Real-Time Scheduling with Renewable Energy

Ordonnancement mono-processeur
dans les systèmes temps réel récupérant l'énergie ambiante

UNIVERSITÉ DE NANTES

IRCCyN, Real-Time Systems Group

Maryline CHETTO

Outline

Introduction

First part: Classical Real-Time Systems

- Real-time systems
- Real-time scheduling algorithms

Second part: Energy Harvesting Technology

- Energy Harvesting Systems
- Power management: issues

Third part: Scheduling for Energy Harvesting Systems

- Slack Energy
- EDeg scheduler

Summary and future works

Central Issues

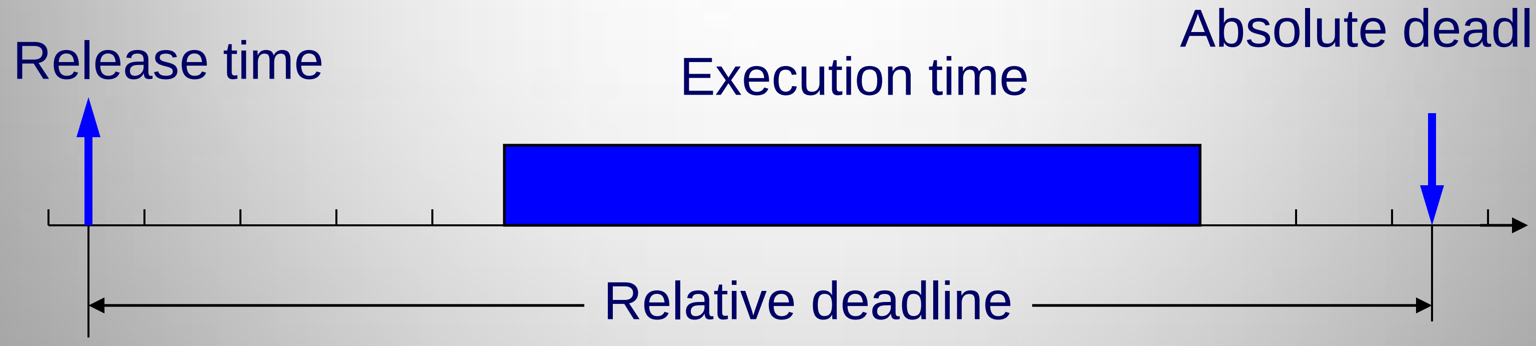
- Classical task scheduling, including EDF and RM only accounts for *timing* parameters of the tasks and consequently is not suitable when considering *energy* constraints.
- The problems we have to deal with are:
  - *How to extend classical schedulers so as to suitably exploit both the processor and the available ambient energy ?*
  - *What is the feasibility test ?*
  - *How to choose the size of the energy storage?*
- We will present a scheduling strategy for time critical tasks that run on a computing system that is powered through a renewable energy storage device.

Real-Time Systems

- Definition
  - Systems whose correctness depends on their temporal aspects as well as their functional aspects
- Performance measure
  - Timeliness on timing constraints (deadlines)
  - Speed/average case performance are less significant.
- Key property
  - Predictability on timing constraints

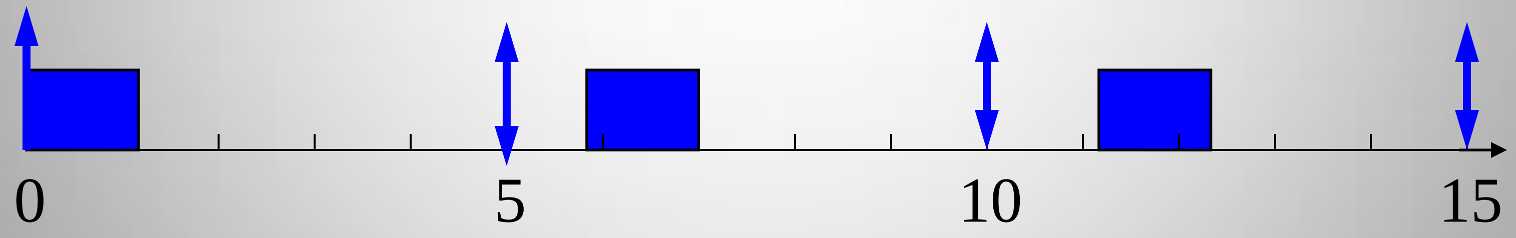
Real-Time Task

- **Task**
  - computation, file read, sensing, message transmission, etc
- **Characteristics**
  - Resources required to run
  - Timing parameters



Real-Time periodic Task

- Periodic task (*C*, *T*)
  - It repeats at regular intervals
  - Period *T* = interval between two successive release time
  - Execution time *C* = WCET($0 < C < T$)
  - Utilization U = C/T

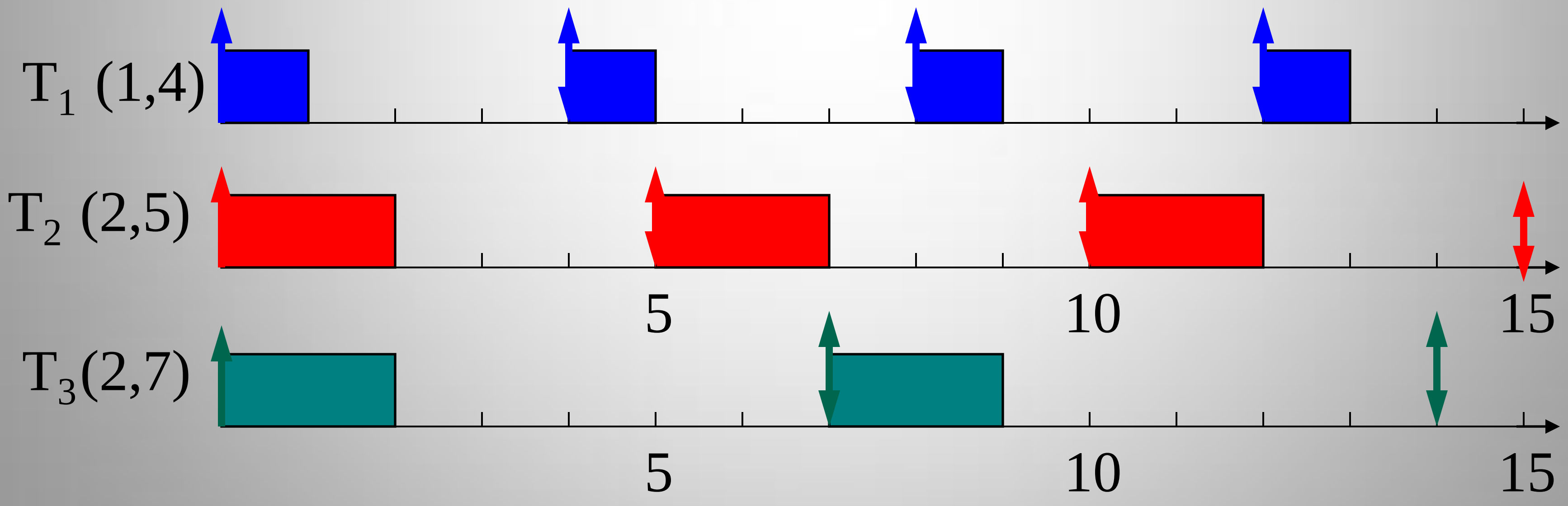


Deadlines: Hard vs. Firm

- Hard deadline
  - Disastrous or very serious consequences may occur if the deadline is missed
  - Validation is essential : can all the deadlines be met, even under worst-case scenario?
- Firm deadline (also said weakly hard)
  - Ideally, the deadline should be met for maximum performance. The performance degrades in case of deadline misses.
  - Best effort approaches / statistical guarantees
  - Optimization of a Quality of Service parameter

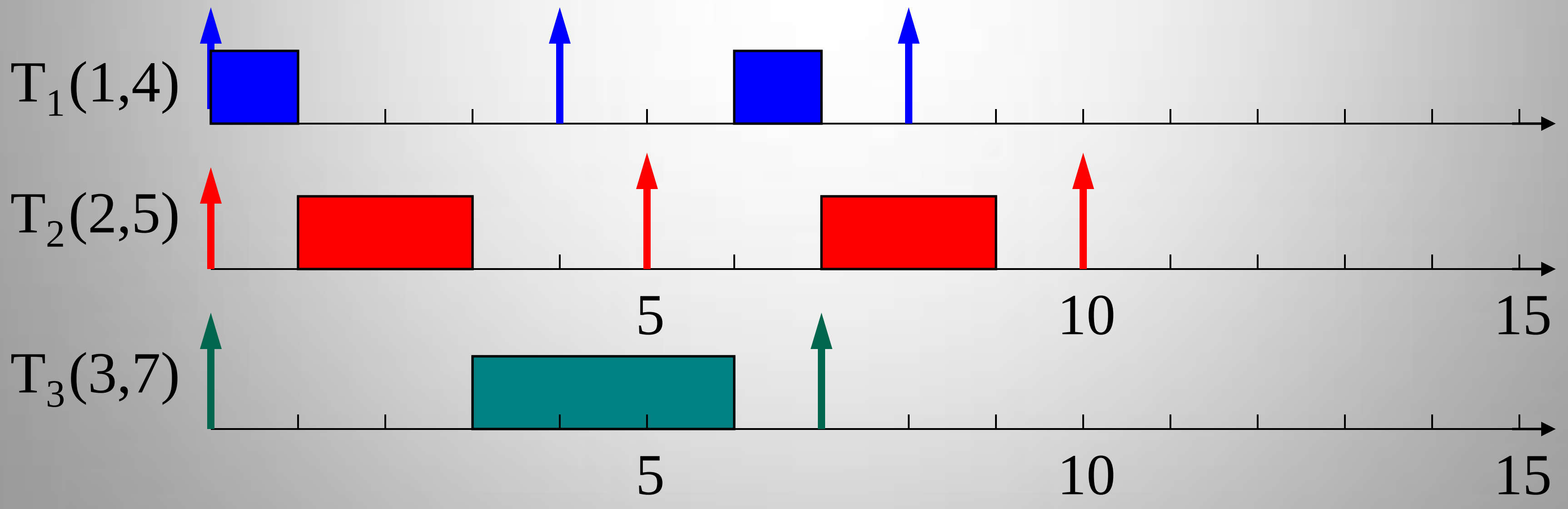
Real-Time Scheduling

- Determines the order of real-time task executions
- Static-priority scheduling (Rate Monotonic)
- Dynamic-priority scheduling (Earliest Deadline First)



EDF (Earliest Deadline First)

- Executes a job with the earliest deadline
  - Optimal and Full processor utilization
  - Non-clairvoyant and non-idling algorithm
  - BUT misbehavior during overload conditions



Design objectives

To execute, each task requires only processor time under the previous assumptions.

Our goal was **to meet deadlines requirements** taking into account **demands of processing time** of all the tasks

- **Given the limited availability of the processor** (limited speed and limited number)
- **Energy is assumed to be non limited**

Our goal will **to meet deadlines requirements** taking into account **demands of processing time AND demands of energy**

the energy availability will be limited

Second part:
Energy Harvesting Technology

Energy harvesting

- Energy harvesting is the capture of ambient energy to provide electricity for small and or mobile equipment

- Energy harvesting = energy scavenging

- can derive energy from a variety of sources including solar, vibration, and temperature variation

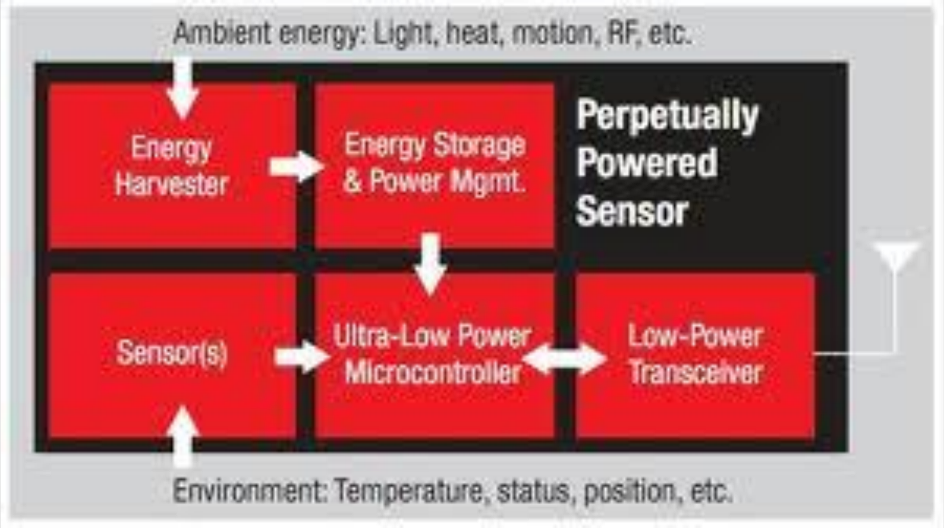
Example: At an average power consumption of 100 mW, with 1 cm3 of lithium battery volume , operation during 1 year→ not always acceptable

Energy harvesting provides an average of 100 mW/cm3 **indefinitely**

Objectives:

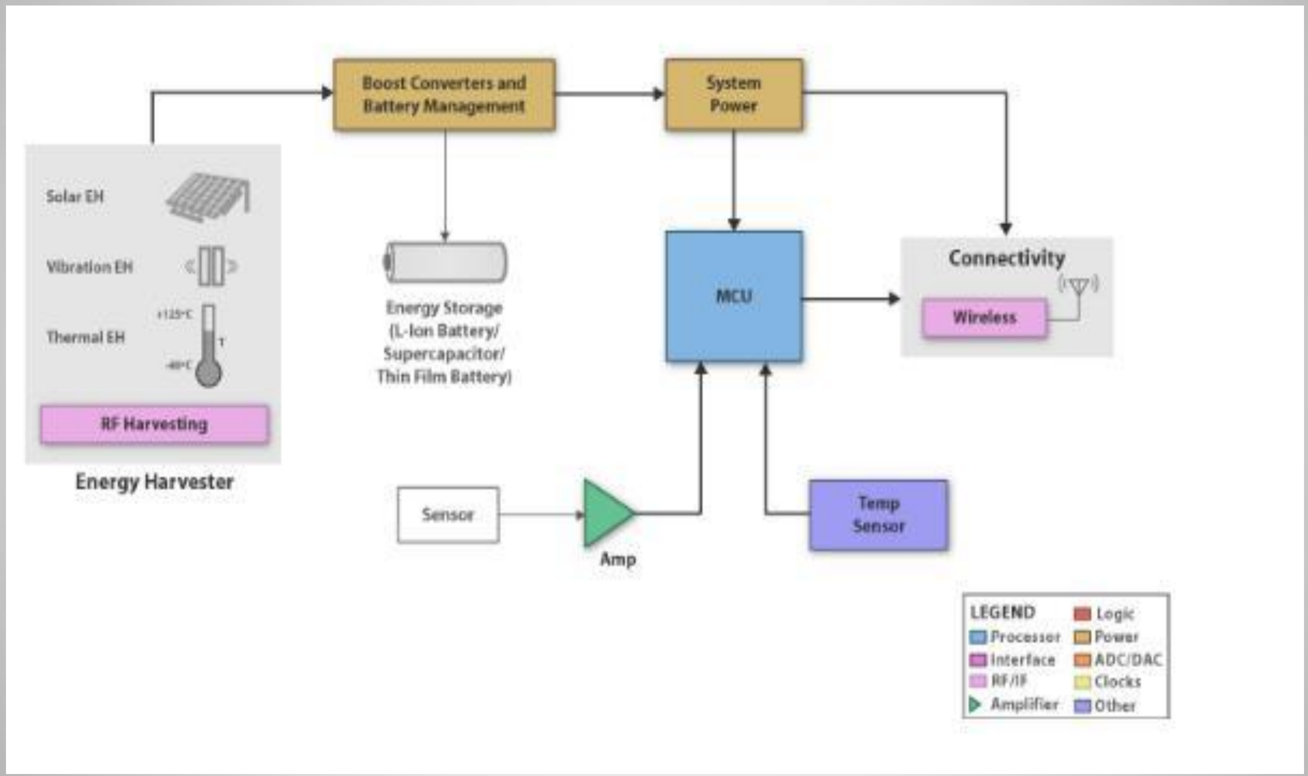
- long life equipment
- Wired power has limitations in out-door applications
- reducing the need for batteries
- Maintenance-free

Anatomy of an energy harvesting system



(Source: Texas Instruments)

Energy Harvesting block diagram



(Source: Texas Instruments)

Characteristics of Wireless Sensors

- In a Wireless Sensor , a task periodically performs the following job:
  - reads data on sensor (temperature, humidity, vehicular movement, lightning condition, pressure, noise levels, …)
  - stores locally the data and perform quick local data processing
  - and sends the data to a base station

Characteristics:

- limited in power, computational capacities, and memory
- dense deployment of disposable and low-cost sensor nodes
- Sensor node lifetime shows a strong dependence on battery lifetime.

→ **Will benefit from energy harvesting**

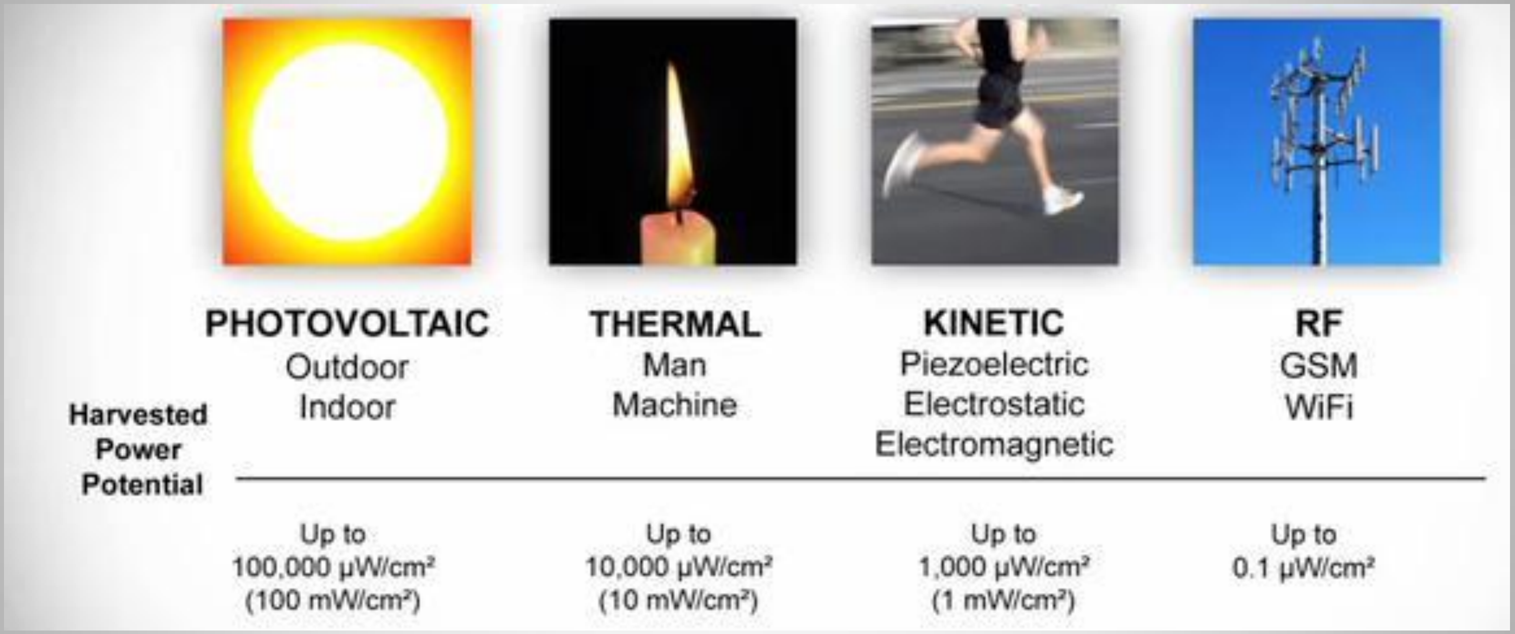
Main application sectors

- **Military** (Battlefield surveillance, Reconnaissance of opposing forces and terrain, Battle damage assessment

- **Environmental** (tracking the movements of animals, Forest fire detection, observation of small size biodiversity, level of air pollution,…)

- **Health** (Telemonitoring of human physiological data,…)

Energy harvesting energy sources



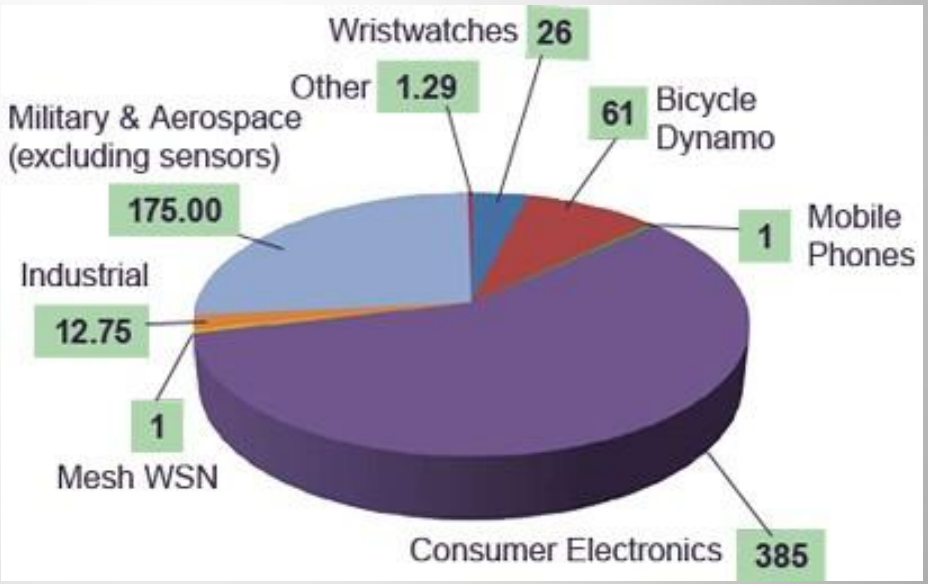
Market Segments using Energy Harvesting

- Mobile phones, Wireless sensor networks, Healthcare Implants,…

250 million wireless sensors
deployed in 2021,
powered by energy harvesting
(1.5 million in 2011)

(Numbers in millions of $)

(Market=$4.4 billion by 2021)



Source: IDTechEx (Energy Harvesting & Storage for Electronic Devices, 2011-2012)

Key specifications for energy harvesting

Source: Silicon Labs

Energy Harvesting becomes reality

New companies with names like *AdaptivEnergy*, *EnOcean*,*Cymbet and Perpetuum*, among others. Are especialized in energy harvesting systems.

And big electronics' manufacturers, such as *Texas Instrument* and *Analog Devic Inc*, are building microcontrollers, digital signal processors and sensors for these applications.

Example of existing renewable energy applications

- –Bridges and buildings monitor structural health with battery-less sensors powered by photovoltaic cells
- –Farmers check soil moisture with solar powered devices
- –Mechanical vibrations can be used for monitoring a vehicle (for example: door lock switches)

Characteristics of these applications: they do not use wires or batteries, while at the same time they save power

Energy Harvesting becomes reality

CBC-EVAL-09 is a universal energy harvesting (EH) evaluation kit that combines any one of multiple EH transducers with the EnerChip TM EP CBC915-ACA Energy Processor and the EnerChip CBC51100 100uAh solid state battery module that has two 50µAh EnerChip solid state batteries connected in parallel.

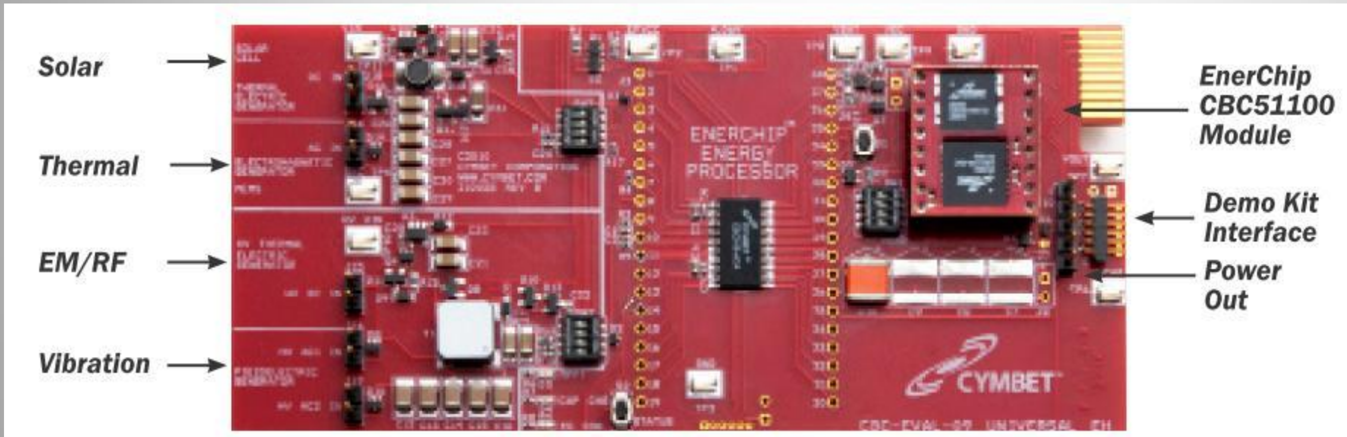


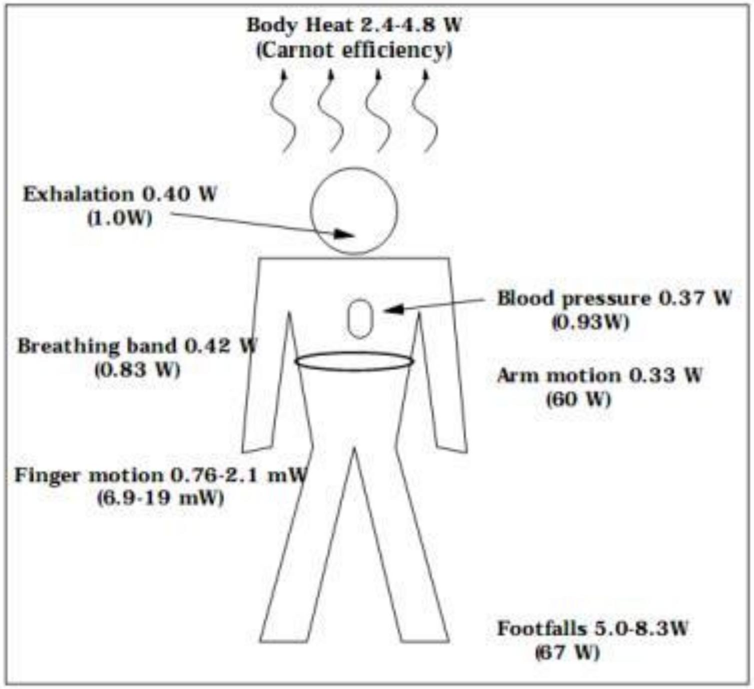
Figure 1: CBC-EVAL-09 Demo Kit - 5 x 2 inches

Source:
Cymbet Corporation

The purpose of this evaluation platform is to enable designers to quickly develop Energy Harvesting applications.

Available human power sources

Activity	Kilocal/hr	Watts
sleeping	70	81
lying quietly	80	93
sitting	100	116
standing at ease	110	128
conversation	110	128
eating meal	110	128
strolling	140	163
driving car	140	163
playing violin or piano	140	163
housekeeping	150	175
carpentry	230	268
hiking, 4 mph	350	407
swimming	500	582
mountain climbing	600	698
long distance run	900	1,048
sprinting	1,400	1,630



Possible power recovery from body-centered sources.
Total power for each action is included in parentheses

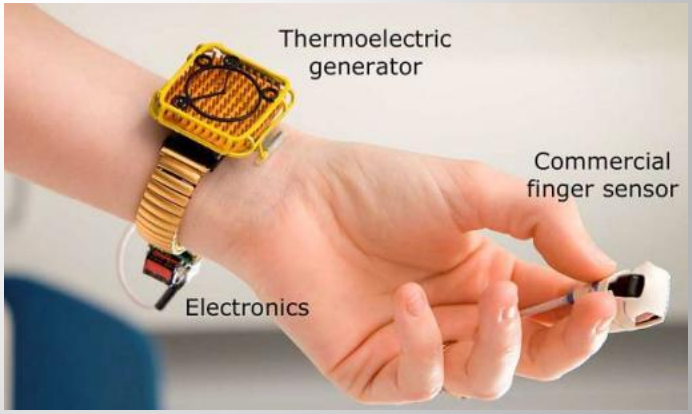
Source: “Human Generated Power for Mobile Electronics,” Shad Starner, Joseph A. Paradiso

Available human power sources

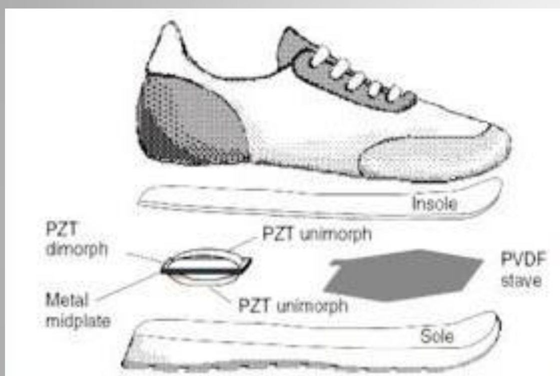
Thermal energy harvesting is usually achieved through the thermoelectric effect. It requires a thermal gradient. It can be achieved in the form of a wearable device.

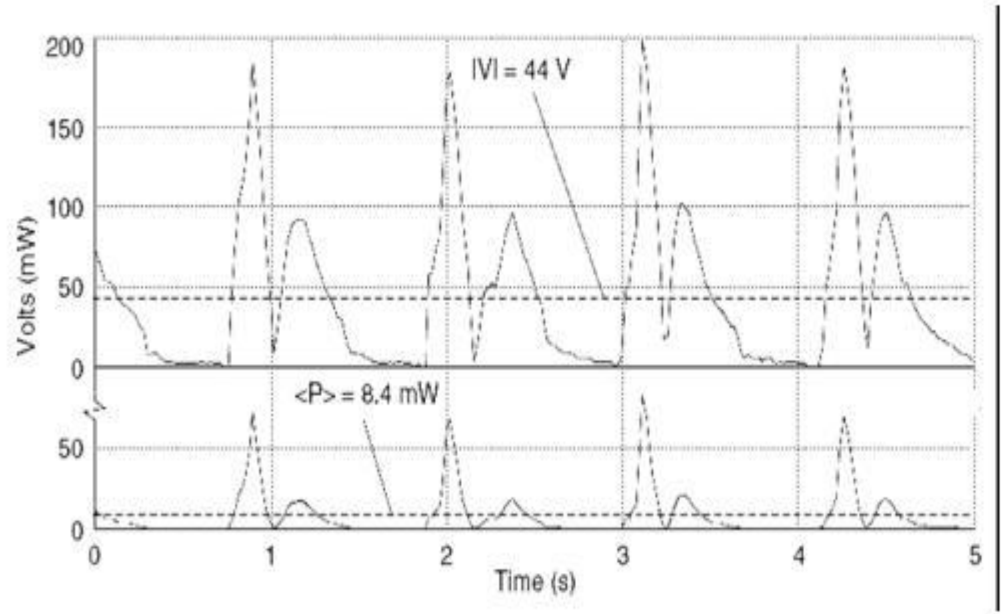
Output of a "normal" adult body is around 100W

Source: Seiko Thermic watch,
a commercially available timepiece
(www.seikowatches.com)



Energy from walking





Example of voltage and power generated

) Nathan S. Shenck, Joseph A. Paradiso, "Energy Scavenging With Shoe-Mounted Piezoelectrics", Publications IEEE, 2001

Two kinds of energy harvesting systems

Energy harvesting can be divided into two architectures:

- **Harvest-Use**: Energy is harvested just-in-time for use

  Example:
  The switches draw power from the mechanical energy that's generated when people press the light switch. Wireless signals are transmitted over an RF frequency

- **Harvest-Store-Use**: Energy is harvested whenever possible and stored for future use.

Energy storage is useful when the harvested energy available is more than the current usage

Example:
During the daytime, energy is used for work and also stored for later use. During night, the stored energy is conservatively used to power the sensor node.

Energy storage

Almost all energy-harvesting systems require an energy storage element or buffer.

Even if the power consumed by an embedded application is so low as to be run directly on power captured or scavenged from the environment, such power would not be produced in a constant way.

This means that an energy storage (**reservoir**) is needed.

Storage is implemented in the form of

- a **capacitor**
- or/ and a rechargeable **battery**,

What kind of energy storage is needed depends greatly on the application.

Thin-film batteries

Some applications require power for only a very short period of time, as short as the RC time constant discharge rate of a capacitor.

Other applications require relatively large amounts of power for an extended duration, which dictates the use of a traditional AA or a rechargeable lithium battery.

Still other applications need the small footprint benefit of the capacitor and the low energy leakage advantage of a tradition battery.
This is where the thin-film batteries are gaining acceptance

	Li-Ion Battery	Thin Film Battery	Super Cap
Recharge cycles	Hundreds	Thousands	Millions
Self-discharge	Moderate	Negligible	High
Charge Time	Hours	Minutes	Sec-minutes
Physical Size	Large	Small	Medium
Capacity	0.3-2500 mAHr	12-1000 µAHr	10-100 µAHr
Environmental Impact	High	Minimal	Minimal

Characteristics of typical energy storage options (Courtesy of TI)

(Courtesy of TI)

Framework of an energy harvesting system

A typical energy harvesting system has four components,

- the **Energy source**: ambient source of energy to be harvested
- the **Harvesting architecture** : mechanisms to harness and convert the input ambient energy to electrical energy
- The **Energy storage:** to temporarily store the harvested energy
- and the **Load**: activity that consumes energy .

Weakness of current systems

Most embedded systems constructed to date do not extract power efficiently from the source.
Consequently, this approach has disadvantages, such as high costs and large space.

They use a much larger harvester (e.g. solar panel) than necessary to yield the same level of power as a more efficient one.

They rely on a larger, more expensive, higher capacity battery than needed in order to sustain extended operation.

Issues:

•Need for **a methodology** for designing a Real-Time Energy Harvesting System in order to determine

- ❑The best suitable energy storage unit
- ❑The best suitable harvester

Energy consumption of WSN

The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains:

sensing, communication, and data processing.

A sensor node expends maximum energy in data communication.
This involves both data transmission and reception.

It can be shown that for short-range communication with low radiation power, transmission and reception energy costs are nearly the same.

Energy consumption in data processing

Energy expenditure in data processing is much less compared to data communication.

Example: *(I.F. Akyildiz, W. Su , Y. Sankarasubramaniam, E. Cayirci Computer Networks 38 (2002) 393–422)*

“Energy cost of transmitting 1 KB a distance of 100 m is approximately the same as Energy cost of executing 3 million instructions by a 100 million Instructions per second (MIPS) processor”.

Design objectives

Power management considerations are very different from those of maximizing lifetime i.e. **Power-aware ≠ low-power**

To execute, each task requires time and energy.

Our goal is **to meet deadlines requirements** taking into account:

1. **demands of energy and processing time** of all the tasks

2. **Limited availability of energy and processing time** due to
   - the energy reservoir (bounded capacity)
   - the energy source (variable and limited charging power)
   - the processor (limited speed)

Our primary goal is not to minimize energy consumption.

Design objectives

1) to operate in an *energy neutral* mode, consuming only as much energy as harvested.
Such a mode of operation raises the possibility of indefinitely long lifetime, limited only by the hardware longevity.

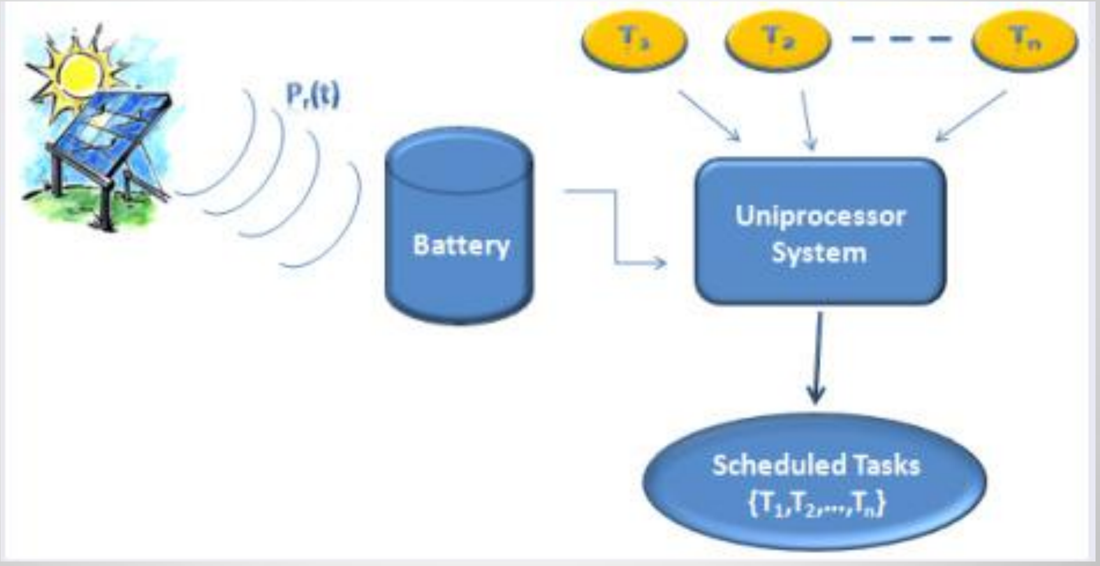
Question: How to operate such that the energy used is always less than the energy harvested?

2) to deal with the real-time tasks (with timing constraints) under the strong variation of energy source with respect to time

Question: What is the maximum performance level that can be supported (in terms of deadline success) ?

Third part:
Scheduling
for Energy Harvesting Systems

A generic energy harvesting system



Assumptions

Energy Storage

A nominal capacity :At any time, the stored energy is no more than the storage capacity, that is $E(t) < E$ $E = E_{max} - E_{min}$

– Recharching and discharging may overlap

Energy Source

– Energy produced with fluctuating power $P_r(t)$

– We define the WCCR (Worst Case Charging Rate), namely $P_r(t)$, which is a lower bound on the harvested source power output.

The energy drawn from the environment is **uncontrolled but predictable**

(**for example,** the intensity of direct sunlight cannot be controlled but it is predictable with daily and seasonal patterns and we can use prediction algorithms)

Assumptions

Task Set: each task is characterized by :

- **Worst Case Execution Time** (WCET).
- **Worst Case Energy Consumption** (WCEC)
- the **deadline** and **period** respectively.

Characteristics of the tasks are known in advance.
Preemption is allowed and tasks execute with fluctuating consumption power .
WCEC is not necessary proportional to WCET

Objective: To satisfy timing constraints on software execution

A new terminology

A scheduler S is said to be optimal if, given an energy source and an energy reservoir, if S cannot produce a valid schedule then no other one can do.

A scheduler S is said to be energy-clairvoyant if it needs the energy profile for the future

A schedule Γ for τ is said to be valid if the deadlines of all tasks of τ are met in Γ, starting with a storage fully charged.

A task set τ is said to be timely-feasible if there exists a valid schedule for τ without considering its energy constraints.

A task set τ is said to be feasible if there exists a valid schedule for τ with considering its energy constraints.

Summary of initial related works

DVS Approach **(University of Pittsburg, USA)**

- A. Allavena and D. Mossé, Scheduling of Frame based Embedded Systems with Rechargeable Batteries, Workshop on Power Management for Real-Time and Embedded Systems **2001**.

- C. Rusu, R. Melhem and D. Mossé, Multi-version Scheduling in Rechargeable Energy aware Real-time Systems, **ECRTS 2003**

Non DVS Approach

- A. Allavena and D. Mossé, Scheduling of Frame based Embedded Systems with Rechargeable Batteries, Workshop on Power Management for Real-Time and Embedded Systems **2001**.

(Swiss Federal Institute of Technology, ETH Zurich)

- C. Moser, D. Brunelli and L. Benini Real-time Scheduling with Regenerative Energy. **ECRTS 2006**

Related Work (Non DVS)

- Mosse et .al **(2001**) proposed an off-line scheduling algorithm
- Specifically, this algorithm is designed to schedule a set of independent periodic tasks with identical periods (**frame based system**).
- **Problem:** to find a schedule which is able to execute all the tasks within the deadline *D*, starting with a battery fully charged, ending at the same energy level.
- this approach is based on an unpractical assumption that the harvested energy from the ambient energy source is **constant**.
- **Drawback**:
  - Restrictive model (on energy consumption, task timing parameters)
  - Off-line scheduler

Related work

- An **optimal** real-time scheduling algorithm called **lazy scheduling (LSA)** has been proposed (Moser et al, **2006**)

Assumptions:

- Tasks may be periodic or aperiodic.
- Energy loss insignificant

Properties:

- LSA is a variant of EDF but is an idling energy-clairvoyant scheduler

NB: Hui Zhang will defend his PhD on an extensive simulation study of the LSA scheduler within the second trimester 2012

Reference: Performance Evaluation of Real-Time Scheduling Heuristics for Energy Harvesting Systems , Maryline Chetto, Hui Zhang, **GreenCom'10**

The LSA scheduler

- The processor executes all tasks at **full power** (one frequency);
- The system starts executing a task if the following three conditions are met simultaneously:

    1) The task is ready;

    2) The task has the earliest deadline among all ready tasks

    3) The system is able to keep on running at the maximum power until the deadline of the task.

Drawbacks v.s. advantages of LSA

Advantages:

- Optimality
- A general schedulability test
- On-line
- Energy-clairvoyant

Drawbacks:

- WCEC is proportional to WCET. The ratio is given by the speed of the processor
- Optimality is true only if the processor is able to adjust its consumption power to the source power

→ **Proposition of a novel on-line scheduler**

EDeg : A new scheduler

Energy demand in $[t_1, t_2[$ $$g(t_1,t_2) = \sum_{D_i < t_2 - t_1} \left(1 + \left\lfloor \frac{t_2 - t_1 - D_i}{T_i} \right\rfloor\right) * E_i$$

Total quantity of energy required by tasks between t_1 and t_2.

Slack energy of a job J: $slack.energy(J_i, t) = E(t) + E_s(t, d_i) - g(t, d_i)$

Maximum amount of energy that can be consumed from current time t while still meeting the deadline d of the job (consequently after having executed all tasks with a priority higher than or equal to J)

Slack energy of the system: Minimum slack energy among all the jobs

Slack energy (t) <0 means that the processor has to idle from t

so as to guarantee sufficient energy for executing all future tasks before deadline.

EDeg Pseudo-code

Algorithm 1 Earliest Deadline with energy guarantee algorithm (EDeg)

```
while (1) do
  while PENDING=true do
    while (E(t) > E_min and Slack.energy(t) > 0) do
      execute()
    end while
    while (E(t) < E_max and Slack.time(t) > 0) do
      wait()
    end while
  end while
  while PENDING=false do
    wait()
  end while
end while
```

Properties of EDeg Scheduler

- EDeg degenerates to an EDS (ASAP) policy if E_{max} = 0 (i.e. no reservoir)

- **Two central rules:**

1. **The highest priority task is executed as long as there is slack energy.**
2. **The processor is idle (for recharging energy) as long as there is slack time od the reservoir is full**

- Slack time (t) =0 means that the processor has to be busy from t so as to guarantee all future tasks before deadline.

- We only waste recharging power when there are no pending tasks and the storage unit is full.

Reference: A real-time scheduling framework for embedded systems with environmental energy harvesting Hussein EL Ghor, Maryline Chetto, Rafic Hage Chehade, **Computers & Electrical Engineering, Vol. 37, Iss. 4, 2011**

Illustrative example

Let the periodic task set given by

$\tau_1 = (3, 6, 9, 8)$, $\tau_2 = (3, 8, 12, 8)$ and $\tau_3 = (3, 12, 18, 8)$

We assume that the energy storage capacity is E = 6.

For simplicity, the rechargeable power,Pr is constant along time and is equal to 2.

We note that Up = 0.75 and Ue = 2.

As Up $\leq$ 1 and Ue $\leq$ Pr the necessary feasibility condition related to time and energy are satisfied.

Illustrative example

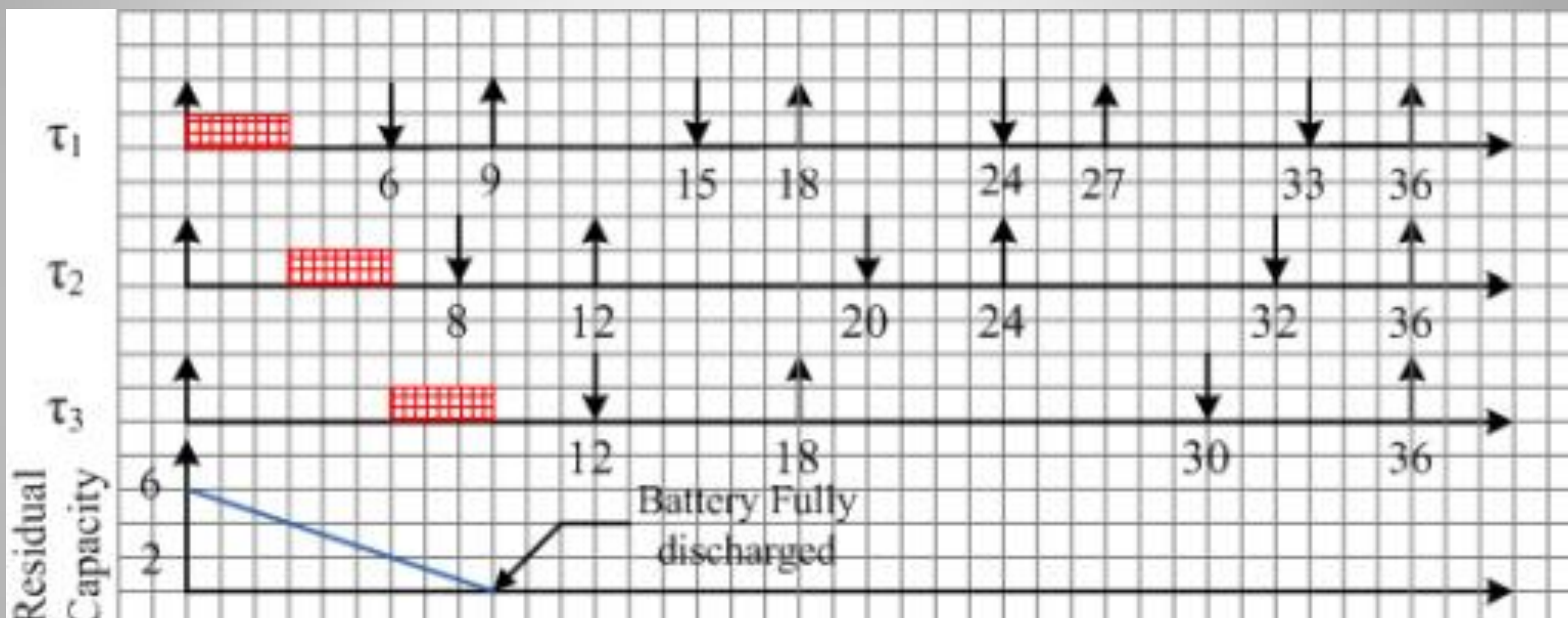


Figure: Weakness of the EDS (ASAP) scheduling strategy

Illustrative example

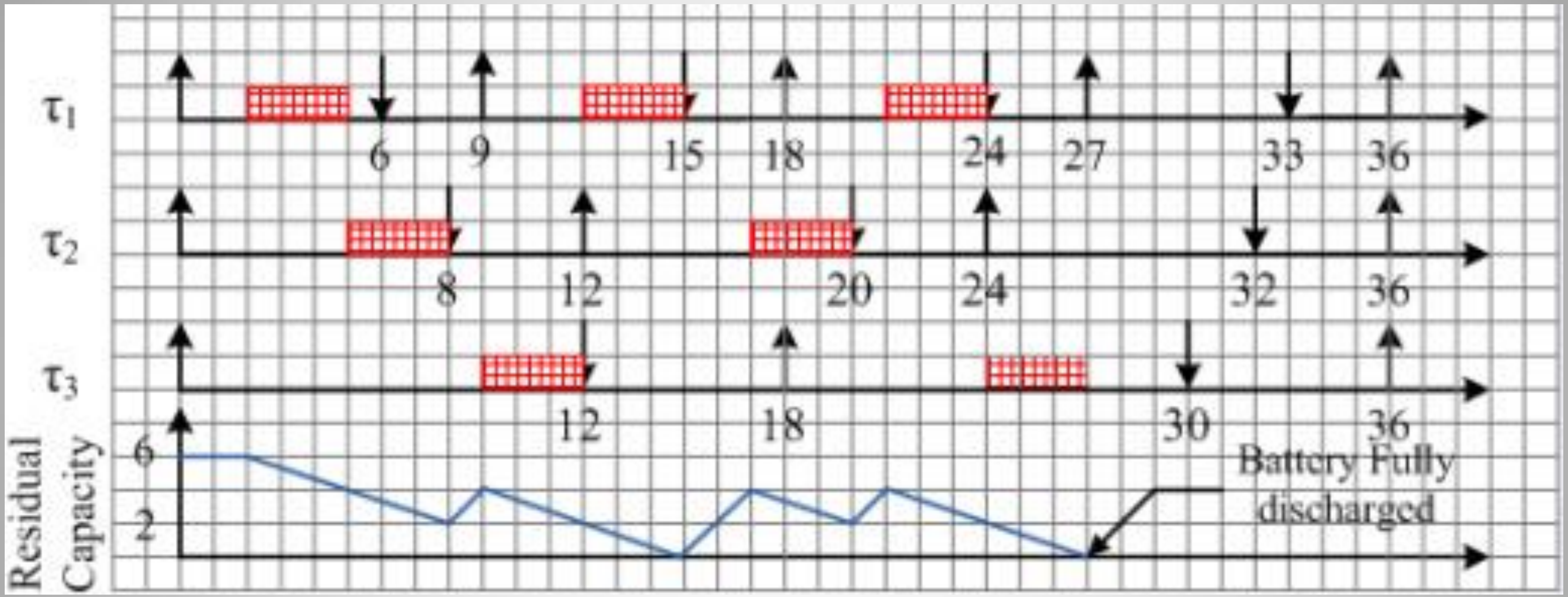


Figure: Weakness of the EDL (ALAP) scheduling strategy

Illustrative example

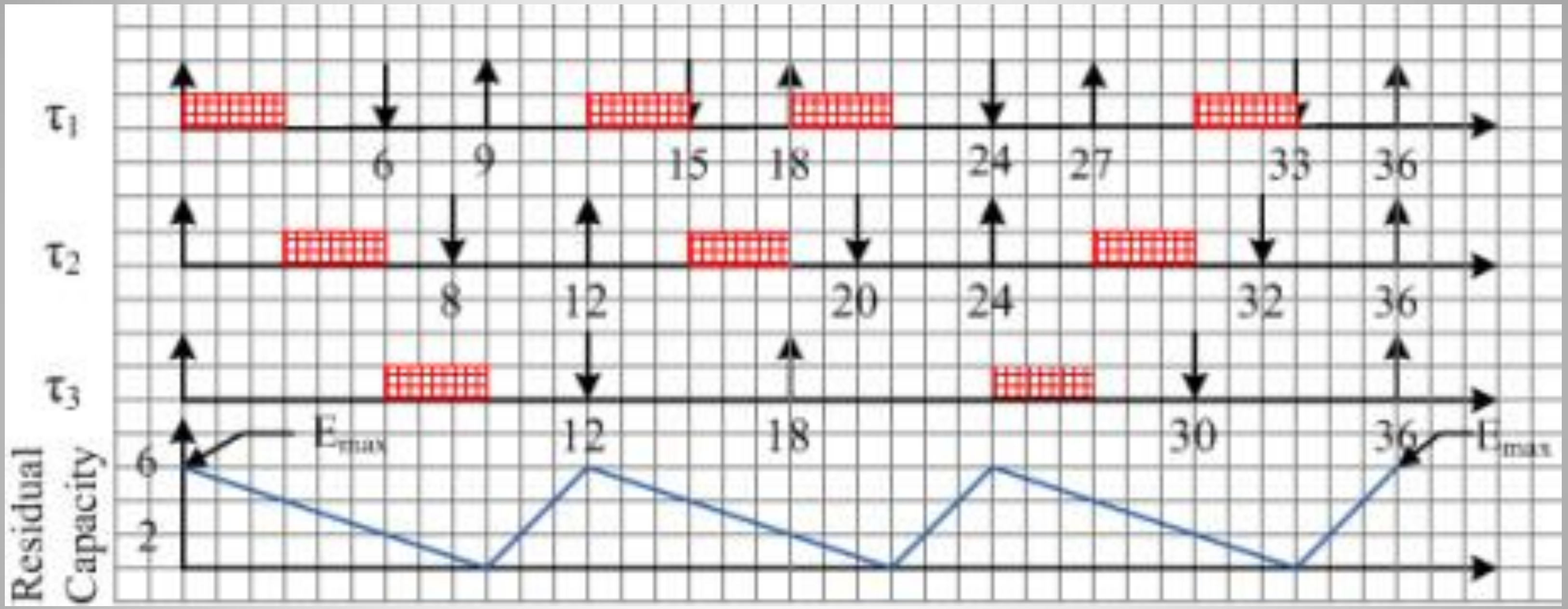


Figure: EDeg scheduling strategy

Comments on EDeg

EDeg has been designed to schedule :

- any set of time critical tasks (periodic or not)
- given any energy source profile (with constant production power or not)
- and given an energy storage unit with limited capacity.

BUT

EDeg is a **CLAIRVOYANT scheduler**:

- Energy clairvoyant (must know future energy profile)
- Processing clairvoyant (must know future processor demand)

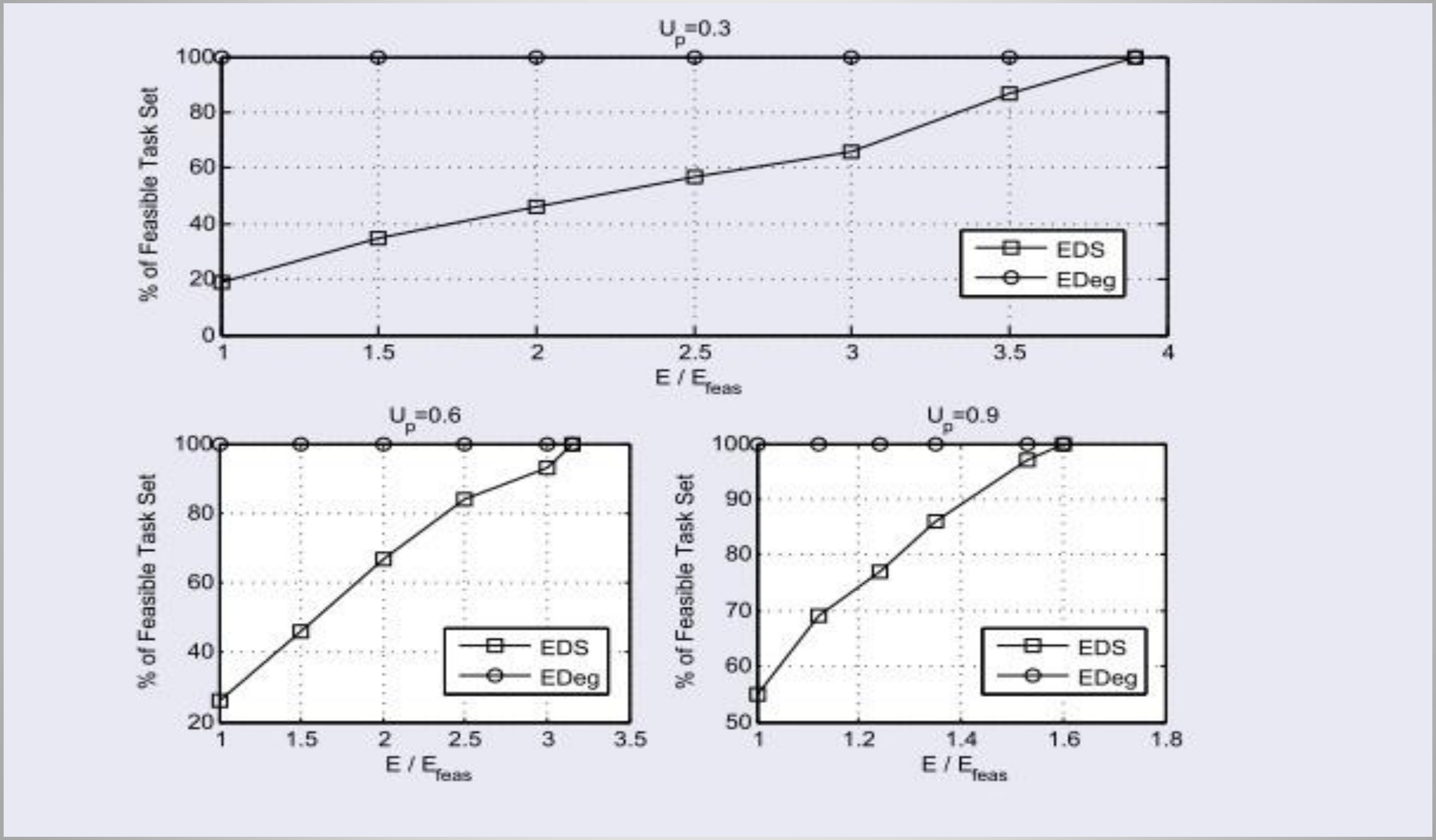
Experimental results about EDeg

To evaluate the effectiveness of EDeg algorithm, we compare it with EDS, EDL and two heuristics EDd 1 and EDd A.

Edd_A is the Earliest Deadline as Soon as possible scheduler that discards ALL the ready instances whenever the storage unit is empty and consequently let the processor idle until the next release time.

EDd 1 is the Earliest Deadline as Soon as possible scheduler that discards only one instance (the highest priority one) whenever the storage unit is empty and then let the processor idle until the next release time.

Experiment 1: Feasibility

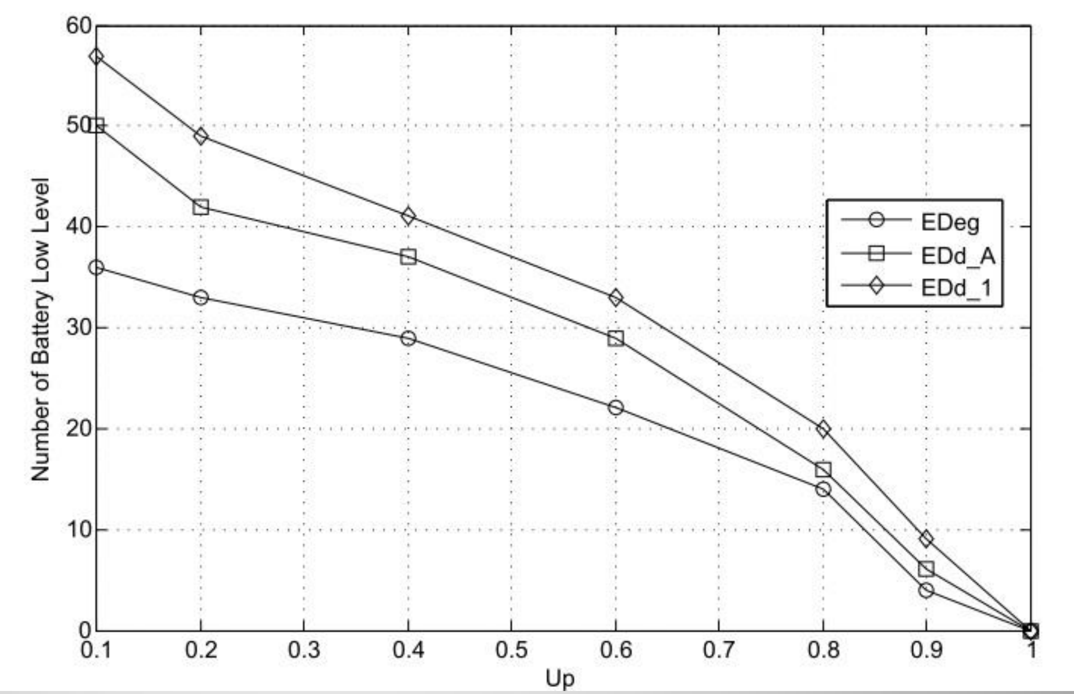


Percentage of feasible task sets

Experiment 2: Battery low level

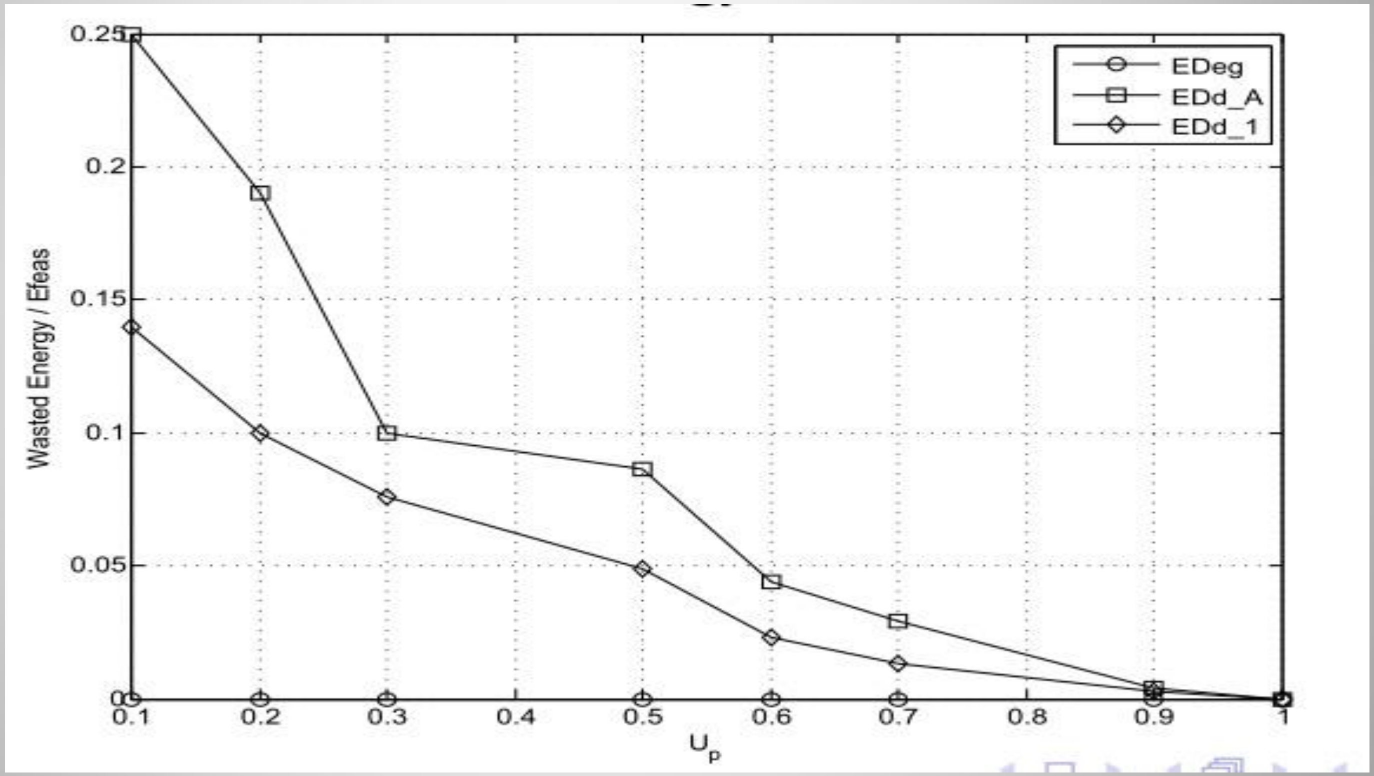
This experiment shows the number of times when the battery becomes empty i.e. number of times the battery is lack of energy and needs to recharge.

The battery low level with EDeg is lower than wit EDd A and EDd 1.



Experiment 3: Wasted energy

Wasted energy = energy which is consumed by tasks that do not complete before deadline.

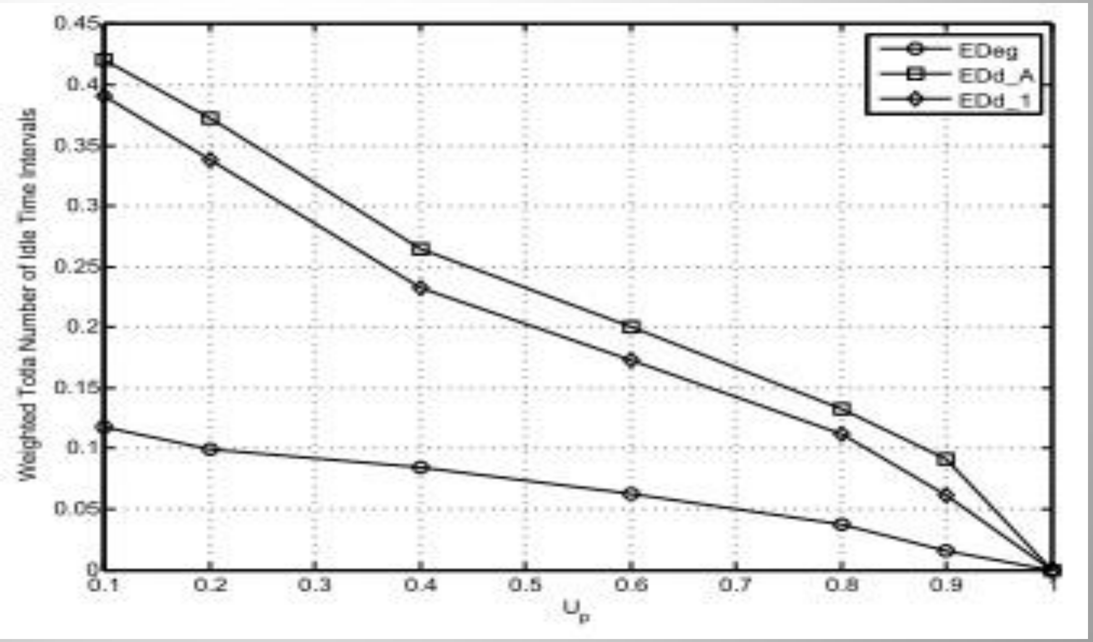


Experiment 4: Idle times

We measure the total number of idle time intervals over the total number of feasible instances.

With DPM (Dynamic Power Management) , idle time means to transfer processor into sleep mode and then into active mode. → cost in time and energy

So we need a schedule
which minimizes
the number of idle times



RTOS Requirements for EDeg

Adding Energy harvesting aware features to a RTOS requires the knowledge of:

- the current available energy
- the estimation of the future harvested power to tune the system behavior
- the energy requirement of every task.

Here are the main technological obstacles !!!

Summary

- Our scheduler EDeg is model-free with respect to the energy source.
- It can be implemented in any energy harvesting system without the need for a priori information about the source which may be uncontrollable and time-varying.
- The scheduler is based on the on-line computation of the **slack time** and the **slack energy** which is a new concept dedicated to hard deadline tasks with regenerative energy constraints.

Current extensions

1) **EDeg with**

- **Quality of Service** requirements (Skip Over)

  2 PhD students since 2010 (Uniprocessor and multiprocessor)

- **Fault-Tolerance** (Deadline Mechanism)

  1 PhD student since beginning of 2012

- ***Aperiodic task servicing***

2) Non clairvoyant scheduling

Issue: How to schedule the tasks with no prediction on:

- future incoming energy
- Future processing load

We propose an on-line scheduler called **EH-EDF.**

3) Same scheduling framework for fixed priority systems

→ DMeg, RMeg,...

4) A generic Scheduling framework

Last communications

- **Energy harvesting scheduling for fixed priority systems**

Maryline Chetto, Damien Masson, and Serge Midonnet. Fixed priority Scheduling strategies for Ambient Energy-Harvesting embedded systems, **GreenCom'11**

- **Non clairvoyant Energy harvesting scheduling for dynamic priority systems**

Hussein El Ghor, Maryline Chetto, Rafic Hage Chehade, EH-EDF: an on-Line Scheduler for Real-Time Energy Harvesting Systems **ICECS 2011**

MERCI

DE

VOTRE

ATTENTION