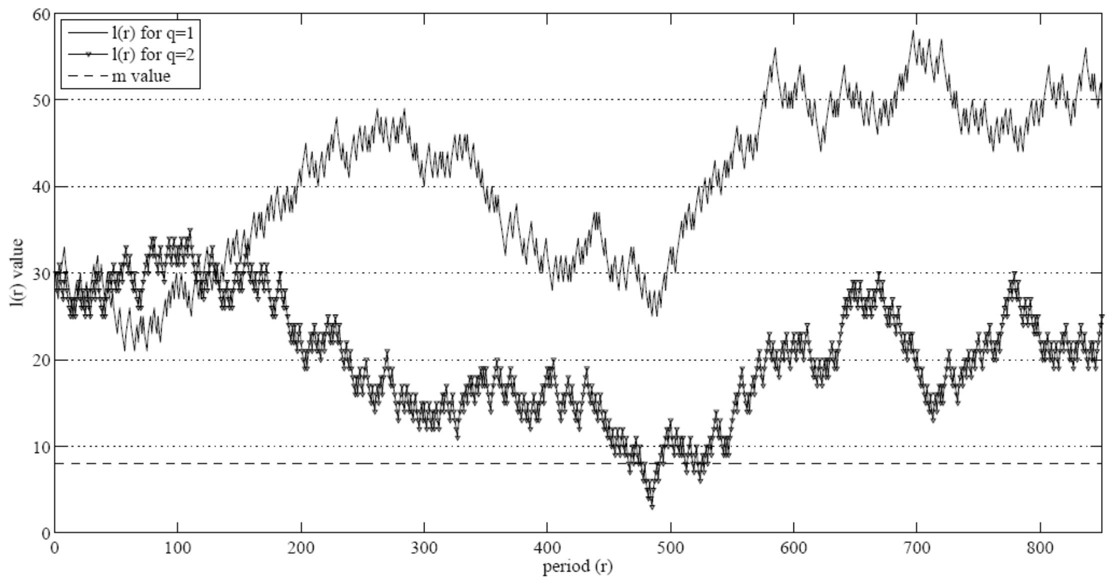
$$e(\ell) = \frac{1}{\ell} \sum_{t=m}^{m+\ell} e^*(t)$$

Adaptive mechanism

- The control parameters for the adaptation rule are:
  - $\Delta e(l)$: change in periodic error
  - Δl : change in compression phase length

THE ADAPTATION RULES FOR $a(r)$

$\Delta \ell$	$\Delta e(\ell)$ < 0	$\Delta e(\ell)$ 0	$\Delta e(\ell)$ > 0
$-\lvert a(r) \rvert$	-1	0	+1
0	0	+1	-1
$+\lvert a(r) \rvert$	+1	0	-1



National and Kapodistrian UNIVERSITY OF ATHENS

Performance assessment

- Real sensor readings of **temperature** (T), **humidity** (H), and **wind speed** (W).
- Information collected from experiments of the Sensor and Computing Infrastructure for Environmental Risks (SCIER) system, capable of delivering valuable real time information regarding a natural hazard (e.g., fire) and both monitoring and predicting its evolution.
- The corresponding contextual vectors are of $n = 7$ dimensions.
- Experiment with **32.25** hours of sensing.

National and Kapodistrian
UNIVERSITY OF ATHENS

Performance assessment

- Mica2 energy consumption model;
  - Mica2 operates with a pair of AA batteries that approximately supply 2200 mAh with effective average voltage 3V. It consumes 20mA if running a sensing application continuously which leads to a lifetime of 100 hours.
  - The packet header is 9 bytes (MAC header and CRC) and the maximum payload is 29 bytes. Therefore, the per-packet overhead equals to 23.7\% (lowest value).
  - For each contextual value the assumed payload is **4 bytes** (float).
  - A Mica2 message contains up to 7 contextual values (message payload: **28 bytes per message**).

Performance assessment

Table 1: Energy costs

Node operation mode	Energy cost
Instruction execution	4 nJ/instruction
Idle – Stand by	9.6 mJ/s - 0.33 mJ/s
Transmitting - Receiving	720 nJ/bit - 110 nJ/bit

National and Kapodistrian
UNIVERSITY OF ATHENS

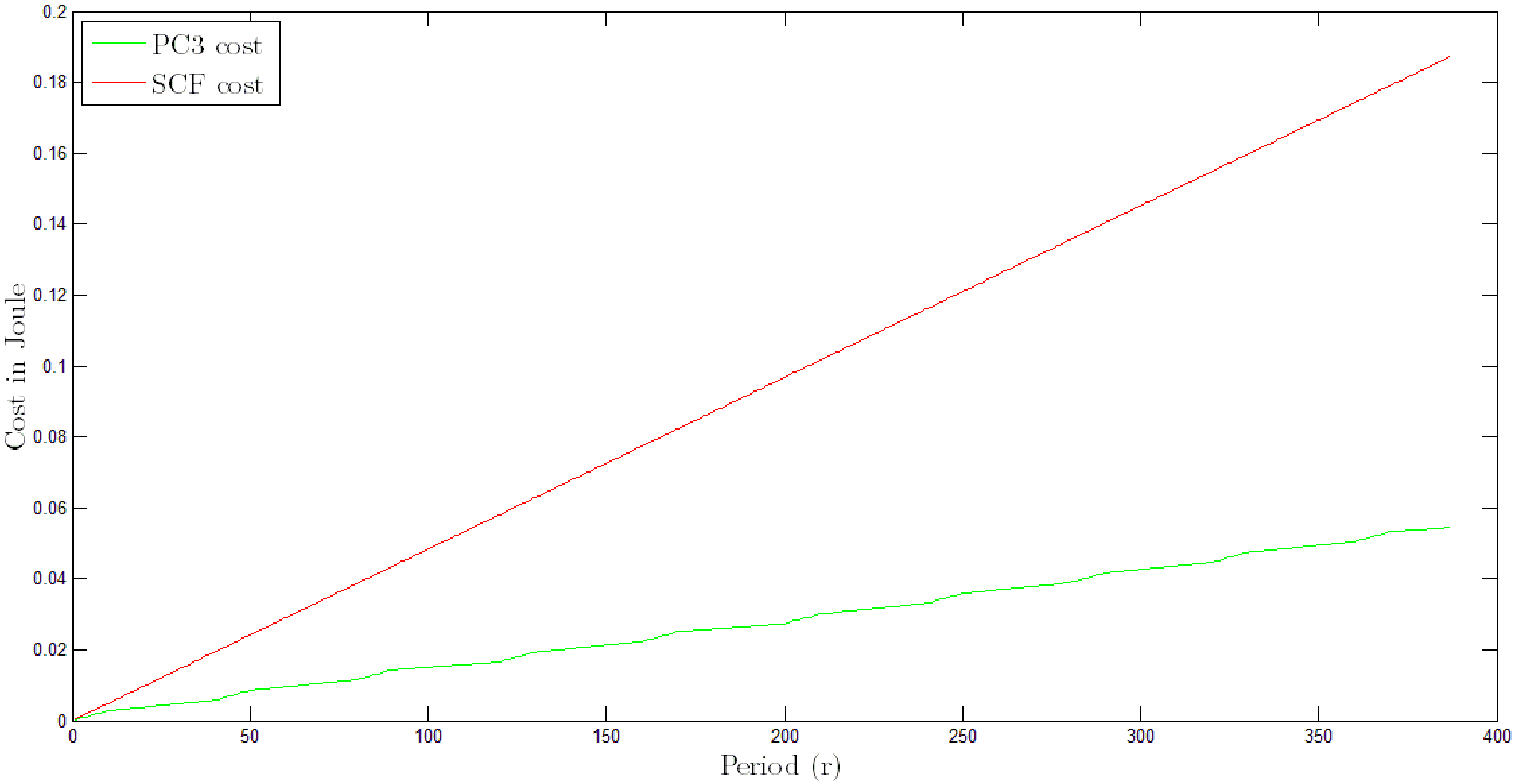
Total energy cost

$$c(t) = c(t-1) + c_R(t) + c_T(t) + c_I(t) + c_0(t)$$

- c_R, c_T are receive (rx) and transmit (tx) costs for CVs and the periodic transmission/reception of the **Y** matrix, respectively,
- c_I is the energy cost for the CPU instructions of the PCA (compression and decompression)
- c_0 is the state transition cost.

National and Kapodistrian UNIVERSITY OF ATHENS

Cost



National and Kapodistrian
UNIVERSITY OF ATHENS

Gain & Efficiency

➢**Gain:** The percentage cost gain $g(t) \in [0, 1]$ when applying PC3 w.r.t. SCF by using the energy costs $c_{PC3}(t)$ and $c_{SCF}(t)$

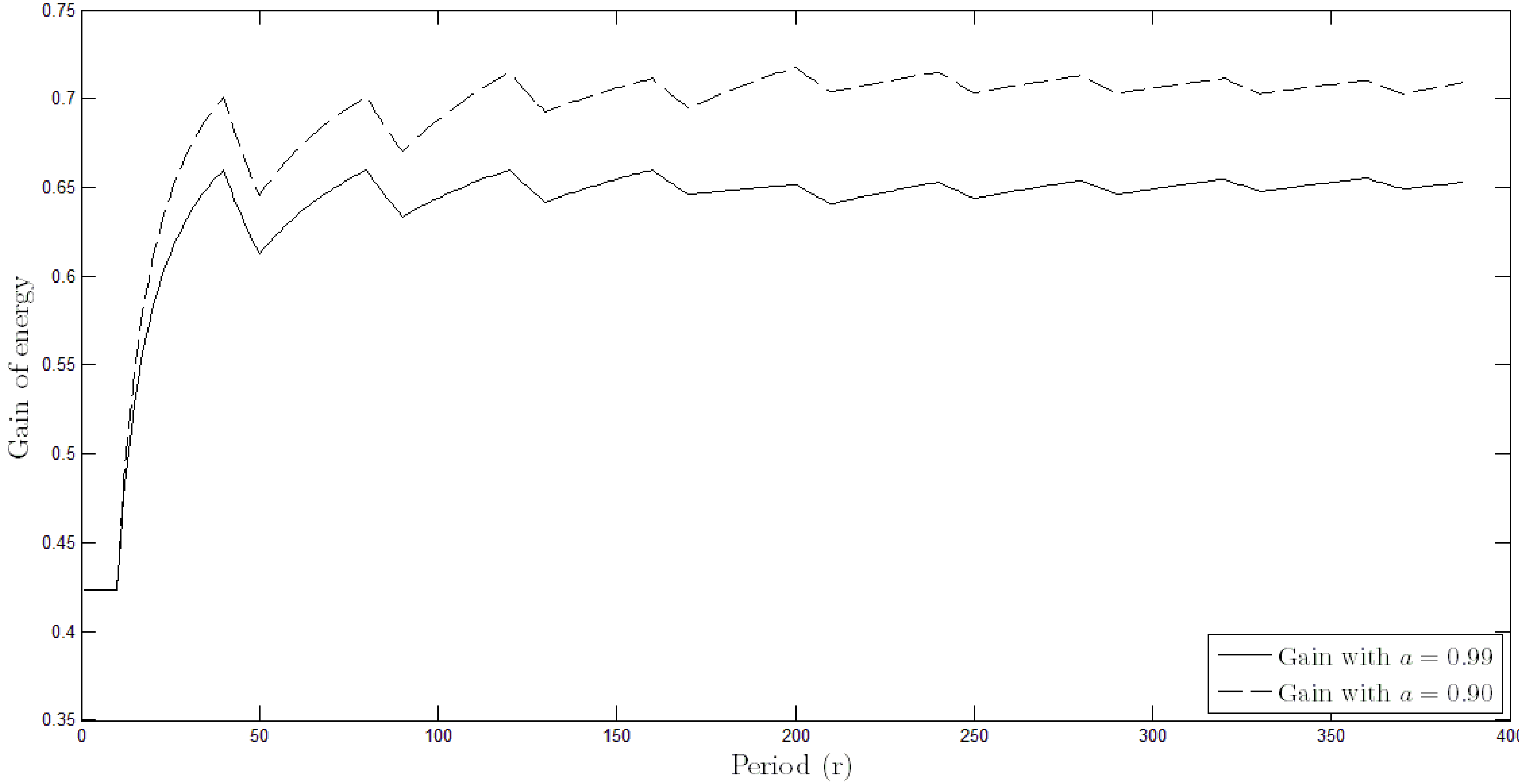
$$g(t) = \frac{c_{SCF}(t) - c_{PC3}(t)}{c_{SCF}(t)}$$

➢**Efficiency:** $w(t) \in [0, \infty)$ for a finite time horizon up to t is the portion of energy cost $c(t)$ out of the data accuracy 1-$e(t)$

$$w(t) = \frac{c(t)}{1 - e(t)}$$

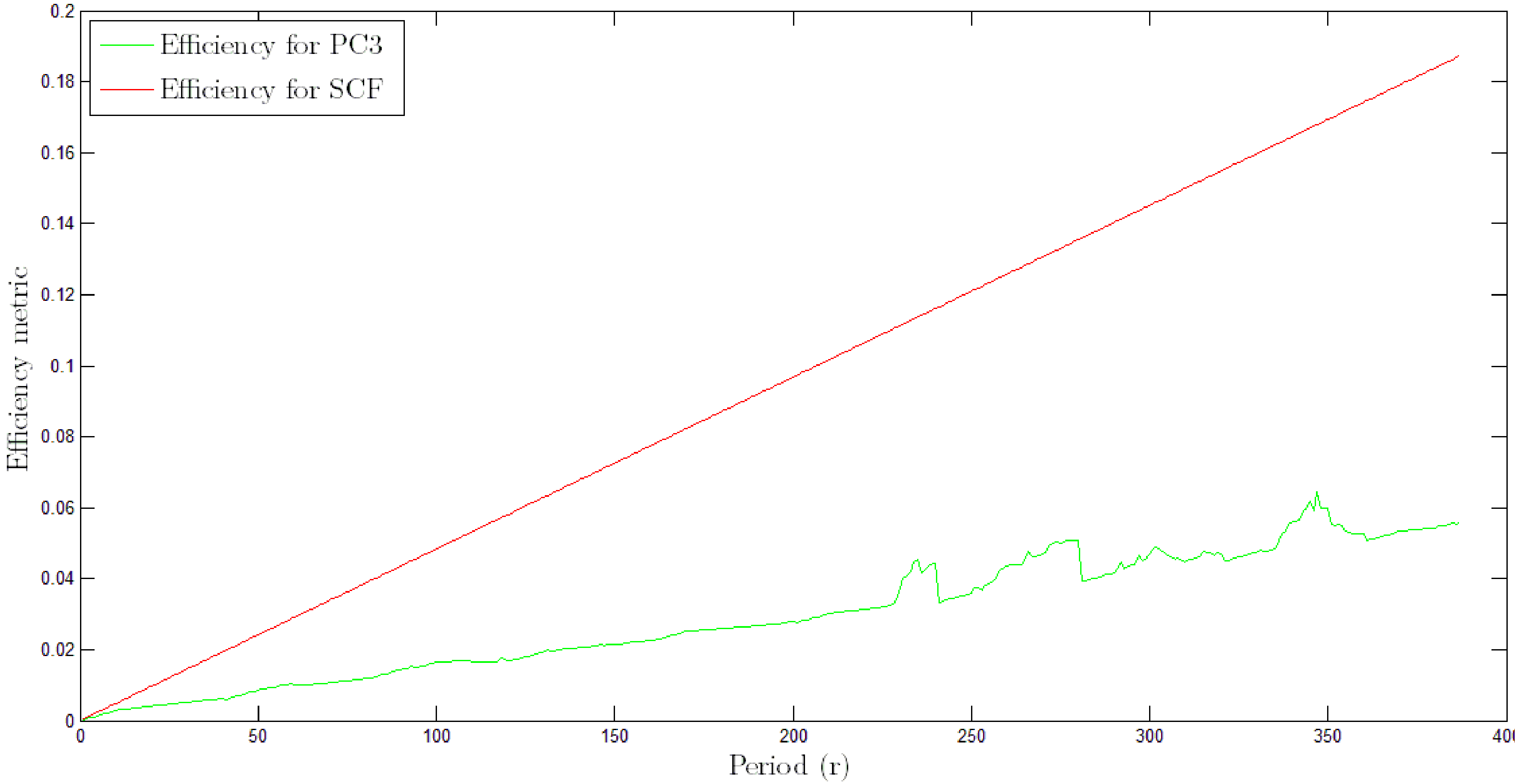
National and Kapodistrian UNIVERSITY OF ATHENS

Performance



National and Kapodistrian UNIVERSITY OF ATHENS

Performance



National and Kapodistrian UNIVERSITY OF ATHENS

Thank you!

National and Kapodistrian
UNIVERSITY OF ATHENS