ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTA' DI INGEGNERIA

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

METODOLOGIE DI PROGETTAZIONE HARDWARE E SOFTWARE L-S

TECNICHE DI GESTIONE DI ENERGIA SOLARE IN SISTEMI EMBEDDED DI ATTUAZIONE

-

MANAGEMENT OF SOLAR HARVESTED ENERGY IN ACTUATION-BASED EMBEDDED SYSTEMS

Tesi di Laurea di:

CARLO BERGONZINI

Relatore:

Chiar.mo Prof. Ing. LUCA BENINI

Correlatori:

Chiar.ma Prof.ssa TAJANA SIMUNIC
Ing. DAVIDE BRUNELLI
Ing. JOAQUIN RECAS PIORNO

Anno Accademico 2007/2008

Sessione III

Index

Introduzione

Negli ultimi anni, eventi drammatici che hanno messo a rischio la vita di persone, come il crollo del ponte sul fiume Mississippi del 2007, ma anche attacchi terroristici o terremoti, hanno messo in evidenza alcune problematiche che hanno dato una forte spinta allo sviluppo di *structural health monitoring* (SHM) soprattutto oltreoceano.

Lo sviluppo di questo lavoro di tesi svolto presso la University of California, San Diego, nasce dallo studio di alcune applicazione della ricerca attuale in ambito informatico ed elettronico correlate allo SHM. A partire dall'analisi di queste applicazioni è stato seguito un approccio top-down per analizzare in particolare 3 aree di interesse principali che hanno portato alla luce le problematiche da cui è stata sviluppata questa ricerca. Queste aree sono:

1. Lo structural health monitoring
2. Le reti di sensori wireless
3. L'energy harvesting con particolare attenzione rivolta all'energia solare e alla sua gestione.

Lo SHM rappresenta una grande sfida che unisce conoscenze dell'ingegneria strutturale, informatica ed elettronica. L'obiettivo è fornire un monitoraggio attivo sulle strutture in grado di rilevare ed eventualmente localizzare discontinuità o danneggiamenti all'interno di materiali omogenei. Questa disciplina è estremamente utile sia per evidenziare quali siano stati i danni inferti ad una struttura da eventi esterni, come appunto un terremoto o un esplosione, sia per contribuire a rilevare lo stato di deterioramento del materiali costituenti le strutture a distanza di anni dalla loro costruzione. La sfida risiede quindi nel trovare tecnologie in grado di produrre un controllo continuo, installando anche sensori in posizioni estremamente difficili da raggiungere. Questi sensori necessitano inoltre di un livello minimo di manutenzione ed un alto grado di autosufficienza, garantendo tuttavia un supporto alla comunicazione semplice e rapido. In questo ambito risulta quasi indispensabile l'utilizzo di sensori wireless e quindi porre una particolare attenzione alle implicazioni connesse con l'introduzione di questo tipo di tecnologia.

Sempre più vasta è l'area di ricerca che negli ultimi anni si è occupata di reti di sensori wireless e che è stata ampiamente studiata per durante questo lavoro di tesi. Gli studi effettuati a riguardo consigliano soluzioni innovative in grado di ottimizzare i consumi ed agevolare la comunicazione. Recenti sviluppi delle wireless sensor network (WSN) permettono di creare reti estese di sensori che, tramite protocolli di ruolo, riescono a fare pervenire le informazioni al destinatario limitando al massimo sia il consumo di energia sia l'overhead di comunicazione e consentono il raggiungimento di nodi che si trovano in posizioni inaccessibili. Il sempre più rapido avanzamento delle tecnologie nei sensori ha consentito negli ultimi anni un continuo miglioramento delle prestazioni, ottenuto in parallelo ad una forte riduzione delle dimensioni dei dispositivi e del costo energetico relativo. Questo ha consentito di includere funzionalità sempre più innovative come capacità attuative presenti nelle nuove generazioni di sensori. Per attuazione si intende, in questo caso, capacità di movimento autonomo, oppure, nel caso di alcune applicazioni di SHM, la capacità di generare onde ad alta energia per rilevare discontinuità nelle strutture.

A questo punto risulta facile intuire come le risorse di energia diventino estremamente importanti. Le batterie da sole faticano a garantire ai sistemi di attuazione una durata sufficiente e diventa quindi indispensabile ricercare in altre fonti rinnovabili l'energia necessaria per prolungare la vita di questi tipi di dispositivi. Sempre più applicazioni delle WSN

investigano su nuove tecnologie di Energy Harvesting. Queste tecnologie sfruttano la possibilità di raccogliere dall'ambiente l'energia richiesta per alimentare i dispositivi. Attraverso alcuni studi si sono sviluppate tecnologie per ricavare l'energia dalle vibrazioni tramite dispositivi piezoelettrici, dal riscaldamento naturale o da fonti eoliche, ma la fonte che in assoluto garantisce una maggiore densità di potenza è l'energia solare. Tramite l'energia solare di può ricavare infatti un'alta quantità di energia anche da pannelli solari di dimensioni relativamente limitate ed in questo modo si può fornire ai dispositivi o ai sensori l'energia utile a portare a termine i propri obiettivi ed estendere indefinitamente la durata di vita di questi dispositivi.
L'energia solare ha permesso di alimentare un numero elevato di applicazioni già realizzate con ottimi risultati, ma l'utilizzo di essa determina una serie di problematiche, da considerare in fase di progettazione, dovute alla periodicità e all'imprevedibilità della sua disponibilità.

Proprio dalle sfide che queste problematiche hanno messo in luce nasce il mio lavoro di tesi sviluppato presso il dipartimento di Computer Science Engineering di UCSD. L'obiettivo del lavoro è stato individuato prendendo in considerazione una piattaforma precedentemente sviluppata da una collaborazione tra il Micrel Lab dell'Università di Bologna e l'Università di San Diego. Lo SHiMmer [2] è una piattaforma di attuazione per lo structural health monitoring completamente autosufficiente alimentata ad energia solare ed utilizza un supercapacitor per immagazzinare l'energia. Principalmente questa piattaforma è in grado di generare un onda ad alta energia per il monitoraggio delle strutture tramite sensori piezoelettrici, elaborare i dati ottenuti utilizzando algoritmi anche ad alto costo computazionale e comunicare i risultati ad agenti esterni (Unmanned Aerial Vehicle) tramite una protocollo di comunicazione wireless. Tuttavia questi compiti sono effettuati solamente dopo una stimolazione dall'esterno tramite radio triggering. Il sistema non è quindi del tutto autosufficiente ma ha bisogno di un agente esterno in grado di attivarlo da uno stato di riposo e richiedergli di compiere determinati task.

Dall'analisi di questo sistema sono sorte le prime domande che hanno condotto allo svolgimento di questa tesi. In particolare ci si è chiesti perché dover aspettare un agente esterno per compiere dei lavori. Se l'agente esterno non arriva per lungo tempo, una grande quantità di energia viene sprecata in quanto il supercapacitor offre una capacità estremamente limitata. Questa energia al contrario può essere sfruttata per compiere delle rilevazioni dei dati e per poter mandare il risultato già elaborato all'agente esterno. Bisogna tuttavia tener conto del fatto che utilizzando tutta l'energia entrante nel sistema si può incorrere nell'assenza totale di energia, che potrebbe invece essere utile in caso di particolari richieste esterne ad alta priorità, oppure per comunicare via radio dati già elaborati. L'alta imprevedibilità delle condizioni solari porta ad una serie di valutazioni indispensabili per sviluppare un sistema affidabile e completo.

Il lavoro sviluppato ha dato luogo alla nascita di una serie di strategie per poter migliorare la piattaforma SHiMmer e generalizzare i problemi per estendere un modello in grado di includere tutti quei sistemi che presentano simili vincoli.

In primo luogo è stata ricercata una metodologia per svegliare il sistema in una condizione di carica completa, evitando un controllo periodico frequente dell'energy storage unit.

Un algoritmo di predizione di energia solare, il Weather Conditioned Moving Average, è stato quindi sviluppato con lo scopo di fornire una previsione dell'energia in ingresso al sistema, per poterne ricavare un tempo di riposo adeguato [1]. Questo algoritmo salva periodicamente i valori di potenza ottenuti dal pannello solare in una tabella circolare dove ogni riga rappresenta un giorno passato. Utilizza poi i valori passati per dare una predizione di quello che sarà il valore futuro. Per ottenere ciò sono state fatte considerazioni sulle variazioni stagionali degli orari di alba e tramonto, ma nello stesso tempo considerazioni sulla varianza tra i valori passati nel giorno attuale ed i corrispondenti valori nei giorni passati. In questo modo si ottiene un'indicazione sulla risposta del pannello solare comparata ai giorni precedenti e si è quindi in grado di adattare la previsione a quelle che sono le condizioni atmosferiche reali.

Una volta ottenuta una previsione dell'energia solare entrante nel sistema è stato necessario creare un modello per poter collegare questo livello di energia ad una stima della velocità di ricarica che essa può fornire. Oltre allo studio della letteratura a riguardo, sono stati effettuati dei test reali per creare una metodologia di realizzazione ad hoc e per creare quindi delle equazioni che siano in grado di relazionare questi valori ad una velocità di ricarica corrispondente ed ottenere in fine la stima di un tempo di ricarica. Data una previsione di energia ed un valore attuale di carica, il sistema sarà quindi in grado di calcolare un tempo di riposo dopo il quale il livello dell'energy storage unit avrà probabilmente raggiunto la sua capacità massima o il livello energetico richiesto.

A questo punto sono state elaborate delle strategie in grado di permettere al sistema di mantenere sempre una certa quantità di energia per poter eseguire compiti di diversa priorità, ma anche per permettere al sistema di "sopravvivere" alle ore notturne o a periodi di scarsa irradiazione solare.

I 3 task principali dello SHiMmer sono stati individuati per estendere il modello ai sistemi con simili caratteristiche. Sistemi che presentano, in primo luogo un active sensing o comunque un task di attuazione ad alto costo energetico, in secondo luogo una fase di elaborazione dei dati ed, infine, una modalità di comunicazione via radio con l'esterno. Sono state quindi elaborate strategie per l'attivazione e il coordinamento delle attività di active sensing, di elaborazione dati e di comunicazione. Queste strategie danno la possibilità di cambiare le priorità che regolano questi task per prevenire problematiche di gestione dell'energia e della memoria. A seconda della disponibilità di energia e memoria infatti possono essere fatte scelte diverse. L'attività di elaborazione dati, ad esempio, può dare vantaggi in quanto consente eventualmente di eliminare dalla memoria l'intera onda acquisita e sostituirla con un'onda risultante che occupa uno spazio più limitato di memoria. Valutazioni di tipo simile possono essere fatte per quanto riguarda la comunicazione. L'elaborazione può ridurre il costo di comunicazione anche in termini di tempo ma, se ha un costo energetico troppo elevato, può essere preferibile inviare l'intera onda ottenuta e relegare all'agente esterno il compito di elaborare i dati.

In seguito sono stati definiti diversi profili energetici. In particolare nella piattaforma di riferimento sono stati definiti un profilo con un alto livello energetico ed uno a consumo limitato. Nelle simulazioni effettuate sono state fatte delle scelte che comportano l'utilizzo del profilo alto come default, in quanto garantisce una maggiore precisione ed affidabilità del risultato. Quando però le condizioni solari, collegate alla previsione ottenuta, indicano scarsità di irradiazione futura, il sistema si converte ad un basso livello energetico che consente l'ottenimento di un risultato utilizzando una quantità minima di energia, a discapito di una minore precisione.

Infine è stata eseguita un'analisi teorica delle code di task. Questa ha portato allo sviluppo di una serie di simulazioni in grado di dimostrare i benefici che l'insieme degli algoritmi e le strategie sviluppate sono in grado di apportare alla piattaforma SHiMmer e tutti i sistemi che presentano caratteristiche equivalenti.

Nella redazione di questa Tesi si è mantenuto l'ordine logico secondo cui si è sviluppato l'intero lavoro.

Nel Capitolo 1 vengono illustrate le maggiori tecnologie attualmente utilizzate per quanto riguarda l'energy harvesting e le tecniche di gestione dell'energia ad esse correlate.

Il Secondo Capitolo descrive le problematiche introdotte dallo studio dello Structural Health Monitoring focalizzando l'attenzione sulla piattaforma SHiMmer, utilizzata come riferimento per i successivi lavori.

Il Capitolo 3 introduce le problematiche affrontate che hanno portato alla suddivisione dei contributi in 3 diverse sottoparti.

L'algoritmo di predizione dell'energia solare (WCMA) è illustrato nel Capitolo 4 con diversi confronti con algoritmi con obiettivi equivalenti.

Nel Capitolo 5 si è introdotto lo studio del Recharge Estimator che è stato utilizzato per calcolare la velocità di carica e quindi il tempo di riposo.

Il Capitolo 6 descrive l'analisi teorica delle code e lo sviluppo di strategie di scheduling prioritarie e di divisione in profili energetici con le relative simulazioni.

Il Capitolo finale illustra i benefici ottenibili con l'utilizzo di queste tecniche ed i possibili sviluppi futuri.

Introduction

In the last years tragic events that have endangered people's lives, such as the collapse of the bridge on Mississippi in 2007 and also terrorist attacks or earthquakes, have pointed up some problems giving a major boost to the development of *structural health monitoring* (SHM) especially in the U.S.A..

The development of this work was born at the University of California, San Diego from the study of some applications of present researches concerning computer science and electronics related to SHM. Starting from the analysis of these applications, a top-down approach has been followed in order to particularly analyze 3 main areas of interest that have highlighted the problems from which this research has been developed. The areas are the following:

1. Structural health monitoring
2. Wireless sensors networks
3. Energy harvesting focusing attention on solar energy and its management

SHM is a great challenge that combines the knowledge of structural, computer science and electronics engineering in order to supply an active monitoring on structures that can detect and maybe localize discontinuities or damages inside homogeneous materials. This is extremely useful both to show the damages caused to a structure by external events, such as an earthquake or an explosion, and to contribute to detect the state of deterioration of materials after years from their construction. The challenge is therefore finding new technologies able to carry out a continuous check, also installing sensors in positions extremely difficult to reach. These sensors therefore need a minimum level of service and a high degree of self-sufficiency, assuring at the same time a simple and fast support to communication. In this area it is almost essential to use wireless sensors and give therefore attention to the implications related to this kind of technologies.

The research area dealing with wireless sensor networks has been increasing in the last years and has been widely analyzed to prepare this thesis.

The reference studies suggest innovative solutions able to optimize consumption and make communication easier.

Recent developments of wireless sensor networks (WSN) permit to create wide sensor networks that, trough role protocols, succeed in sending information to the receiver limiting more than possible both energy consumption and communication overhead, so allowing to reach nodes that are in inaccessible positions. The faster and faster progress in sensor technologies has produced in the last years a continuous improvement of performances, obtained together with a strong reduction of devices dimensions and of the related energy cost. This has permitted to include more and more innovative functionalities like actuation capabilities present in new generation sensors. In this case, by actuation we mean ability to provide autonomous movements or, in the case of some SHM applications such as the lamb wave method, the capability to produce high energy waves to detect discontinuity in structures.

It is therefore easy to perceive how extremely important energy resources can become. The batteries alone fail to guarantee to the actuation systems a sufficient durability and it is therefore essential to search in other renewable sources the necessary energy to extend the life of this kind of devices. More and more WSN applications are exploring new energy harvesting technologies. These technologies exploit the chances to harvest from the environment the energy requested to feed the devices. Thanks to some studies, technologies have been developed to obtain energy from vibrations through piezoelectric devices, from natural heating or from wind power, but the source that guarantees the greatest power density is solar energy. In fact through solar energy it is possible to obtain a high quantity of energy also from solar panels of relatively limited dimensions and in this way, the energy needed to reach the targets or to indefinitely extend the devices life, can be supplied to devices and sensors.

Solar energy has permitted to power a high number of already implemented applications with very good results, but its utilization causes a series of problems, to be considered at the development level, due to its periodicity and availability.

This work, developed at the Computer Science Engineering Department of UCSD, was just born from the challenges that these issues have pointed out. The target of this work has been identified taking into consideration a platform previously developed thanks to a collaboration between Micrel Lab of the University of Bologna and the University of San Diego. SHiMmer [2] is a completely self-sufficient actuation platform for structural health monitoring. It is powered by solar energy and uses a supercapacitor to store energy. This platform is able to create a high power wave for the monitoring of structures through piezoelectric sensors, to elaborate the obtained data using algorithms also at high computational cost and to communicate the results to external agents (Unmanned Aerial Vehicle) through a wireless communication protocol. However, these tasks are accomplished only after a stimulation from the outside through radio triggering. Therefore the system is not completely self-sufficient but needs an external agent able to wake it up and request particular tasks.

The first question to which we tried to give an answer regarded the reason why an external agent was needed to perform some tasks. If the external agent does not arrive for long, a great quantity of energy is wasted since the supercapacitor offers an extremely limited capability. On the contrary, this energy can be exploited to collect data and to send the already processed result to the external agent. It is however necessary to consider that using all the input energy can bring to a total lack of energy that could be used, instead, in case of particular external higher priority requests or to communicate by radio already processed data. The high unpredictability of solar conditions causes a series of evaluations essential for the development of a reliable and complete system.

The developed work has produced different strategies to improve the SHiMmer platform and generalize the problems in order to extend a model that can include all the systems with similar constraints.

First a method to wake the system in a condition of complete charge has been searched, avoiding a periodical and frequent check of the energy storage unit.

A solar energy prediction algorithm, called Weather Conditioned Moving Average, has therefore been developed with the target to provide an input energy prediction, so to obtain a suitable sleeping time [1]. This algorithm periodically saves the power values obtained from the solar panel in a sliding window table where every row represents a past day. It then utilizes the past values to predict the future value. To obtain this, the seasonal variations of dawn and sunset times have been taken into account. At the same time, the variance between the past values of the present day and the correspondent values of the past days has been considered. In this way, an indication regarding the last values of the solar panel, compared to the previous days, is obtained and it is therefore possible to adapt the prediction to the actual weather conditions.

Once the input solar energy prediction has been obtained, it has been necessary to create a model to connect this energy level to an estimation of the recharging rate it can supply to the energy storage unit. Besides the study of the related literature, actual tests have been carried out to create an ad hoc development method and, therefore, equations that can relate these values to a correspondent recharging rate obtaining in the end the estimation of a recharge time. Given an energy prediction and a present charge value, the system will be able to calculate a sleeping time after which the energy storage unit level will have probably reached its maximum capacity or the requested energy level.

At this point, some strategies have been devised so to permit the system to keep a certain amount of energy to perform various priority tasks, but also to allow the system to "survive" during the night or in periods of poor solar conditions.

The 3 main SHiMmer tasks to extend the model to systems with similar characteristics have been identified. Systems that, firstly, have an active sensing or anyway an actuation task with a high energy cost, secondly a phase of data processing and, at last, a radio communication. Therefore strategies have been developed to schedule and coordinate active sensing activities, data processing and communication. These strategies give the possibility to change the priorities regulating these tasks to prevent energy and memory management. In fact, according to the availability of energy and memory, different choices can be made. The activity of data processing, for example, can give advantages since it permits, if necessary, to eliminate from the memory the whole acquired wave and to replace it with a resulting wave that occupies a more limited memory space. A similar evaluation can be made regarding communication. The processing can reduce the communication cost also in terms of time but, if it has a too high energy cost, it can be better to send the whole obtained wave and leave the task of processing data to the external agent.

Different energy profiles have then been defined. In particular, in the reference platform a high energy level profile and a limited consumption one have been defined. In the simulations carried out some choices have been made that involve the utilization of the high energy level profile as default, since it guarantees a higher precision and reliability of the result. However, when the solar conditions, related to the obtained prediction, show a future solar energy scarcity, the system is converted to a low energy level that permits to achieve a result using a minimum quantity of energy, in spite of a lower precision.

At last a theoretical analysis of the task queue has been carried out. This has brought to the development of a series of simulations that can show the advantages that all the algorithms and the strategies developed can give the SHiMmer platform and similar systems.

This thesis has been written keeping the logical sequence that has been followed while developing the whole project.

Chapter 1 concerns the most important technologies presently utilized as regards energy harvesting and the related techniques of energy management.

Chapter 2 describes the problem introduced by the study of Structural Health Monitoring, focusing on the SHiMmer platform used as a reference for the following works.

Chapter 3 introduces the problems faced that have brought to the division of the contributions into 3 different parts.

The algorithm of solar energy prediction (WCMA) is explained in Chapter 4 with different comparisons with algorithms with the same targets.

In Chapter 5 a study of the Recharge Estimator, used to calculate the recharging rate and therefore the sleeping time, has been introduced.

Chapter 6 describes the theoretical analysis of queue and the development of strategies regarding priority scheduling and division into energy profiles, with the related simulations.

The last Chapter explains the advantages obtainable using these techniques and the possible future development.

1. Power Management strategies in wireless systems with energy harvesting

1.1 Power management

The increasing of technology in the research and the decreasing of costs and dimensions in electronic components gives the opportunity to include a big amount of new features into the devices. One of the primary concerns in wireless systems that adopt newer technology features is energy consumption. Wireless systems development is a real challenge that depends on the different constraints associated with them. While the computational demand has drastically increased, the battery capacity is not following the same direction and can increase of a factor of 2 to 4 over a 30 years' period. To be able to fulfill the demand, the developer needs new strategies to permit the system to improve the computational performance using the current battery technology. Increasing techniques of low-power circuit design have helped in reaching a better battery lifetime, but also managing power dissipation can give an increase in the battery lifetime through a safe in the energy consumption.

A first solution proposed for dynamic power management is the system level energy conscious design. The aim is to selectively place idle components into low power states to reduce energy consumption. Many developers have introduced a new state based abstraction. Every state (for example active, idle, sleep) is a trade-off between power consumption and computation but the time for switching can be a relevant cost and can affect the performance. Developers must then define algorithms and policies to decide when to switch a component from a state to another. These policies must be able to maximize the performance respecting power constraints.

Timeout policy is one of the most common power management policies and is implemented in many operating systems. Unfortunately, even if this strategy is very easy to implement, the system wastes power while it is waiting for the timeout to expire. From that derives the need to develop predictive policies that are able to switch the device to a low power state, when it becomes idle, if the predictor estimates that the idle period will last enough time. Also this strategy cannot reach the optimality because its heuristic nature and a wrong estimation can affect energy consumption and performance.

Because of these problems the research focused on finding optimal solutions based on stochastic models. These models use distribution to describe inter-arrival times (the distance in time among the arrivals of different users' requests), the time it uses to serve the user and the switch time between different power states. The distribution can be general or static such as exponential or geometric.

We can classify power management strategies in different main branches depending on different behaviors: clock based or event driven depending on the switching policy, stationary or non stationary depending on the potential changing of policy over time. Since the stochastic model is based on Markov chains the optimality is granted both in discrete time (clock based) and in an event driven approach. Discrete time setting approaches waste energy because they require policy evaluation even when in low power state, while event driven models based on exponential distribution do not save enough power in real implementation because of the imprecision of exponential in describing the user inter-arrival times.

Two other approaches have been proposed in order to reach the optimality. The first model is based on the renewal theory and is used in the system with one decision time, while the second one is based on the time-indexed semi-Markov decision process model and is more general and complex. Both these approaches permit a solution solving a linear program in polynomial time and guarantee an optimal solution.

1.2 Wireless sensor networks

One of the main area where low power strategies are used nowadays is in fact wireless sensor networks. These networks combine sensing of the physical environment with a simple wireless communication and minimal computation facilities. This new approach to the concept of network can be deeply embedded in the physical environment. Because of the actual low cost technology and the wireless communication facilities, wireless sensor networks have a much larger number of applications than hardwired ad hoc networks and also a number of very different and specific challenges. The large number of conceivable combinations between sensing, computing and communication technology, cause the development of a high number of different sensor nodes. At the same time it usually happens that a sensor node designed for a specific scenario is not easily adaptable to a different network also because of the traffic characteristic in the environment. WSNs are likely to exhibit very low data rates over a large time scale, but can have very intense traffic when some particular event happens.

New types of applications have been developed thanks to WSNs: applications that include environmental control also installing sensors on buildings to understand earthquake vibration patterns, surveillance tasks of many kinds like intruder surveillance in premises. Because of the limited flexibility due to wires, deeply embedding sensors into machinery could presents higher costs and maintenance problems. The possibilities offered by WSN could potentially offer bigger changes when the basic size and cost problems are solved. Wireless sensor networks have, in fact, recently received a lot of attention in the research community.

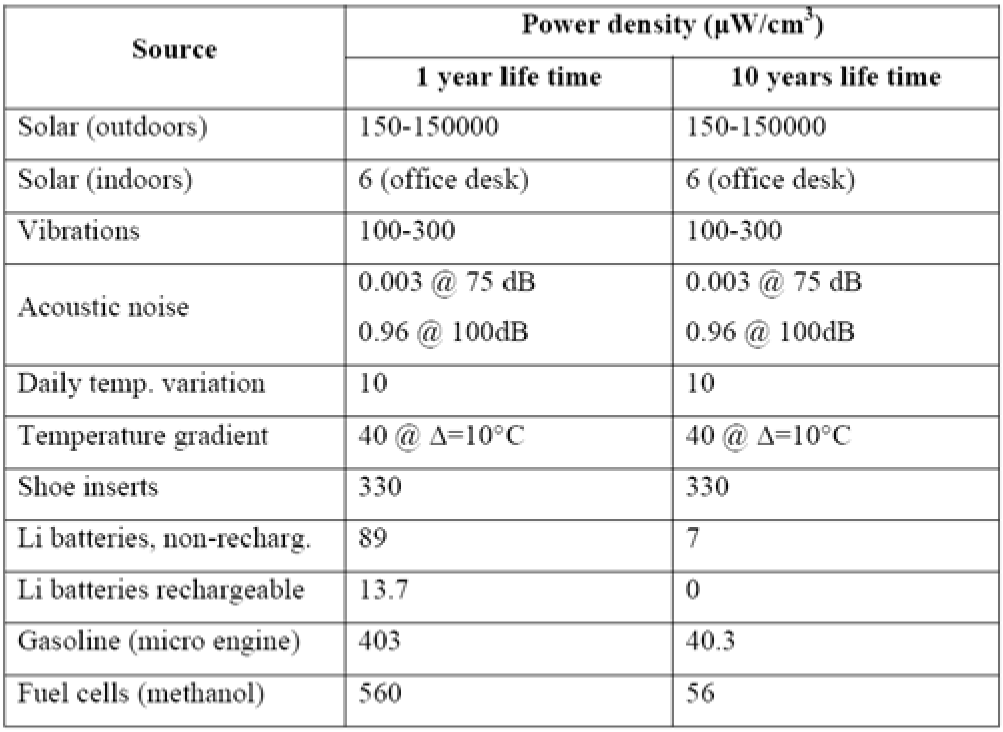
Considering a wireless sensor network (WSN) composed of several nodes, we must consider the possibility of a different strategy for every node. A sensor node must usually incorporate at least some sensors and a radio communication device but is also requires a computing unit. It must also include an energy supplier that can be a battery or a supercapacitor or an energy harvesting circuit that obtains energy from the environment. All these features have to be assembled in a very small circuit and it is easy to understand how important the power constraints are. A first solution to avoid the wasting of energy, considers the fact that every node performs its operations periodically. It is then possible to wake up the node when it needs to perform some work and switch the node in a low power state whenever it finishes. This strategy permits to save a big amount of energy that is able to extend the life of the sensor if the microcontroller has a good power mode management. The management of energy has a big impact in the node's costs and size because of both the circuit and the storage system. Because of the scarcity of energy the operational computation must remain some order of magnitude lower than in today's computers.

To face the typical WSN's constraints, some operating systems have been developed such as TinyOS and Free RTOS. TinyOS is one of the most used ones and is based on an event driven programming model instead of multithreading as many other operative systems. This operative system and the programs for it are written in nesC (network embedded system C). NesC is an extension of C programming language designed to embody the structuring concepts and execution model of TinyOS. TinyOS calls the appropriate event handler every time that an external event occurs, such as incoming data packets or sensor reading. The kernel has the function to schedule the handler. Free RTOS is a simple real time kernel open source. Free RTOS is mostly written in C language and few assembler functions and provides the main features to run simple applications in embedded systems and mainly in microcontrollers.

1.3 Energy harvesting strategies

In the previous sections the evolution of the wireless system has been presented in its relation with the progressive reduction of components in terms of size, costs and power consumption. When these wireless systems, which usually run on their own battery, are part of wider and more advanced networks, new problems are faced. Considering that a wireless node can be just one over hundreds in the network and can be placed in some less reachable locations the idea of installing and periodically replacing the battery in every node can be impractical. Batteries with average cost and size cannot respect the lifetime request of many applications but replacing the battery can be very time consuming, costly and in some cases impossible. For these reasons many researchers are developing new and different methods to collect ambient power from the environment in order to make wireless nodes self sufficient. There are different sources from where techniques for harvesting energy have been developed. There can be different ways to harvest energy from the environment. Easier methods are implemented to extract power from light, vibrations or air flow sources. The research is exploring all of these sources but for some of them the actual technology is still not sufficient to achieve important results. The main characteristic that distinguishes among power sources is represented by their power density.

As shown in Table 1.1, the solar source has the highest density among all the other sources and is hundreds of times bigger than the other alternatives. Considering this point of view, it would result useful to keep going in the investigation on the other sources but it must be considered that the solar power is strictly related to the location, the climate and the time of the day and, for these reasons, it can be very unreliable and not constant during the day. The choice of the source is in fact related to the application in which it will be used.



Source	Power density ($\mu W/cm^3$)	
	1 year life time	10 years life time
Solar (outdoors)	150-150000	150-150000
Solar (indoors)	6 (office desk)	6 (office desk)
Vibrations	100-300	100-300
Acoustic noise	0.003 @ 75 dB 0.96 @ 100dB	0.003 @ 75 dB 0.96 @ 100dB
Daily temp. variation	10	10
Temperature gradient	40 @ Δ=10°C	40 @ Δ=10°C
Shoe inserts	330	330
Li batteries, non-recharg.	89	7
Li batteries rechargeable	13.7	0
Gasoline (micro engine)	403	40.3
Fuel cells (methanol)	560	56

Table 1.1 Comparison of energy scavenging sources

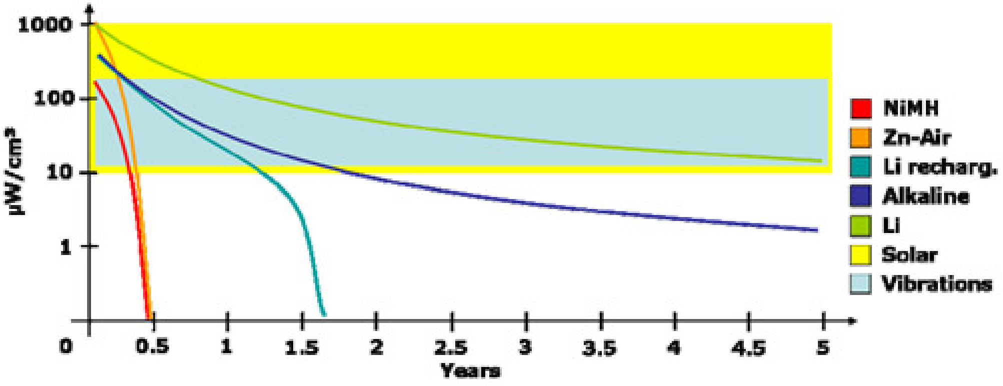


Figure 1.1 Continuous power/cm3 vs. life for different power sources

1.3.1 Solar energy

Solar energy is the best alternative when big sources of light are available such as outdoor light or relatively intense indoor light. In favorable sunny conditions the solar cells can harvest energy with a power density a hundred times the energy harvestable by vibration that is the second best choice.

The solar cells, that are used for harvesting solar energy, are currently a mature and reliable technology. Their technology is based on the photovoltaic effect acknowledged for the first time by A. Becquerel in 1839. The first real cell, built in 1883 coating the semiconductor selenium with a thin layer of gold from the junction, was just 1% efficient. The efficiency then increased to 6% with the following introduction of silicon doped with certain impurities in 1954.

The technology presently used comprises three different categories developed in three different generations. The first stage is a single layer p-n junction diode usually made with silicon wafer. In the second stage, from the use of a thin film as photovoltaic material, derives a great reduction of costs and a 15% increase of efficiency. To reach these results the technology uses different materials and semiconductors such as amorphous silicon, polycrystalline silicon, micro-crystalline silicon, cadmium telluride, copper indium selenide/sulfide. The third generation of solar cells, which are defined as semiconductor devices, includes photo-electrochemical cells, polymer solar cells, and nanocrystal solar cells. They do not reach the same performance as the second generation of cells but are useful for some specific applications where, for example, mechanical flexibility and disposability are important. The present research is focusing on new internal structures and materials. Multi-junction cells are obtaining very good efficiency results. These cells are obtained putting together multiple thin films produced using molecular beam epitaxy with different light-absorbing materials each film using the p-n junction principle. Each type of material has a characteristic band gap energy which permits to absorb electromagnetic radiation over a portion of the spectrum. The semiconductors are carefully chosen to absorb nearly all of the solar spectrum to generate as much electricity from the solar cells as possible. Using these technology, the efficiency grows to more than 40% with GaAs multi-junction devices but currently the costs are almost prohibitive (up to 40$/cm²).

From an electrical point of view solar cells can be considered as a current source in parallel with a diode. Using two shunt resistors to account for non idealities can improve the performance.

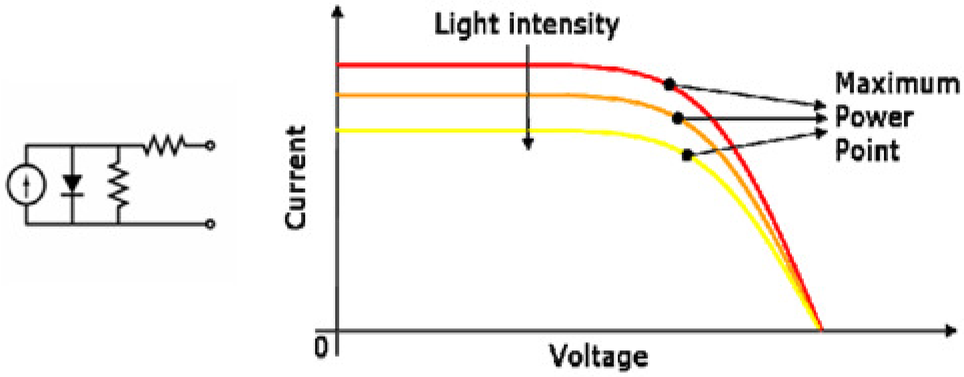


Figure 1.2 Equivalent electrical model and typical I/V characteristic of a solar cell

Depending on the resistive value of the load a solar cell can work with different voltage loads. From this derives that the issue is to make the solar cells work at their maximum operating point that maximizes the output voltage multiplied by the output current. This point is called maximum power point (MPP).

1.3.2 Energy from vibrations

A new source of energy has been explored for the applications that have to run in places not reached by the sun. Although the available power density is smaller, vibration to electricity conversion can be a good alternative. There are many techniques that can help in producing energy from vibrations. Displaceable inductors, variable capacitors and piezoelectric materials are some of the more effective solutions.

The basic principle using displaceable inductive elements is the magnetic induction using a coil that moves in a magnetic field. The movement obtained by vibration produces a current that flows through the coil's wire.

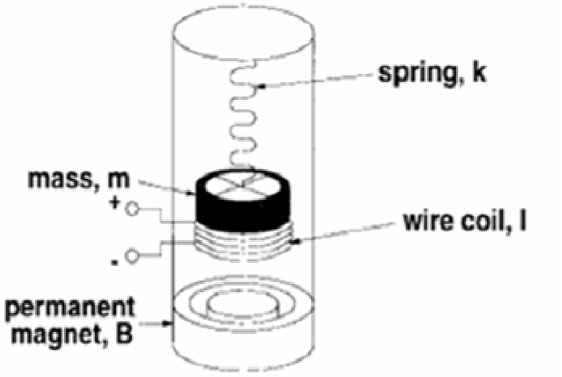


Figure 1.3 Scheme of principle of vibrations conversion based on moving inductors

Variable capacitors give a second opportunity to convert vibrations into electricity using the specific geometrical characteristic of capacitors modified by vibrations. These variations produce a change in the capacitance of the device that can be exploited to convert mechanical work into electrical work by inserting the capacitor in proper circuits. The circuit transfers energy from a source to the load as charge pumps do.

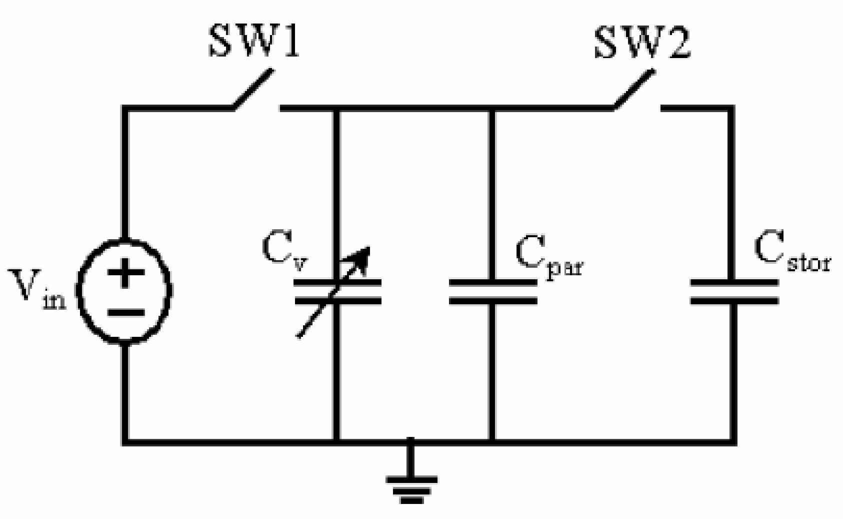


Figure 1

Figure 1.4 Charge pump with variable capacitor

The energy increases during the transfer because of the mechanical work done on the variable capacitor. The system is, in fact, able to produce energy given by the following equation:

$$E = \frac{1}{2} V_{in}^2 (C_V^{MAX} - C_V^{MIN}) \left(\frac{C_V^{MAX} + C_{par}}{C_V^{MIN} + C_{par}} \right) \tag{1.1}$$

where the variable capacitance is represented by the values C_V^{MAX} and C_V^{MIN}.

Currently the Micro Electro-Mechanical Systems technology (MEMS) gives the opportunity to have chips with small efforts embedding variable capacitors in many different configurations. One of these includes two arrays of metal plates, one fixed and the other one displaceable. The second one, while moving because of the vibrations, changes the overlapping area of the plates of the two arrays and global capacitance of the device. The second MEMS configuration considers two similar arrays and the non fixed one, while moving, creates a variation in the distance between the plates. The third configuration comprises a bigger metal plate suspended over another plate with variant distance due to the vibrations.

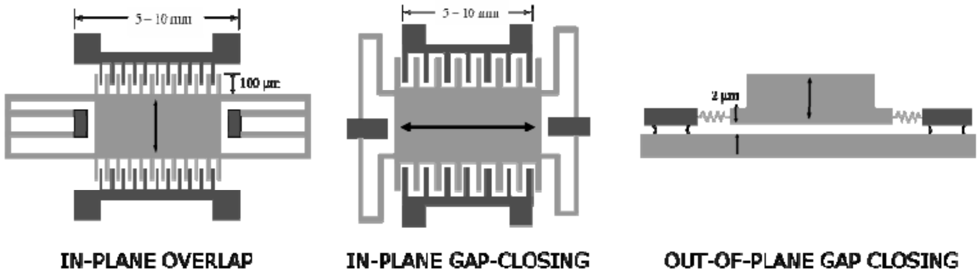


Figure1.5 MEMS configurations/variable capacitors

The maximum power density obtainable is around 100µW/cm^3.

The highest power density among vibration to electricity conversion systems is obtained using piezoelectric devices. Piezoelectric materials can be used for vibration-oriented energy scavenging solutions because they present good coupling between the mechanical and electrical domains. While the device is vibrating, it converts into electrical signals the mechanical stress and the power output is determined by the amplitude and the frequency of the vibration.

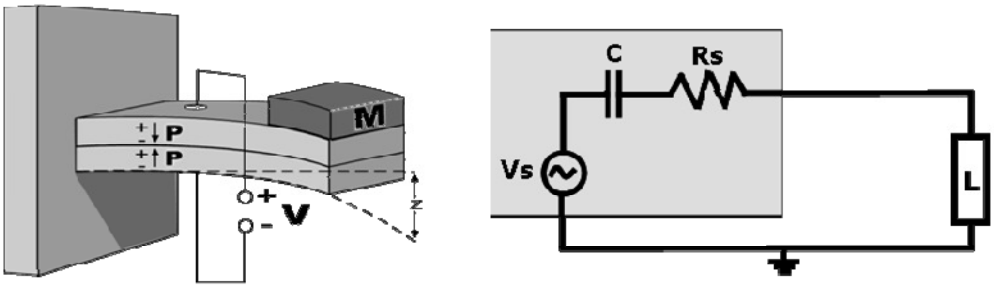


Figure 1.6 Vibrations to electric conversion by means of piezoelectric materials

Current research is focusing on embedding the piezoelectric generators in chips in the same way as variable capacitors realize it in MEMS technology.

1.4 Energy storage systems

1.4.1 Batteries

A battery is a device composed of two or more electrochemical cells that is able to store chemical energy and makes it available in an electrical form. The cells can be galvanic cells, fuel cells or flow cells. Modern batteries derives from the prototype created by the Italian physicist

Alessandro Volta in 1800. In theory a battery is composed of an array of similar voltaic cells interconnected but it is common to consider a single cell used on its own battery. Each cell is divided into two half cells that are connected in series by an electrolyte and a positive terminal and a negative terminal. The two terminals are both immersed in an electrolyte that can be solid or liquid but are not touching each other. In a practical cell the materials are enclosed in a container, and a separator between the electrodes prevents them from touching. This compliance permits the faradaic reaction where the electrolyte conducts current allowing the passage of ions between the electrodes. The electrical potential across the terminals of a battery is known as its terminal voltage.

There are two main classes that can define batteries. Primary batteries transform chemical energy into electrical energy in an irreversible way. In rechargeable batteries or secondary batteries the chemical reaction can be reversed by supplying electrical energy to the cell. Since we are focusing in harvesting energy, it is obvious that just the second class can interest our studies. Secondary batteries can be used, in fact, to store the energy harvested from the environment and make it available in different periods.

The various chemicals used in the cells correspond to different voltage values because of different electrochemical potentials that is why the investigation of materials to be employed is very important. The currently most used kind of cells include nickel-cadmium (NiCd), nickel metal hydride (NiMH) and lithium-ion (Li-Ion) cells.

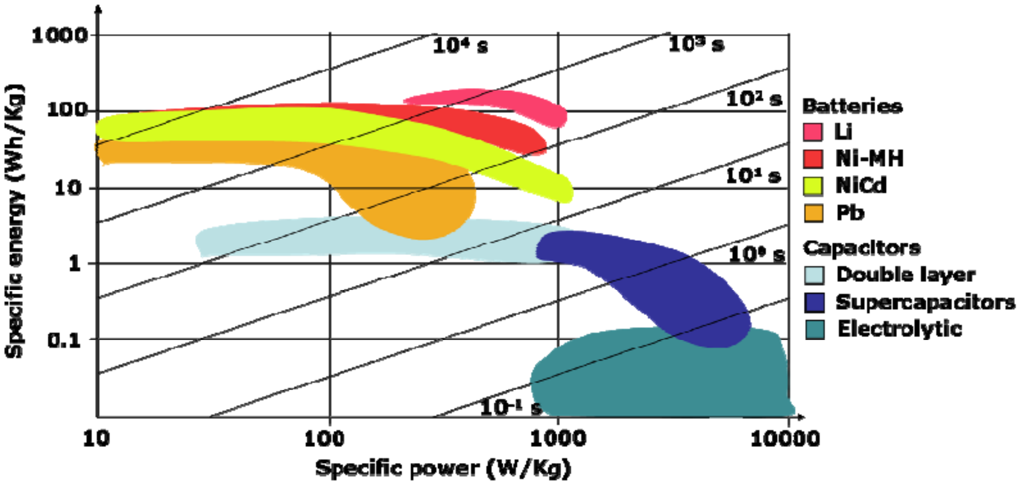


Figure 1.7 Comparison among technologies for energy storage

Independently from the main material of the battery, the more electrolyte and electrode material in the cell, the greater the capacity of the cell. The capacity of cells also depends on the discharge conditions such as the magnitude and the duration of current, the temperature, and other factors. Another important issue is the self discharge phenomenon due to non-current-producing side chemical reaction within the cell. Even when no load is applied to the battery, in fact, the energy decreases but the rate of this reduction depends on different factors among which the temperature in which the battery is working. If the battery is stored at a low temperature the rate of the side reaction is reduced but at the same time the battery can be damaged by freezing. Disposable batteries can lose from 2% to 25% of the original charge per year if they are working at inappropriate temperatures. Rechargeable batteries can also have a bigger reduction of the performances, up to 3% a day and, besides, the batteries present long term deterioration due to the aging of materials.

In addition to these problems, the capacity of batteries vary depending on the depth of discharge. This phenomenon is called "memory effect" and causes big problems to some kinds of batteries, among which NiCd, while it creates minor effects in NiMH and Li-ion types.

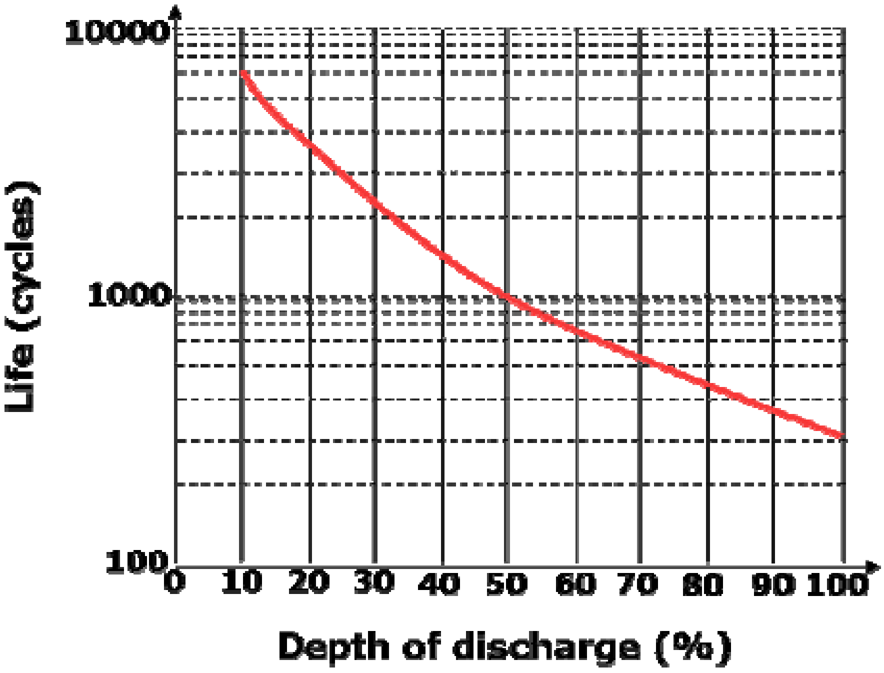
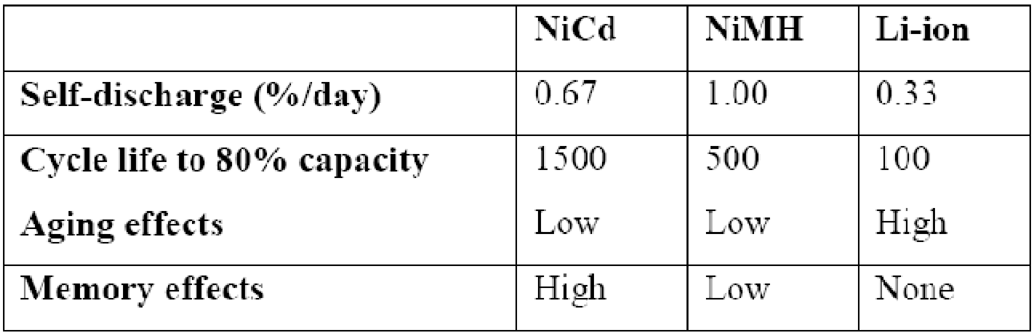


Figure 1.8 Life of a battery vs. depth of discharge



	NiCd	NiMH	Li-ion
Self-discharge (%/day)	0.67	1.00	0.33
Cycle life to 80% capacity	1500	500	100
Aging effects	Low	Low	High
Memory effects	High	Low	None

Table 1.2 Characteristics of most common rechargeable batteries

1.4.2 Supercapacitors

Supercapacitors are a family of capacitors with recent history who have outstanding performances considering the quantity of charge they can store compared to the typical capacitors. Because of this reason they can be a good alternative to batteries.

Supercapacitors derive from the electrolytic capacitors developed by Cornell Dubilier Corporation when, in 1930, the scientist introduced a new approach in designing capacitors introducing three major improvements. The etching of one Al electrode by acid increases the surface area available to charge, the following oxidation of the electrode creates an insulating layer (Al_2O_3) that is used as a positive and negative charge separator and eventually the immersion of the electrode in an electrolyte created by the reaction of the boric acid and the glycol, provides an increase in energy density because of the reduction of the distance between the layers and the increasing of the surface.

Supercapacitors have further improved electrolytic capacitors with new innovations developed by Standard Oil of Ohio Research Center (SOHIO). The charge separation distance has been reduced to few nanometers that is the dimension of the ions themselves within the electrolyte. That permits the ratio of available surface area to charge separation to be extremely big (1012) and consequently also the storage capabilities are high. In the 1960s in the SOHIO laboratories they discovered that two pieces of activated carbon immersed in aqueous electrolyte solution act as capacitors if connected across the terminal of a battery. When, in the following research, the use of organic electrolytes has been introduced, there was no market for that kind of devices until, in the 1971, SOHIO licensed the double layer capacitor technology to NEC.

In the 1980s Matsushita Electric Company patented a method of manufacturing supercapacitors with improved electrodes and in the next years many different applications started to be developed. At the moment automated assembly techniques permit a rapid decreasing of costs.

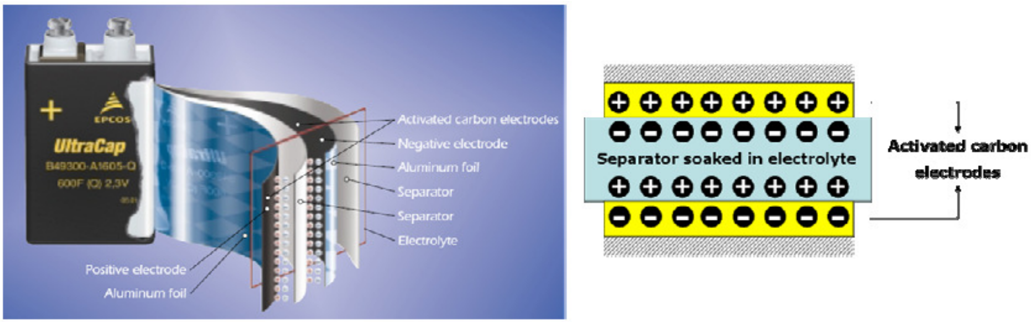


Figure 1.9 Internal structure of a supercapacitor

Charge and discharge of the supercapacitors performed upon movement of ions within the electrolyte, follow a reversible electrostatic effect. This characteristic strongly divides supercapacitor technology from battery one that is based on the faradic reaction. This results in a bigger lifetime in terms of number of cycles and life expectancy and substantially decreases the need of maintenance in the devices that adopt these techniques.

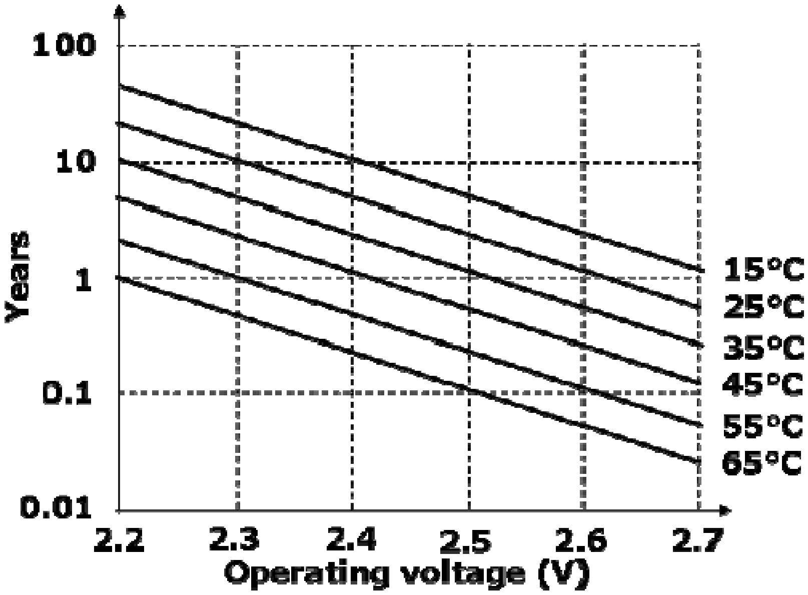


Figure 1.10 Life expectancy for supercapacitors

The capacity of current supercapacitors is up to 270F while the normal capacitor offers capacitances of picofarads. The main manufacturers of supercapacitors, such as Maxwell Technologies, NESS, Okamura and EPCOS provide carbon-carbon, or symmetric, supercapacitors where both electrodes have an identical construction. In 2003 a new class of devices has been announced. These devices, called nanogate or nano-carbon capacitors, present an energy density 10 times higher than previous supercapacitors and exploits a patented material with high porosity and accessibility for storing ions. Researcher are exploring also the possibility of using carbon nanotubes for supercapacitor electrodes because of their uniform nanoscopic pores of about 0,8 nanometers in diameter which could store more charge than nanogate capacitors if the nanotubes could be assembled into macro scale units in a proper way.

The use of supercapacitors have been widely increased with the introduction of a big amount of new applications in wireless nodes especially when energy harvesting capabilities are present. A supercapacitor can be used both as a unique storage device and connected with batteries using hybrid strategies. The main reason why supercapacitors are widely used is because of the high life expectancy supported by a low performance degradation over time, but also because they can deliver higher peak currents than batteries, without suffering any damage. The charges in supercapacitors, in fact, can be faster released than in batteries because of the chemical processes that they require and the speed limits of the current flow.

2. Structural Health Monitoring and SHiMmer platform

2.1 Structural Health Monitoring Methods

Structural Health Monitoring (SHM) is defined as the process of observing a structure over time, identifying a damage sensitive feature in the observations and performing a statistical analysis of these features to determine the health of the observed structure.

A major focus of the current research of structural engineering community is, in fact, to develop systems and structures that can monitor their own structural integrity in real time. Besides preventing catastrophic failures, online damage detection would reduce costs by minimizing maintenance and inspection cycles. One of the most promising means of developing these self monitoring structures is through the integration of smart materials into the structures themselves. Smart materials are materials that couple two forms of energy such as magnetic and mechanical energy for magnetorheological fluid, heat and mechanical energy for shape memory alloy, and electric and mechanical energy for piezoceramics. These materials can often serve as both sensors and actuators. The usefulness and effectiveness in SHM applications of Lead-Zirconate-Titanate (PZT) piezoelectric transducers have been investigated [50][67]. Commercial PZT devices are fairly low cost and small-sized so they can be integrated at high density in structures in an unobtrusive and inexpensive way.

The impedance-based technique and the Lamb waves technique are nondestructive evaluation (NDE) methods, which utilize the benefits of piezoelectric materials and show great promises for structural health monitoring systems. The basic concept of the impedance-based method is to use high frequency vibrations to monitor the local area of a structure for changes in structural impedance that would indicate damage or incipient damage. This is possible using piezoelectric sensor/actuators whose electrical impedance is directly related to the structure's mechanical impedance. The impedance measurements can easily give information on changing parameters, such as resonant frequencies, that will allow for the detection and location of damage. Lamb waves are a form of elastic perturbation which can propagate in a solid [55]. They are considered the preferred choice for detection of defects in plate-like structures [49]. Lamb waves method consists in a PZT actuator integrated into a structure which is vibrated by applying a proper electrical signal to it. Given the coupling between the electrical and mechanical domains which characterizes these kinds of devices, the shape, amplitude and frequency of the input wave determine the features of the actuator vibration. Then the sensing phase begins, during which the response of the structure to the vibration is sampled by means of a second PZT device which serves as a sensor. In a way dual to the actuating process, the sensor produces an electrical signal with properties correlated to the vibration acting on it [61]. The response of the structure depends on its state of integrity. Presence of cracks, delaminations or other defects will alter the shape of the wave produced by the PZT sensor with respect to a damage-free condition. Thus, it is possible to monitor the structure health condition by analyzing the differences between the two waves [38][54]. Many kinds of analyses can be performed, both in the time and frequency domains. A thorough review can be found in [61] and [74].

2.2 SHiMmer Platform

The main contribution of our research is strictly dependent on the studies previously developed about SHiMmer platform [2]. SHiMmer is a wireless platform for actuation and sensing that uses localized processing with energy harvesting to provide long-lived structural health monitoring. One of the most important issues that provides long life to the system is the introduction of supercapacitors instead of batteries which permit the platform to work completely maintenance-free. The node is capable of harvesting up to 780J per day. This makes it completely self-sufficient while employed in real structural health monitoring applications. Unlike other sensor networks that periodically monitor a structure and route information to a base station, the device acquires the data and processes them locally after being radio-triggered by an external agent. The localized processing allows us to avoid issues due to network congestion.

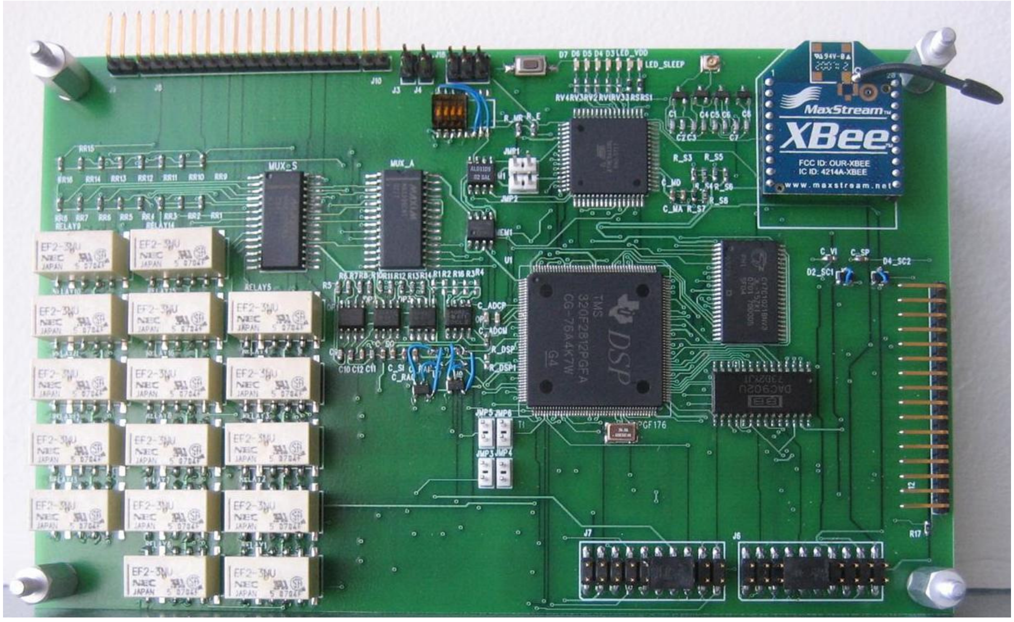


Figure 2.1 SHiMmer platform

The development of SHiMmer derives from a joint project with the Los Alamos National Laboratory (LANL) and aims at developing a wireless sensor network to be deployed over civil infrastructures for SHM purposes such as the monitoring of bridges, industrial plants or airplanes. The network comprises 2 layers of sensor nodes. The first one is composed of RFID sensing devices coupled with PZTs. It uses a signal provided by an unmanned aerial vehicle (UAV) to detect peak strains. SHiMmer nodes are activated if the UAV finds unsafe values of peak strains. Every node can be connected up to 16 PZTs. A PZT can actuate a lamb wave within the structure to let another PZT sense it. The node is able to detect and localize the damage and either store the results in memory or immediately transmit them back to the UAV.

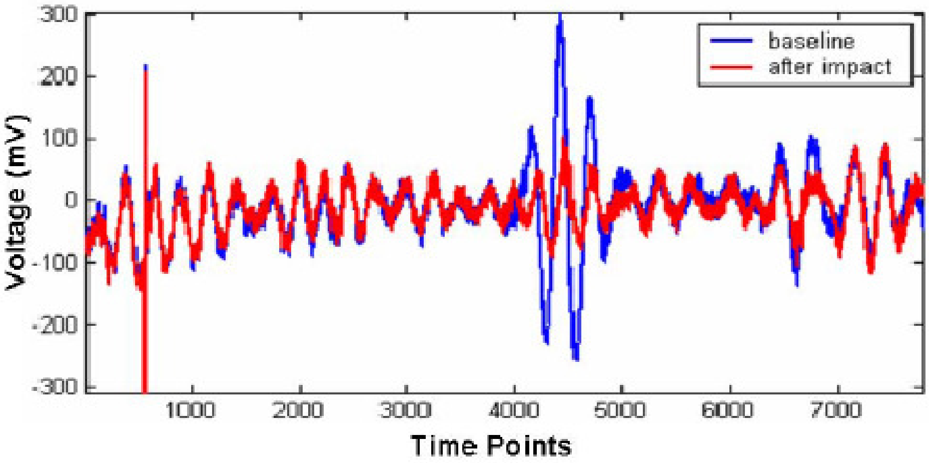


Figure 2.2 Lamb wave method

Both if the structure needs frequent on line evaluations to analyze the deterioration, or if it needs to be controlled just after special events such as earthquake, the system must be able to wake up, acquire and elaborate the data and transmit the results in case of external request. To attain this target, the node must be structured in such a way that the power consumption is near to zero during its inactivity periods. The period of inactivity then correspond to most of the day. For this reason the choice of the components is accurate and the node architecture presents different power aware solutions. The damage detection approach used in the system derives by the Lamb Waves method that offers the possibility to detect and localize a damage in the structure using a set of piezoelectric devices organized in grid over a portion of the structure. The organization of the grid permits to sense the damage by using multiple paths among the devices.

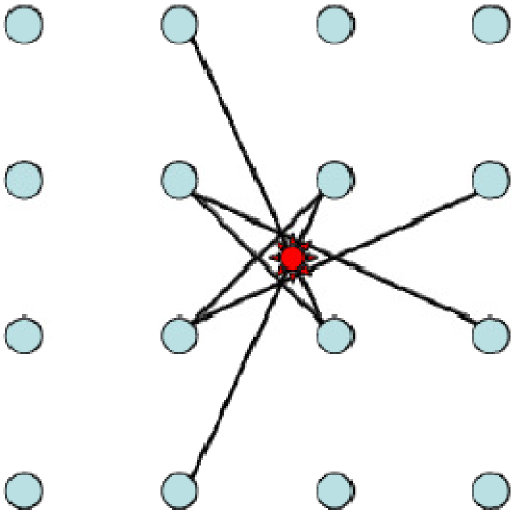


Figure 2.3 Multiple path among PZTs

The network can be deployed over a variety of structures, made of different materials.

The choice of the frequency and amplitude of the actuation wave is a main issue in the analysis and depends on the structure material. A multiplexer coupled with a demultiplexer allow the selection of a specific pair of PZTs as the actuator and the sensor of the acquisition operation. The node is capable to control a device with waves up to 1MHz frequency, with a 15V peak-to-peak maximum amplitude. The A/D converter is set at a sampling rate of 10MHz because of the accuracy needs of the sensed wave in the time domain. The expected response of the structure is at the same frequency. Because of the Nyquist theorem it requires at least 2MHz of sampling rate to be used in the application, so the waveform should be reconstructed using the following formula:

$$x(t) = \sum_{k=1}^{n} x(kT_S) * sinc(f, t) \qquad (2.1)$$

The equations (2.1) represent the way to obtain the waveform where T_S is the sampling period and $x(kT_S)$are the samples of the signal. The sampling frequency has been set at 10MSPS, so that it is possible to acquire 10 samples per period in the case of actuating wave at the maximum frequency of 1MHz. This allows to achieve the required accuracy in the time domain without further processing of the samples by the Digital Signal Processor. The computational characteristic and the configurability of the DSP enables the node to run most analysis algorithms typically needed [70].

2.2.2 Hardware architecture

The main tasks that the node have to accomplish are the actuation process with the control of the PZT devices, the sampling of the response, the processing of the required data and the communication with the UAV. These tasks involve activities both in the analog and digital domains, with different requirements in terms of power, peak current and supply voltages.

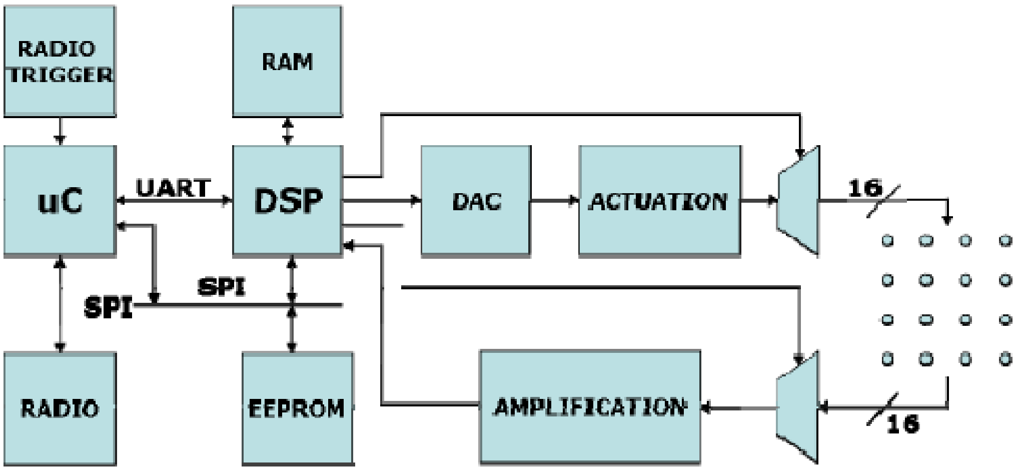


Figure 2.4 Block diagram of the sensor node

To coordinate all these activities, the node has been provided with a low power microcontroller which consumes less than 1mA in active mode and only 5µA in sleep mode. The microcontroller is connected with a radio transceiver working in the 433MHz band, a 32-bits DSP and an EEPROM with the target of storing the code for the DSP. The microcontroller is also used to control a network of CMOS switches used to selectively disconnect the different parts of the node from the power source. For the event triggering by UAV a passive radio-triggering circuit is used to generate an interrupt to wake it up when a proper signal is received. The microcontroller is then the only component drawing current from the power source in sleep mode and this permits a very low power consumption in the quiescent state. After waking up, the microcontroller switches on the other parts of the node and relinquishes control to the DSP.

The DSP is interfaced to 1Mb of SRAM, a DAC and two signal conditioning stages dedicated respectively to the output wave generation and to the input signal filtering and pre-amplification.

The external SRAM is used to store the samples generated by the A/D converter integrated in the DSP. The sensing phase lasts for about 1ms, therefore a data series is composed of 10K samples, given the desired sampling frequency (10MSPS). The resolution of the A/D converter integrated in the DSP is 12 bits. As a result, a data series occupies 20KB. This has imposed the size of the external RAM, which has to be capable enough to store a data series and provide the space needed during the data processing. An FFT algorithm on N samples requires 2N memory locations. With respect to this, a 64KB SRAM would have been sufficient but the platform is provided with a 128KB memory to allow the node to perform more complex algorithms. In addition, the short access time of the chosen memory chip allows the data to be transferred easily in a sampling period.

The DAC has a 12-bit resolution and works at a 20MSPS update frequency. The input port width allows a sample to be received in a single transfer. The high update frequency is needed to generate the output wave with an adequate accuracy. The DAC complementary current outputs are amplified and converted to a voltage by the actuation conditioning stage. This stage is the part of the node associated to the highest requirement in terms of peak current.

A PZT device is a capacitive load and the actuators used show an input capacitance in the 5-10nF range. The maximum current to be provided while driving a capacitance with a sine wave is given by the following equation:

$$I_{MAX} = \pi C f V_{PP} \tag{2.2}$$

where f is the frequency of the signal applied to the capacitance and VPP is the peak-to-peak wave amplitude. To control the PZT devices and achieve the desired actuation performances, a peak-to-peak wave amplitude in the range 10-20V is required. A high current output amplifier permits to generate waves of up to 15V peak-to-peak amplitudes, corresponding to a peak current of 470mA.

The DSP also have the aim to control a multiplexer and a de-multiplexer, which are used to select a specific couple of external piezoelectric devices. The actuation multiplexer is not directly interfaced to the PZT devices, but is interfaced to relays inserted between the output of the amplification stage and the PZTs because the high current peak associated with the actuation process exceeds the maximum ratings of commercial multiplexers. The extremely low resistance of the relays (< 0.1) minimizes the power dissipation and avoids a reduction of the output voltage swing.

2.2.3 Energy Harvesting in SHiMmer

To collect energy by means of solar cells SHiMmer is equipped with an energy harvesting circuit that is able to store in supercapacitors the energy collected. No batteries are included in the node because of their lower durability and faster performance degradation [66]. The solar cells integrated in the node sum up to an area of 100cm^2 and can produce up to 360mW in sunny conditions, and more than an order of magnitude smaller in cloudy conditions. The supercapacitors are commercial devices with a working voltage of 2.5V and a low ESR, which result in a low leakage current. The node is provided with a total capacity of 250F and from this derives that the maximum energy which can be stored is 780J.

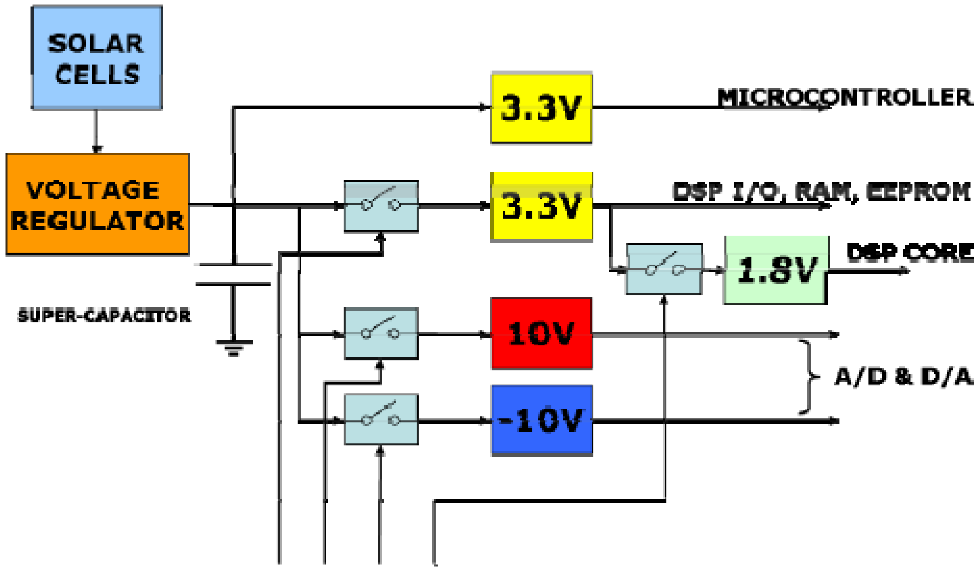


Figure 2.5 Energy circuit of SHiMmer

A 2.5V voltage regulator connected to the solar cells charges the super-capacitors. The component can be associated with a maximum power point tracking circuit to maximize the efficiency of the charge process [63][64]. The supply voltage for the microcontroller is produced by a 3.3V boost converter directly connected to the supercapacitor. Its minimum input voltage is 0.8V. As a consequence, the exploitable amount of energy is reduced to 700J.

In addition to the harvesting circuit, the node includes the components needed to produce the suitable supply voltage for its different parts. The DSP requires 1.8V for its digital core and 3.3V for the analog interfaces, while the signal conditioning stages require a dual supply (10V and -10V) to generate the actuation wave and pre-amplify the signal produced by the sensing PZT.

A dedicated boost converter powers up the EEPROM, the RAM, the I/O interfaces of the DSP and the input and output multiplexers. The 1.8 volts required by the digital core of the DSP are produced by a voltage regulator cascaded to this boost converter. Finally, a boost/inverting converter outputs the 10V and -10V supply voltages. The microcontroller can completely disconnect all the converters from the supercapacitors when entering the sleep state by controlling CMOS switches inserted between the supercapacitors and the converters input pins. This solution allows the node to achieve a power consumption during the sleep phase as low as 100uA. As a result, the node consumes 26J per day while inactive and over 96% of the stored energy can be used for acquisition and computation. SHiMmer offers an outstanding performance in sleep mode, the energy section delivers a high peak current and continuous power during the operational phases. Given the amount of energy which can be harvested in one day and stored in the supercapacitor, the DSP could work at its maximum speed for 15 minutes. This time interval is enough to execute the damage detection analysis.

2.2.4 Radio Triggering Circuit

The radio-triggering circuit is a passive circuit connected to a planar antenna working in the 2.4GHz band and to an external interrupt pin of the microcontroller. The circuit has low barrier schottky diodes and capacitors which form a charge pump [48].

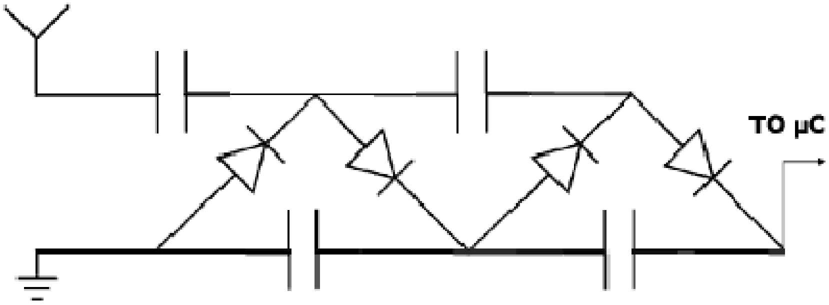


Figure 2.6 Radio triggering circuit

The network of diodes and capacitors rectifies the signal received by the antenna. The microcontroller requires 2.0V to trigger an interrupt. The output voltage depends on the complexity of the circuit such as the number of diodes and capacitors used. Depending on operational conditions, the circuit can be tuned to obtain the required output level. The number of components can be kept low using chips integrating two diodes in the same package. The output of the triggering circuit can be used to turn on the gate of a MOSFET acting as a pull-up for the interrupt pin of the microcontroller. Since MOSFETs can have a threshold voltage as low as 0.2V, this further reduces the minimum voltage the circuit needs to output to wake-up the node. As a consequence, the power of the radio triggering signal can be reduced as well.

2.2.5 Software Architecture

The two main components of the code are the microcontroller code and the DSP code. Operating system primitives such as interrupt service routines, radio communication, and power management are handled by the microcontroller, while the DSP handles the high level aspects of structural health monitoring, such as wave actuation, response sampling, and analysis.

The core of the **microcontroller** code can be thought of as a state machine where each state represents what operations should be performed and the transitions represent task completions or interrupt firing. The basic use case of the system is represented by the following flow.

An external agent (UAV) sends the sensor a signal, which wakes up the microcontroller. As a result, the microcontroller powers on and configures the radio transceiver, and waits for a command. If a false trigger occurred, the correct command does not arrive within the time slot allotted and the microcontroller returns to deep sleep. When a correct command is received, the microcontroller starts up the detection sequence. It first powers up the rest of the system, namely the DSP, its external EEPROM and SRAM, and the signal conditioning stages. Once this is complete, the microcontroller enables the DSP which begins actuating, sampling, and processing the necessary data. The details of the DSP operation are described in the following section. When the data has been processed, the result, indicating if there is damage or not, is transmitted to the microcontroller. The microcontroller shuts down the power supplies of all the other parts of the node except for the transceiver. At the end, the final result is transmitted back to the external agent and the radio transceiver is powered down too.

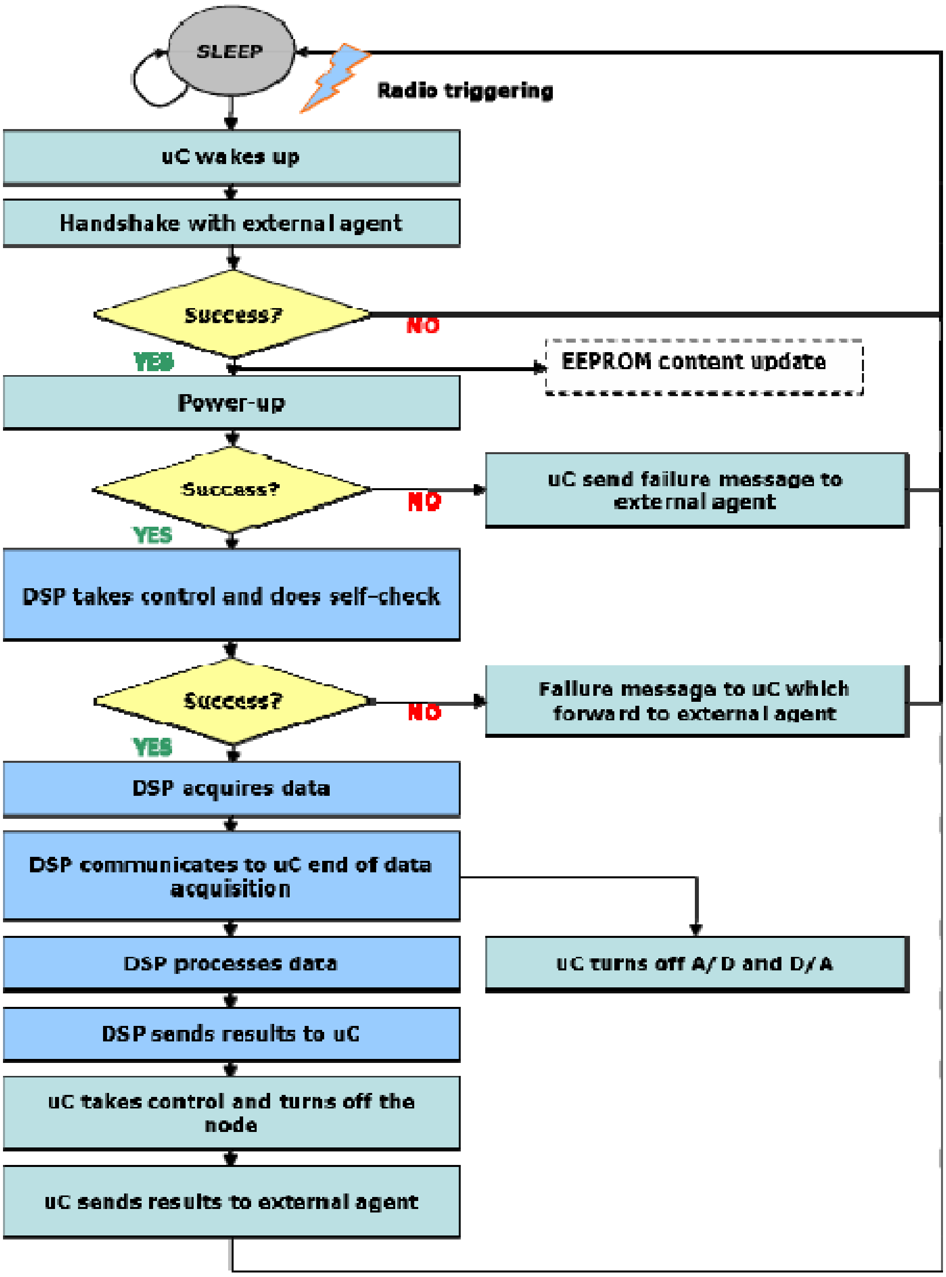
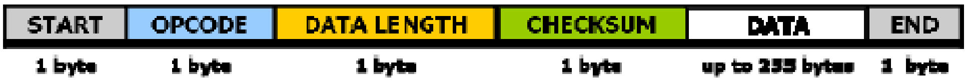

SLEEP
Radio triggering
uC wakes up
Handshake with external agent
Success?
NO
YES
EEPROM content update
Power-up
Success?
NO
YES
uC send failure message to external agent
DSP takes control and does self-check
Success?
NO
YES
Failure message to uC which forward to external agent
DSP acquires data
DSP communicates to uC end of data acquisition
uC turns off A/D and D/A
DSP processes data
DSP sends results to uC
uC takes control and turns off the node
uC sends results to external agent

Figure 2.7 SHiMmer activity diagram

The microcontroller communicates with the external agent following a specific three steps protocol. The external agent sends first a command to the sensor node. Then the sensor node sends an acknowledge message, after receiving the command. Finally, the sensor node sends the external agent the results of the task associated with the command. The packet format is lightweight, with only an opcode byte, payload length byte, payload, and cyclic redundancy check (CRC) byte.



START	OPCODE	DATA LENGTH	CHECKSUM	DATA	END
1 byte	1 byte	1 byte	1 byte	up to 255 bytes	1 byte

Figure 2.8 Packet format

Two fixed-value bytes are used as packet delimiters. This packet format provides a high degree of flexibility to the communication protocol. This makes it easy to extend the set of commands for both operational and debugging purposes while providing a basic error detection feature with the CRC byte. The length byte enables different types of commands to be executed, ranging from self test and actuate, which require a small number of bytes, to software reconfigure, which may require substantially more. As a convention, we use opcodes starting from lower numbers to signify communication from the UAV to the sensor and higher numbers for the opposite direction. The CRC byte is simply an exclusive-or of the opcode and length. There is no use of complicated error correction schemes to minimize the computational load associated with the communication. In the event of a dropped message, the operator of the UAV simply retransmits the command. Although it is possible to implement a transport layer for service, the development of a network stack would introduce an unnecessary overhead and increase the power consumption.

The **DSP** needs to provide computation for wave generation, sampling, storing and processing. The interaction between the different phases is extremely complicated when sampling at such a high frequency. For example, the samples collected from the ADC must be stored into the external RAM at least as fast as the sampling rate or data will be dropped.

The first DSP task is the wave generation which is done via a lookup table (LUT) of discrete points. The TMS320 DSP has a built-in LUT for sine wave generation in its internal ROM. Custom waveforms can be generated by loading a specific LUT in the external EEPROM. The generation of the wave must be done at 1MHz and the sampling frequency is set equal to 10MSPS. The sampling and generation is done sequentially instead of simultaneously to minimize the impact of sample jitter caused by interrupt processing. The wave has a finite propagation speed of in the structure and the time of flight of the wave from the actuator to the sensor enables the node to finish the actuation process before starting the sensing phase. Because of the requirements of the SHM, the platform needs to acquire 10,000 samples at 10 MHz, that means to sample for 1 ms. Because of this, 20KB have to be sent through the D/A converter at the proper rate. Achieving this rate requires special code routines to maximize the efficiency of the DSP saving several key cycles by converting a conditional loop to an unconditional one that would break out via timer interrupt. This increases the performance and the D/A rate by an order of magnitude. Once all the data are sampled and stored, the DSP performs the data application-level processing. It is important to notice that several processing algorithms are used for SHM. Different algorithms are more effective in different scenarios with different materials [47]. Thus there is no clear SHM algorithm that outperforms any other. A representative SHM algorithm used at LANL uses a fast Fourier transform, a Hilbert transform, as well as magnitude and phase calculations of complex numbers. Conceptually, the algorithm works by comparing a baseline signal with a potential damage signal. An FFT is initially used to denoise the two signals. Then the amplitude of each wave is normalized before a Hilbert transform is used to find the envelope of each normalized signal. Finally, the difference of the two envelopes is computed to determine the damage. Once the results are finally computed, the information is simply passed back to the microcontroller through a serial interface.

2.2.6 Evaluation

The main functional blocks which constitute SHiMmer are microcontroller, radio circuitry, DSP with actuation and sensing circuits and energy harvesting unit. A typical detection sequence starts with a radio trigger which wakes up the microcontroller. After being triggered, the microcontroller checks the level of charge of the supercapacitors and turns on the supply voltages for the rest of the node. Then, some packets of data are sent by the UAV to set the parameters of the analysis. Finally the DSP performs the actuation, sensing and processing.

The actuation circuit amplifies the signal produced by the DAC converter and is directly interfaced to the actuating PZT. The sensing circuit amplifies and shifts the output of the second PZT so that it matches the voltage input range of the ADC on the DSP. The most critical task in terms of power consumption and peak currents is the actuation. During this phase, the energy circuit has to deliver up to 900mA at the highest actuation frequencies. The actuation circuit has proven to be effective through the whole required range of frequencies. The amplitude of the actuation wave decreases by less than 10% at the top frequency with respect to the lower-bound frequencies. The duration of the actuation phase depends on the application, but it is still expected to last for a short time interval. For example, generating 10 periods of a 100KHz sine wave would require only 100us. This makes the energy consumption associated with the task small. The shortness of the actuation phase eliminates the need for a heatsink. The acquisition phase lasts for a fixed amount of time, determined by the sampling frequency and the number of samples needed. In case of a 10MSPS frequency and a 10k samples series, the duration is 1ms. The consumption of this stage is 8mA. The result is quite satisfactory considering that the gain of this stage is 15, and it includes a fourth order anti-aliasing filter.

The DSP is able to implement a critical set of the algorithm such as an FFT, a normalization, a Hilbert transform (represented as an FFT and an inverse FFT) for both the sampled data points and the original wavelet data points. These operations are at the base of most SHM algorithms together with the computation of the difference between the sample and the original data points.

The energy circuit is implemented with a 100F supercapacitor charged by a 100cm^2 solar panel. The time to fully charge the supercapacitors increases depending on the solar conditions. The circuit draws a quiescent current equal to 80uA. This corresponds to an energy consumption of about 18J per day.

3. Contributions

3.1 Contributions in SHiMmer platform

The main problems that we faced in the development of this thesis derive from the accurate study of the SHiMmer platform explained in the previous chapter.

3.1.1 Software

First of all, our target was to evolve the platform algorithms to improve the flexibility of the system and extend the core code so to adapt it to possible further capabilities of the system.

The core of the microcontroller code behaves as a state machine where each state represents what operations should be performed and the transitions represent task completions or interrupt firing.

The basic use case of the system is shown in Figure 2.7. We therefore want to go over the model of the state machine to restructure the code in methods able to respond to the requests from the UAV, but also to the decision of the algorithms that will be implemented in the microcontroller itself. For this reason the microcontroller code has been totally rewritten permitting to communicate with it using the same interface. The DSP and the GUI can communicate in fact with the microcontroller using the same protocol described in Chapter 2, but more features have been added such as the request to the DSP of performing a Fast Forward Transform and the possibility of receiving the result in a similar way as the original sensed wave is received.

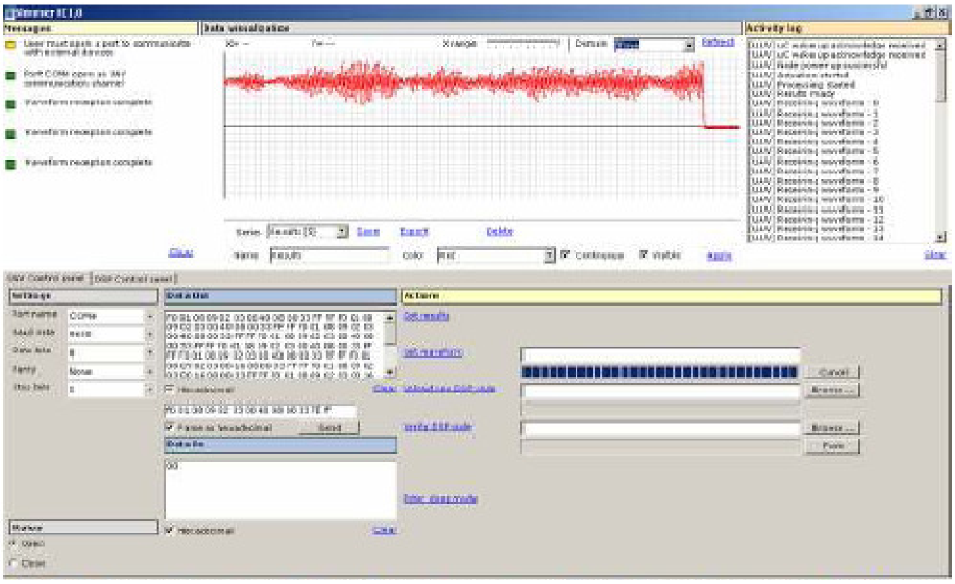
A new timer has also been added to the code. The timer is used to give the possibility to implement the solar energy predictor that will be explained in Chapter 3. Because of the constraints, the timer must be able to wake up both with a fixed period *T* and after the time selected by the predictor together with the recharge estimator. Because of this, even if the selected time is high, the maximum sleeping time of the system will be the fixed period *T*. This permits to use a timer with a maximum period of about 1 hour but with a high level of granularity.

Another issue in the evolution of the code in the microcontroller is to give the possibility to extend the memory usage. While in the past just the external 64KB SRAM was used to contain the resulted wave of the active sensing part, the memory usage has been reorganized to obtain the usage of the internal 128KB flash memory in order to contain the response of the DSP elaboration. The table of the value used in the solar energy predictor can be then contained in the external memory.

Also the DSP code has been partially rewritten. To perform the calculation of the Fast Fourier Transform on the sensed waves we used and modified a library for a complex FFT developed by Advanced Embedded Control of Texas Instruments.

The code is modified so to be able to obtain an FFT of 128 point from an original wave composed of 8192 points. In SHM application this provides an easier way to obtain the detection of the damage in the structure and, at the same time, a reduction of the memory usage because the original wave can be deleted to free space for a new actuation and sensing phase.

The Graphical User Interface has been firstly adapted to the new FFT wave generated by the DSP. We introduced in the GUI the capability to receive both the entire wave of 8192 values and the 128 values FFT. The various waves can be transmitted from the DSP to the user interface, through the microcontroller, using packets with a payload of 128 short integer values (256 bytes). The Visual Basic code is able to collect all the packets that form the wave and display it in a graph.



User must open a port to communicate with external devices

Port COM4 open as 1st communication channel

Waveform reception complete

Waveform reception complete

Waveform reception complete

```
UAV  uC wake up acknowledge received
UAV  uC wake up acknowledge received
UAV  Node power up successful
UAV  Actuation started
UAV  Processing started
UAV  Results ready
UAV  Receiving waveform - 0
UAV  Receiving waveform - 1
UAV  Receiving waveform - 2
UAV  Receiving waveform - 3
UAV  Receiving waveform - 4
UAV  Receiving waveform - 5
UAV  Receiving waveform - 6
UAV  Receiving waveform - 7
UAV  Receiving waveform - 8
UAV  Receiving waveform - 9
UAV  Receiving waveform - 10
UAV  Receiving waveform - 11
UAV  Receiving waveform - 12
UAV  Receiving waveform - 13
UAV  Receiving waveform - 14
```

Port name: COM4
Parity: None

Figure 3.1 Graphical User Interface displaying the wave sensed by the DSP

The GUI is also provided with new capabilities for the communication with the microcontroller. New methods to upload the binary code are written to permit any user to make the development and test of the microcontroller code easier. The entire code has been restructured to obtain an easier interaction with it and to allow future changes.

3.2 System Evolution

The main contribution derives from the evaluation of the activity diagram of the SHiMmer that could be simplified as in the following figure.

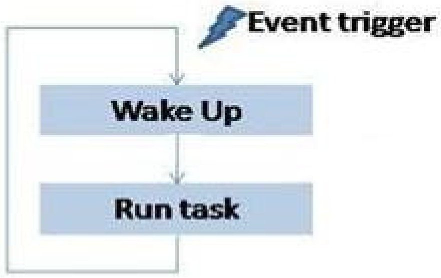


Figure 3.2 Initial activity diagram

In fact, the SHiMmer performs the required tasks after the event triggering from an external device that is the unmanned aerial vehicle (UAV).

We decided to evolve the system to firstly make it able not to be completely dependent on the event triggering, but to schedule is own tasks when there is enough energy available. If the system is able to wake up when the energy storage unit is full, it will be able to use that energy for performing active sensing and replace it with the new energy that will enter the system. This energy, on the contrary, would be wasted because of the full capacitance of the unit.

In Chapters 4, 5 and 6 the improvement that bring the activity diagram of the system to the following scheme will be explained.

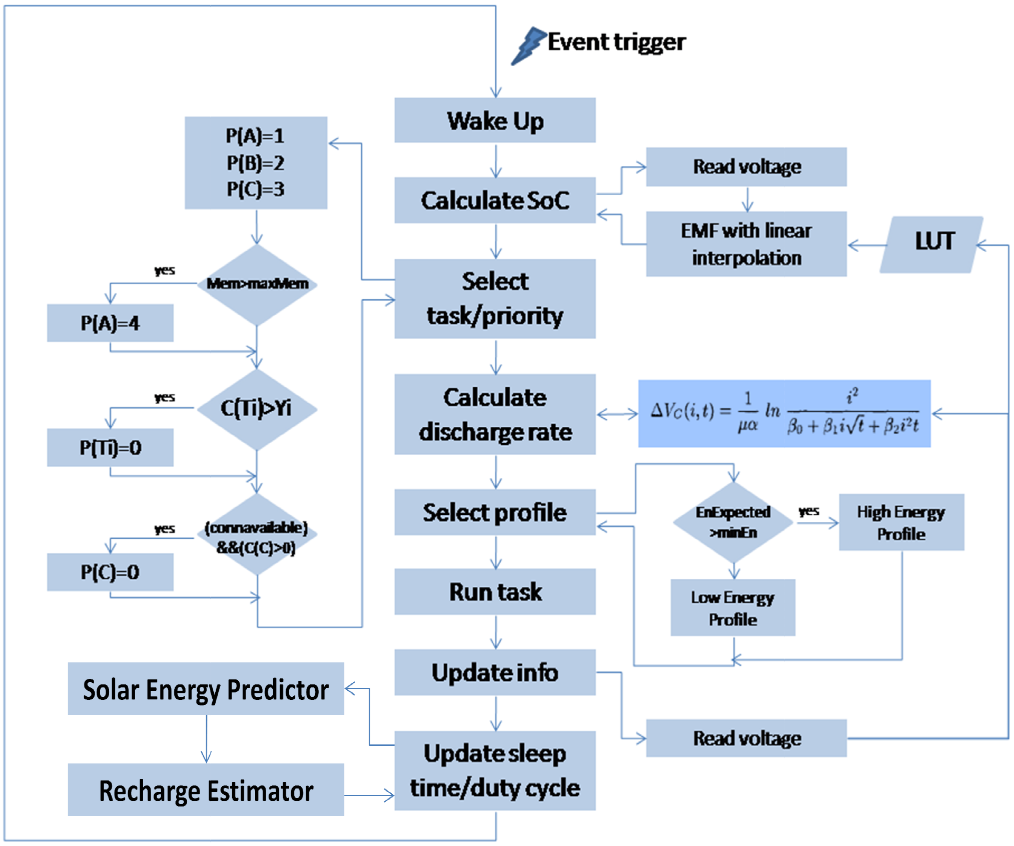


Figure 3.3 Updated activity diagram

The system is provided with:

- An implemented model to read the state of charge of the energy storage unit that uses possibly a lookup table with linear interpolation. This model compares the voltage of the battery or the supercapacitor before and after the running of the different tasks to calculate the discharge rate associated with each of them.

- A prioritization system that is able to dynamically change the priority of the scheduled tasks depending on the availability of energy and memory in the system.
- A selection of different energy profiles that are able to switch between a high energy profile which provides higher precision but also higher energy costs, and a lower profile which guarantees the progress of the tasks to the detriment of performance and precision.
- A Solar Energy Predictor (WCMA) that uses past values of solar irradiation to predict the energy that will enter the system in the future time.
- A Recharge Estimator uses the values from the solar energy predictor to obtain an estimation of the recharging speed of the storage unit to calculate from it a sleeping time. The system will have to sleep for the above mentioned time to wake up with the energy storage unit at a full capacitance.

All these improvements cut the dependency of the system on the event triggering. The system is now able to perform its own scheduled task depending on the energy availability but also on the requests of the external device that can require special commands.

The whole activity diagram and the included strategies can be extended to all the systems that present similar features and constraints.

In the next Chapter the solar energy predictor called Weather Conditioned Moving Average will be explained, optimized, tested with the comparison with similar algorithms.

In Chapter 5 the Recharge Estimator will be illustrated together with suggestions to utilize the same methodologies in comparable systems and to different energy storage techniques.

Chapter 6 will report a theoretical analysis of the task queues and different simulations about the queue evolution using the task dynamic prioritization and different energy profiles.

4. Solar Energy Predictor: Weather Conditioned Moving Average

4.1 Introduction

In the previous chapters the main problems that the developers have to face have been introduced. As explained before, recently proposed battery powered embedded sensor nodes are designed to gather data and then periodically transmit them wirelessly over a network. An example of such a sensor node is SHiMmer. For this kind of application, solar energy harvesting is the way to provide the highest power density and is very easy and economic to implement in comparison to other strategies. Thus, sensor nodes need to carefully consider strategies for spending energy on actuation, sensing, processing, and communication with the available energy from energy harvesting sources such as solar cells, and stored energy in batteries or supercapacitors. Managing the energy between these tasks is critical in order to ensure sufficient sensor node lifetime. Two main issues need to be solved in order to maximize the energy consumption of the sensor node:

1. The sensor node should use the extra energy available from energy harvesting sources once the batteries or supercapacitors are charged up.

The energy can be used for executing tasks such as actuating, sensing, processing, and transmitting.

2. When energy harvesting is either not available or minimal, the sensor node still needs to be able to respond to the outside queries for data (event-based triggering). Therefore, adaptation is needed as the energy availability, and the outside demands change.

In this Chapter we designed a new solar energy prediction algorithm that gives highly accurate estimates on likely available energy over the next time period. It is computationally simple and has a small memory footprint thus allowing it to be implemented in many different solar harvesting platforms. The algorithm takes into account both the available stored energy, as well as potentially available harvested energy from solar cells.

4.2 Motivations

The idea of the solar energy predictor is to be able to tell in some way the system how much energy will enter in a defined period of time. Which could be for a system the benefit of this knowledge? The reason why we started to exploit this issue is a problem introduced by SHiMmer but extendible to all the similar systems. This kind of systems, in fact, have very strict power constraints and they use to switch in a low power state, for example sleep mode, whenever they are not producing any work. But if the system is in sleep mode it is not able to monitor the energy storage unit, that can be either battery or supercapacitor or both, and is not able to know if the capacity is full or empty. The risk of this is that, if the system is sleeping and the storage system is full, all the solar energy from the environment that enters the system in this situation risks to be wasted. We know how big issue could be to waste energy in a very low power system and how big can be the interference of this waste on the system's performances. The first target is, therefore, to find a way to tell the system how long is possible to sleep before the energy storage unit is full. We want a system that assure us that no energy is wasted.

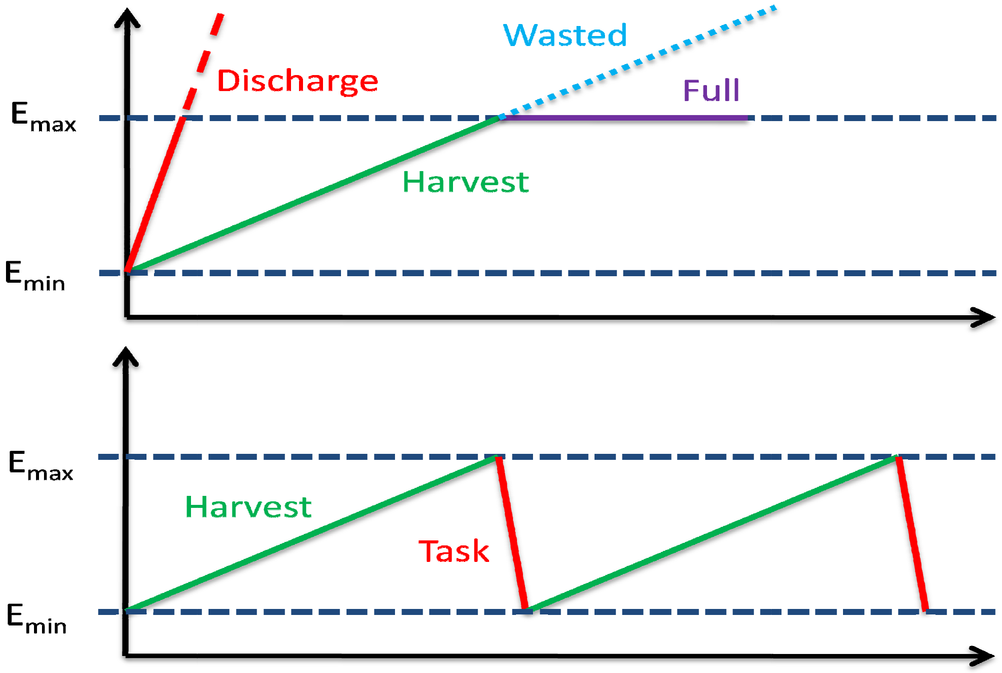


Figure 4.1 Energy wasted vs. efficiently used

The Figure 4.1 shows a comparison between a system that is not able to detect when the storage unit is full and a second system that is able to notice that the unit is full and uses the energy to perform some kinds of operations and therefore optimizes the use of energy without any waste.

A strategy that can solve this problem could be a hardware trigger system that cannot be always implementable. Another way to solve the problem is to periodically wake up the system to check the status of the energy storage unit. In case the energy state is higher than a threshold the system is instructed to perform some tasks to reduce the energy level and let new energy enter the system. In this last case, the period must be defined but must be taken into account that, in low power systems, the energy used to wake up and set up the system can be relevant compared to the energy used to perform tasks. From this derives that, if the period is too short, this strategy can be considered a failure because the energy saved using the sleeping state can be inferior than the energy used every time that the system is waken up to monitor. At the same time, a long period of monitoring can cause a decreasing of performance. If the period is too long, in fact, the storage unit can reach the superior limit between two consecutive readings and the energy is wasted. A new strategy must be explored with the target of giving the system a good estimation of the energy that will enter the system every period of time. Knowing this estimation the system will be able to know how long to sleep without waking up to check the energy level or, otherwise, to do it the lower number of times possible.

Another important problem that can be solved by using the energy prediction concerns the possibility of saving energy for the periods during the day when no energy, or little energy will enter the system. If the system, in fact, uses all the energy every time the storage unit is full, there is the risk that it is not able to respond to requests or to perform tasks since there is not enough energy available. In this case the energy predictor can be very useful. If the system knows that little energy enters the system in a close future, it can choose not to use all the energy but to save some to be able to run other tasks later.

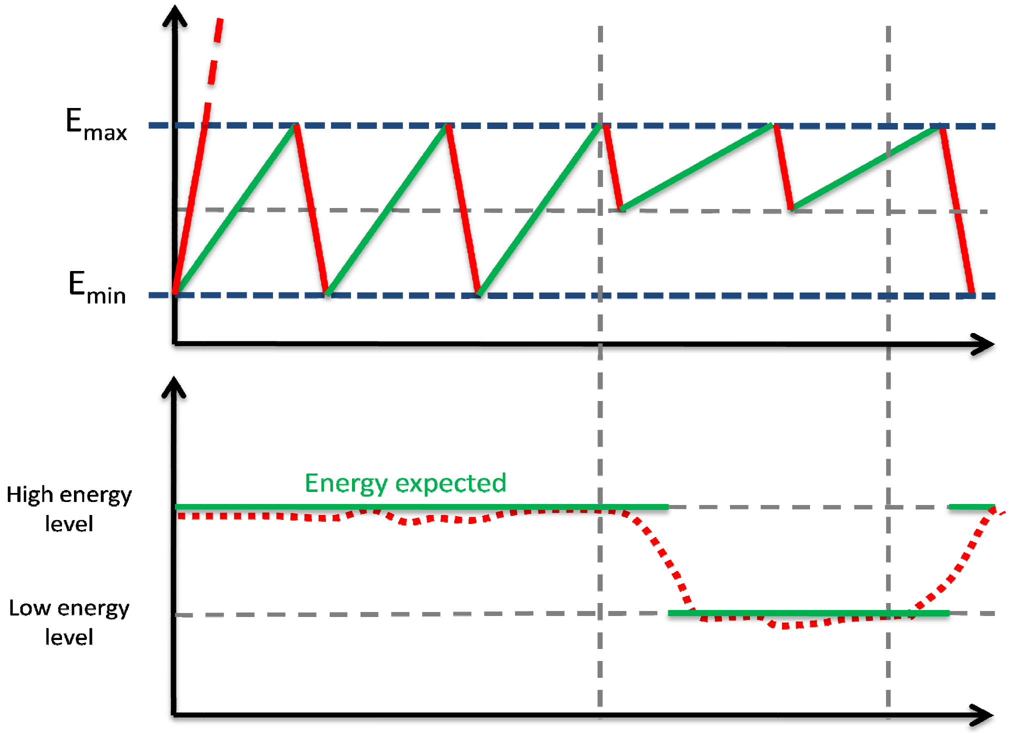


Figure 4.2 System adaptation in dependence of energy

The Figure 4.2 shows an example of system that is able to adapt the use of energy in dependence on the level of energy that will enter the system. At the beginning the system expects a high flow of energy and for this reason it can use a big amount of energy to run applications. In the following time, since the system expects a lower level of input energy, it switches the mode to a power saver mode and decides not to consume all the energy until the lower limit but to save some of it. That is just an example but it shows one application among many whose performances can considerably increase knowing the information about the future energy level.

4.3 Harvested energy from solar cells

Our aim is to find a good and “cheap” predictor for solar energy that is able to have a good estimation without consuming precious energy and memory. The typical graph of the energy enters from a solar cell in one day has a shape similar to a bell.

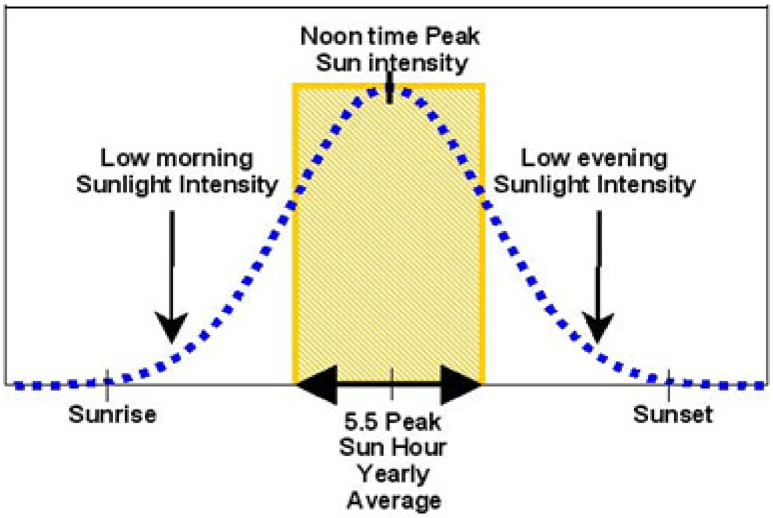


Figure 4.3 Daily sun profile

Figure 4.3 shows the changes of the solar energy during a sunny day. Another important issue to take into account is the seasonal changing during the year. During the summer, in fact, the graph will be different compared to the winter, both considering the intensity of the light and considering the length of the “bell”. That is because the sun rises before and sets later in the day, and has also higher intensity because it depends on the earth’s inclination.

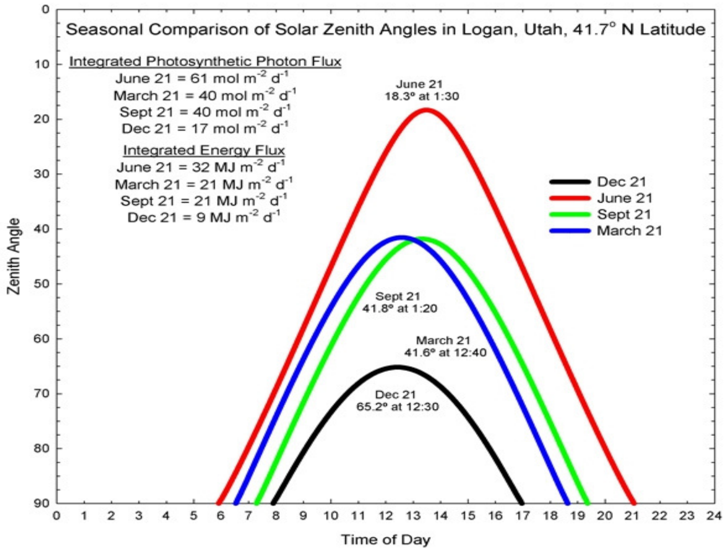


Figure 4.4 Seasonal changes in time

Figure 4.4 shows the seasonal changes in terms of time of the solar irradiation due to the earth's inclination.

Another important problem that we faced implementing our system is the weather changes during the day that in some regions of the world can provide important differences in energy levels. Finally, we add an important consideration: the position of the device can considerably influence the graph of the energy adding to it picks or height differences due to reflexes or shadows, but, since these differences will repeat in different days they can be used to create a better prediction in the following days.

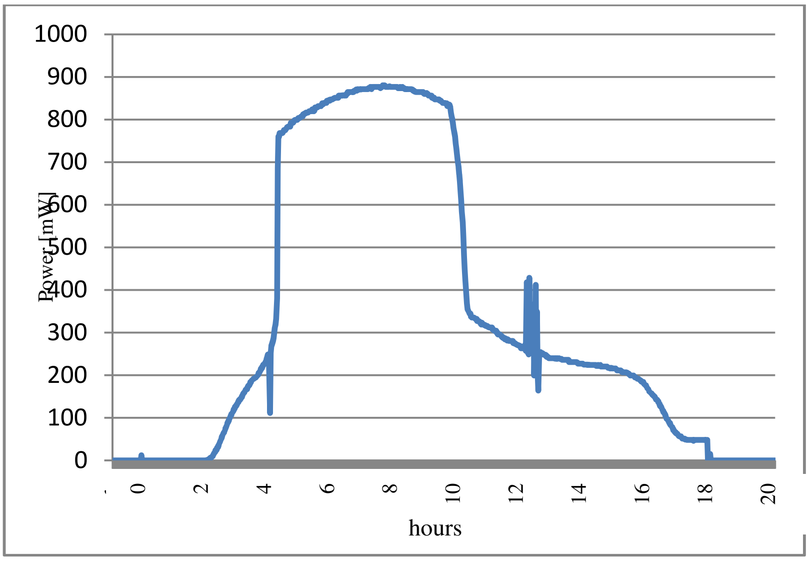
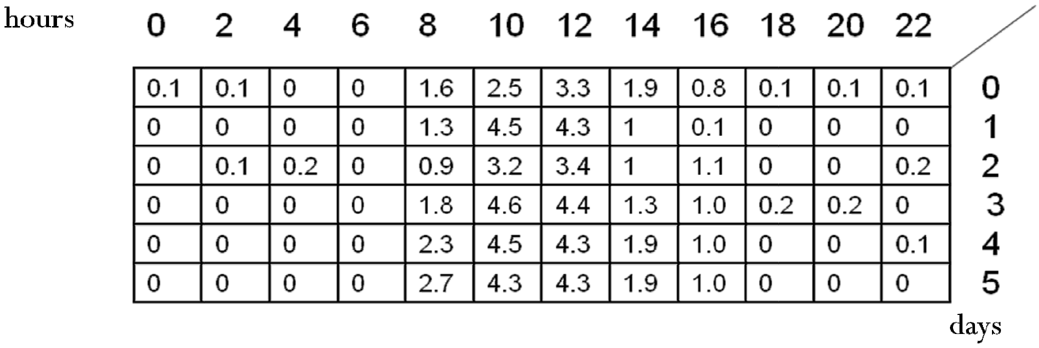


Figure 4.5 Daily energy profile in UCSD lab

The figure shows the energy level read from a window of a CSE department office in UCSD campus. It shows that real experiments usually give results that differ from theoretical analysis in ideal environments. These differences are due to shadows because of objects that cover the cells but, in most cases, are periodical with the period of a day and can be used to have a better prediction.

4.4 Implementation

Our solar energy predictor is named Weather Conditioned Moving Average (WCMA) [1] because it is able to consider how the weather condition affect the future solar irradiation. The WCMA base its idea on the collection of energy values that enter the system in different time in the past. We decided to collect these values in a matrix where every column represents a different moment in the day and every row represents a different day.

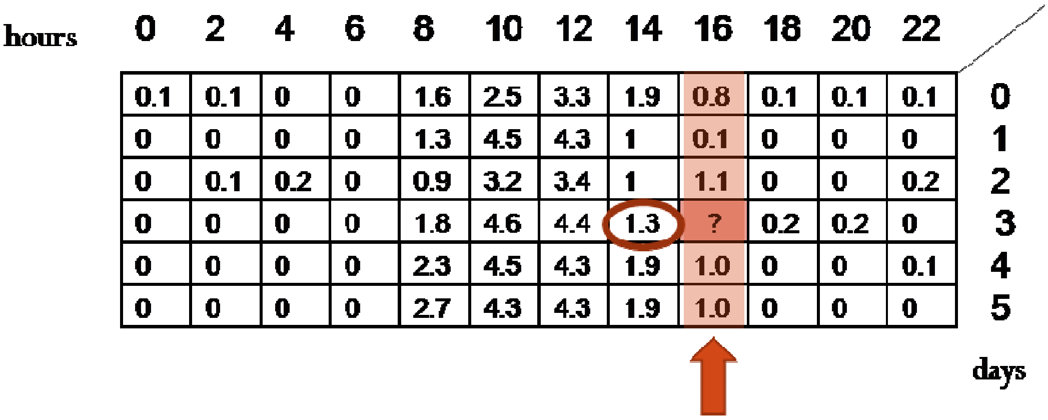


hours	0	2	4	6	8	10	12	14	16	18	20	22	days
	0.1	0.1	0	0	1.6	2.5	3.3	1.9	0.8	0.1	0.1	0.1	0
	0	0	0	0	1.3	4.5	4.3	1	0.1	0	0	0	1
	0	0.1	0.2	0	0.9	3.2	3.4	1	1.1	0	0	0.2	2
	0	0	0	0	1.8	4.6	4.4	1.3	1.0	0.2	0.2	0	3
	0	0	0	0	2.3	4.5	4.3	1.9	1.0	0	0	0.1	4
	0	0	0	0	2.7	4.3	4.3	1.9	1.0	0	0	0	5

Figure 4.6 WCMA matrix of past value

It is important that all the cells of the table correspond to moments at the same distance in time, in the example figure this time is 2 hours. The table must also respect the constraints that depend on the use of memory and on the energy consumption.

Our target is to be able to provide a prediction of the new value of the cells. If we assume, for example, that the cell that corresponds to day 3 and hour 14 is the last one to be written because it corresponds to the present moment, we want to develop a system that can predict the value of the following cell on the same day. To do that we use the values of the cells of the same period of the day on the previous days.



hours	0	2	4	6	8	10	12	14	16	18	20	22	days
	0.1	0.1	0	0	1.6	2.5	3.3	1.9	0.8	0.1	0.1	0.1	0
	0	0	0	0	1.3	4.5	4.3	1	0.1	0	0	0	1
	0	0.1	0.2	0	0.9	3.2	3.4	1	1.1	0	0	0.2	2
	0	0	0	0	1.8	4.6	4.4	1.3	?	0.2	0.2	0	3
	0	0	0	0	2.3	4.5	4.3	1.9	1.0	0	0	0.1	4
	0	0	0	0	2.7	4.3	4.3	1.9	1.0	0	0	0	5

Figure 4.7 Average of correspondent values in previous day

The WCMA algorithm takes a weighted sum of the current energy collected at the present time with the values gathered during the past days from matrix *E*. The following values are then calculated as the equation.

$$E_{(d,n+1)} = \frac{\sum_{i=1}^{D} E_{(i,n+1)}}{D} \tag{4.1}$$

Where *E(i,j)* is the energy in the matrix of the jth sample on the ith day and *D* is the number of days that correspond to the number of rows in the matrix.

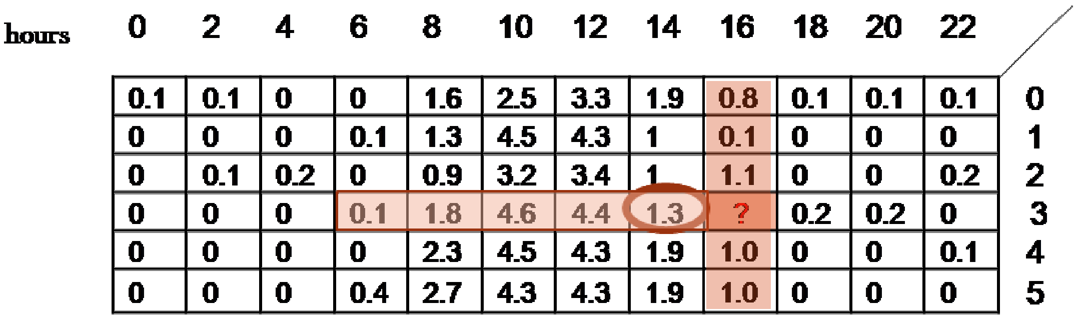
We also introduced an *α* weighting factor that decides how important the present value is compared to the calculated value. Since the applications we are considering can have access to this value we found that it could be useful to use it in the equation to calibrate the prediction avoiding over range results.

$$E_{(d,n+1)} = \alpha \cdot E_{(d,n)} + (1-\alpha) \cdot \frac{\sum_{i=1}^{D} E_{(i,n+1)}}{D} \tag{4.2}$$

The *α* value is also useful if the application does not need the values at the end of the period (as predicted in Equation (4.1)) but it needs an average value during the whole period of time.

This algorithm is now able to consider the seasonal changes by adapting to the change in the hour of sunrise and sunset and to the differences in solar power among seasons. We now want to improve the algorithm accuracy considering also the changing in weather conditions. Knowing the average values in the previous day permit, in fact, to give a good prediction in places where the weather conditions are stable; but a cloudy day after many sunny days, can produce a great error because of the great relevance of the weather conditions in the solar cells power input.

To avoid these errors we decided to improve the algorithm using the values in the current day but in the previous hours. These values can provide some information about the current weather conditions compared to previous days'.



hours

0	2	4	6	8	10	12	14	16	18	20	22	
0.1	0.1	0	0	1.6	2.5	3.3	1.9	0.8	0.1	0.1	0.1	0
0	0	0	0.1	1.3	4.5	4.3	1	0.1	0	0	0	1
0	0.1	0.2	0	0.9	3.2	3.4	1	1.1	0	0	0.2	2
0	0	0	0.1	1.8	4.6	4.4	1.3	?	0.2	0.2	0	3
0	0	0	0	2.3	4.5	4.3	1.9	1.0	0	0	0.1	4
0	0	0	0.4	2.7	4.3	4.3	1.9	1.0	0	0	0	5

Figure 4.8 *K* previous values

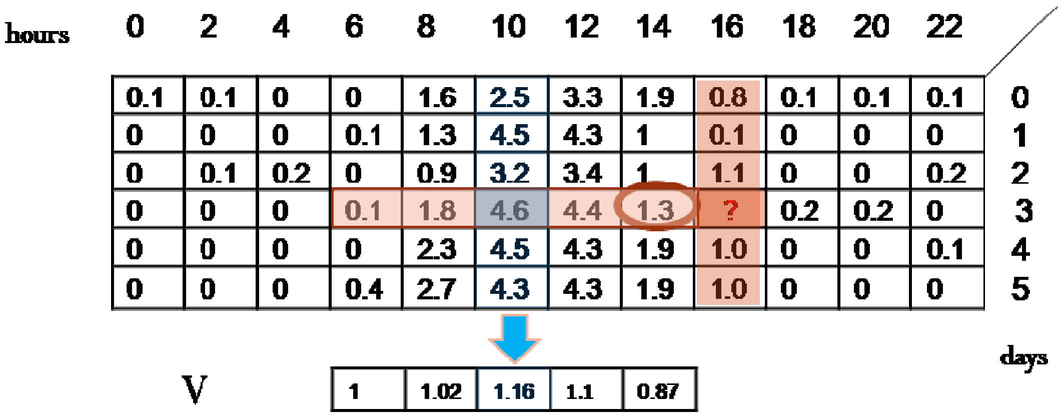
We created a new weighting factor, *GAP(d,n,K)* where *K* is the number of past intervals considered. Factor *K* should be large enough to consider the weather during the day but small enough to ignore periods of time that are not related to present conditions. For example, we do not want to consider night time when we are predicting energy available during the day. To compute the *GAP* factor, we define a vector $V^{d,n}$:

$$\overline{V}^{d,n} = \left[v_1^{d,n}, v_2^{d,n}, \ldots, v_K^{d,n} \right] \tag{4.3}$$

Every value of the vector correspond to the fraction represented in the following equation:

$$v_k^{d,n} = E_{(d,n-k+1)} \Bigg/ \frac{\sum_{i=1}^{D} E_{(i,n-k+1)}}{D} \tag{4.4}$$

On the numerator there is the correspondent value on the same day, while the denominator contains the average solar energy at a certain time period *n*, over a number of days *d*. This result is a number that represents the difference between the weather on the current day and the previous days in the correspondent range of time. This number is higher than the value 1 if the weather conditions of the present day are better (there is more light) than the previous ones, and it is lower than 1 vice versa.



hours	0	2	4	6	8	10	12	14	16	18	20	22	days
	0.1	0.1	0	0	1.6	2.5	3.3	1.9	0.8	0.1	0.1	0.1	0
	0	0	0	0.1	1.3	4.5	4.3	1	0.1	0	0	0	1
	0	0.1	0.2	0	0.9	3.2	3.4	1	1.1	0	0	0.2	2
	0	0	0	0.1	1.8	4.6	4.4	1.3	?	0.2	0.2	0	3
	0	0	0	0	2.3	4.5	4.3	1.9	1.0	0	0	0.1	4
	0	0	0	0.4	2.7	4.3	4.3	1.9	1.0	0	0	0	5

V					
	1	1.02	1.16	1.1	0.87

Figure 4.9 Vector *V* representation

Since the weather conditions can change many times over the same day it is better to give more importance to the values closer to the present one. A weighting vector P, is then defined as:

$$\overline{P} = [p_1, p_2, \ldots, p_K] \tag{4.5}$$

Each element in vector *P* is inversely proportional to the distance from the present value p_k:

$$p_k = 1 - \frac{K - k + 1}{K} \tag{4.6}$$

This equation guarantees that the relevance of the values close to the present value is higher than the farther ones.

The weighting factor, *GAP(d,n,K)*, for a day *d* at an interval *n* using *K* past values is computed using the equation:

$$GAP_K^{d,n} = \overline{V}^{d,n}(K) \times \left(\overline{P}(K) / \sum \overline{P}(K)\right)^T \tag{4.7}$$

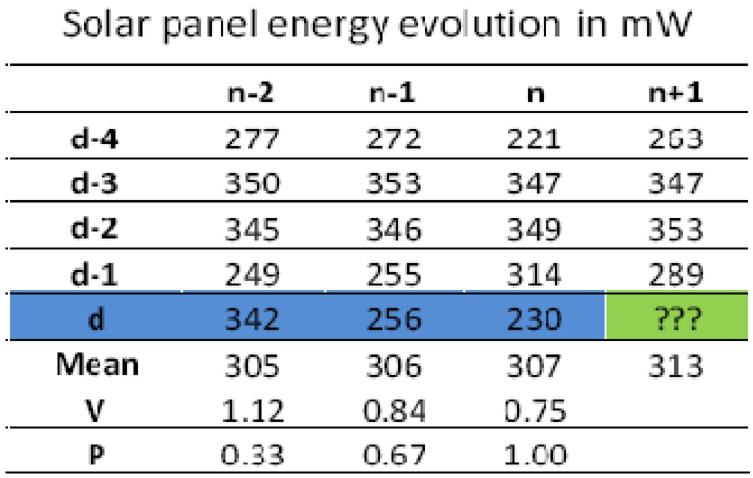
It represents the sum of values in the vector *V* weighted using the vector *P* and it respects the same features as all the elements of *V*.

The *GAP* value is, in fact, close to 1. If it is above 1, then there is more energy available because the current weather is “better” than before, whereas if it is below 1, then there is less energy because the weather is "worse." Finally, we use this value to calculate the prediction. The predicted energy for the next time interval is computed using:

$$E_{(d,n+1)}^{GAP} = \alpha \cdot E_{(d,n)} + (1-\alpha)\cdot GAP_K^{d,n} \frac{\sum_{i=1}^{D} E_{(i,n+1)}}{D} \tag{4.8}$$

With this improvement the prediction is able to consider both the seasonal and the weather condition changes using a very simple computation level and moderate memory occupation.

In the experimental section we show that our algorithm is within 10% of the measurement.



Solar panel energy evolution in mW

	n-2	n-1	n	n+1
d-4	277	272	221	263
d-3	350	353	347	347
d-2	345	346	349	353
d-1	249	255	314	289
d	342	256	230	???
Mean	305	306	307	313
V	1.12	0.84	0.75	
P	0.33	0.67	1.00	

Figure 4.10 Solar energy prediction example

The table shows an example of how WCMA computes the GAP_K factor for the next predicted value *E(d,n+1)* with *D=4, K=3.* The *Mean* vector contains mean values of the previous four days, *V* has the quotient of the elements in row *d* divided by the *Mean* (element by element), and *P* is the weighting factor for *V*. Finally, the GAP_K value is defined as follows:

$$GAP_K = \frac{(1.12,0.83,0.75)\times(0.33,0.67,1.00)^T}{\sum(0.33,0.67,1.00)} = 0.84 \tag{4.9}$$

And the predicted value with α=0.7 is:

$$E(d, n+1) = 0.7 * 230 + 0.84 * (1 - 07) * 313 = 239mW \tag{4.10}$$

4.4 Optimization

In this section we show how to determine the appropriate size of the matrix and the coefficients *K* and *α* to optimize the accuracy of the prediction. The aim is to minimize the error in prediction by calculating the optimal values for number of samples taken per day *N*, number of past days *D*, number of past values *K*, and weighting factor *α*.

In order to have more reliable results we decided to collect information about the energy that can enter a system through a solar cell. We set up a simple application using a previous version of the SHiMmer platform connected to a solar cell. The solar cell was placed facing a window in a part of the building in direct sunlight for most of the day. The platform was connected to a computer that memorized the values of input energy every 5 seconds for 45 days. We used 45 days of solar data to simulate our algorithm and compared the predicted values of the next period with the real measurements values.

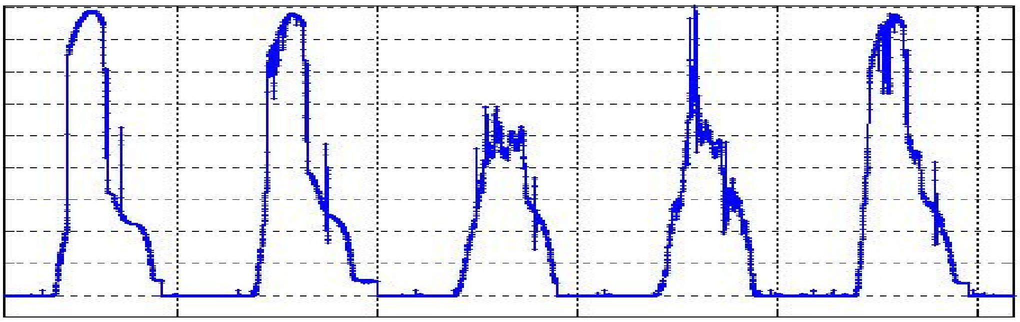


Figure 4.11 Energy profiles with different solar conditions

The figure shows the results we obtained through our application on 5 consecutive days. It is easy to understand the difference among higher values in sunny days, for instance the first, the second and the fifth days, and lower and less regular values on cloudy or partially cloudy days as the third and the fourth.

The error between the estimated and the present values is given in percentage by equation:

$$Err = \frac{1}{N}\sum_{i=1}^{N} abs\left(1 - \frac{E_{\mathrm{Re}al}}{E_{\mathrm{Pr}ed}}\right) \tag{4.11}$$

In order to have a higher accuracy of the results in the following part of the optimization we considered only hours of sunlight. We used, in fact, those values that have a minimum energy threshold of 20% of the maximum energy. Lower energy values, such as those during the night, affect this average error. However, the performance of the algorithm is unaffected when using all values.

We then focused our research on finding a way to determine an appropriate dimension for the matrix. Obviously, if the matrix records more samples every day the prediction is more accurate both because of the limitation of error in case of shadows or mirroring considering the previous days, and because of the calibration with the last values read, that give a higher reliability in the prediction. At the same time, using a lower sampling rate consistently increases the computational costs that affect the energy consumption and the memory usage. In low power systems these problems are very relevant. Most the applications use memory not bigger than some hundreds of Kilobytes and very small energy storage units. The energy cost increases with a bigger matrix for two main reasons: the computational cost increase exponentially due to more complicated and numerous calculations and the system must wake up a higher number of times consuming more energy in switching the state on and off. A tradeoff between prediction performances and energy and memory consumption must then be considered. The features and characteristics of the system will be relevant in this decision.

Figure 4.12 shows the influence of the number of samples per day on the result in terms of error and memory usage. When more samples are collected per day, the estimation of the next value is more precise at the cost of increased computation and memory energy consumption.

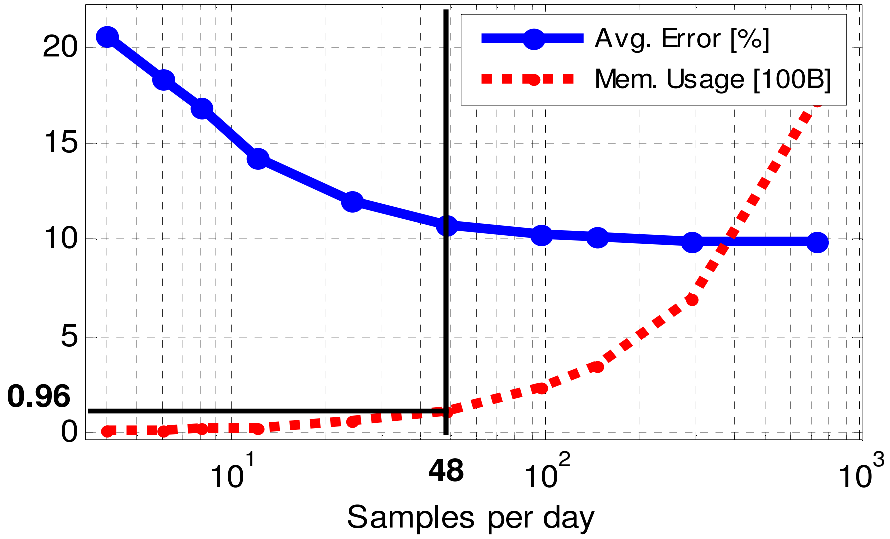


Figure 4.12 Optimization of number of values per day

Figure 4.12 shows that as the samples per day increase, the average error percentage decreases while the memory usage and the computational cost increase dramatically. On the other hand, too low a sampling rate would not give us the sufficient data to calculate the rate of charge in the energy storage unit which would make the sensor node calibration difficult. We found that for an average low power application as SHiMmer, the accurate and effective solution is with *N*=48 samples per day resulting in a duty cycle of 30 minutes and a memory usage of just few hundreds of bytes.

To decide the best values for the weighting factor *α* and for the other dimensions of the matrix we decided to use the Matlab platform to calculate the average error using many different values of our variables and a fixed number of samples per day. In the following part some of these simulations are presented. The following plot presents the error changes in dependence on the changes of two variables. In our tests we considered the result error in dependence on all the possible changes of variables but, in order to explain our results the plot of a system with 3 or more variables would be less readable.

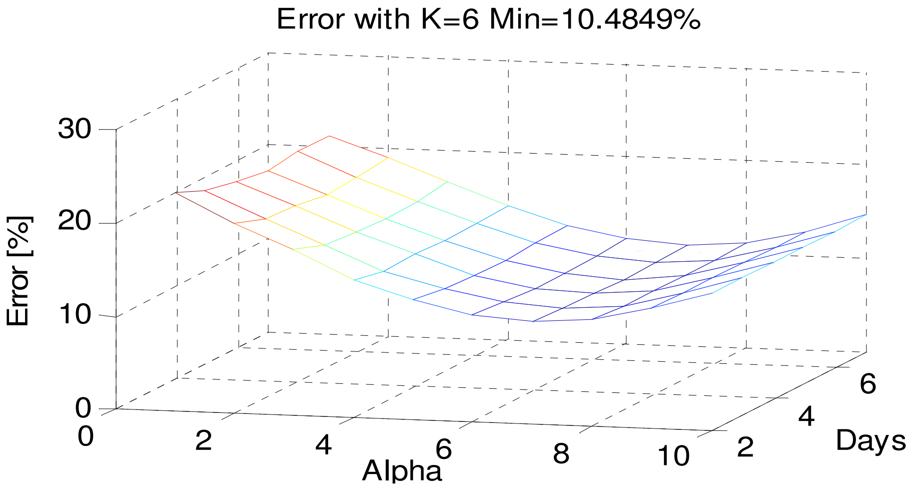


Figure 4.13 Estimated error for N=48 and K=6

Figure 4.13 shows the estimated error in prediction as a function of the weighting factor *α*, and the number of days *D*, for a fixed number of past values *K*=6. Selecting a weighting factor *α* at 0.7 gives a minimal error independent on the number of days used.

We then fixed the resulted *α* factor to optimize the value of the previous days *D* and the number of the past samples of the same day *K*.

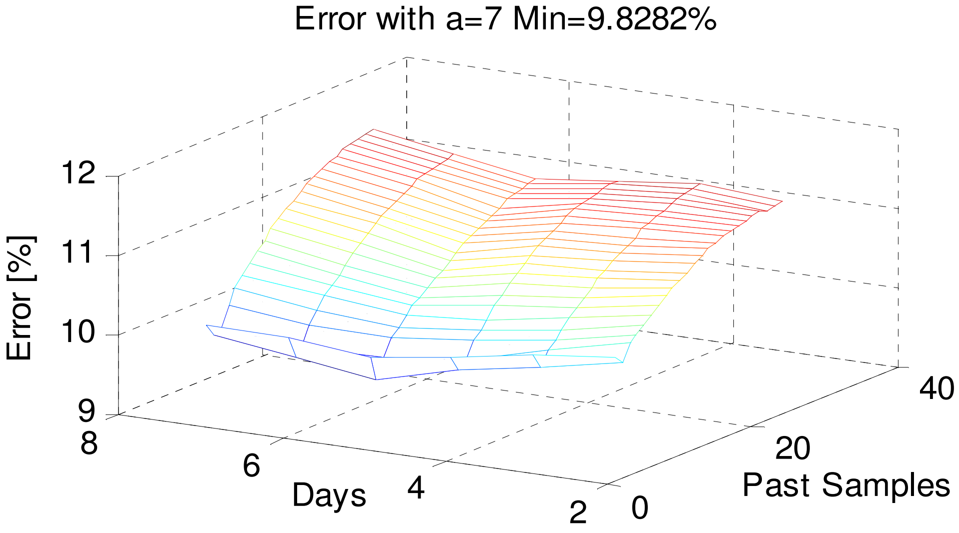


Figure 4.14 Estimated error for N=48 and α=7

Figure 4.14 shows the resulting plot of the prediction error versus *D* and *K*. If the number of past samples *K* is above 5, then the error increases quickly because it takes into account too many past solar energy values. Since the number of past days does not influence the error as much as the number of past samples, we can use fewer days to lower the computational cost with great accuracy of estimation.

Given this analysis, we select the following parameter values to minimize the error: *D*=4 days, *N*=48 samples per day, *K*=3 past samples, and *α*=0.7. The WCMA is now able to predict the following value with good results by spending less than 400 Bytes of memory in SHiMmer platform. That is because every value needs only 2 Bytes that can potentially be reduced to 1 Byte.

The computational cost is given by the equation $C = D * (K^2 + 1)$ that involves a very small number of operations also with different values of variables and consequently low energy costs.

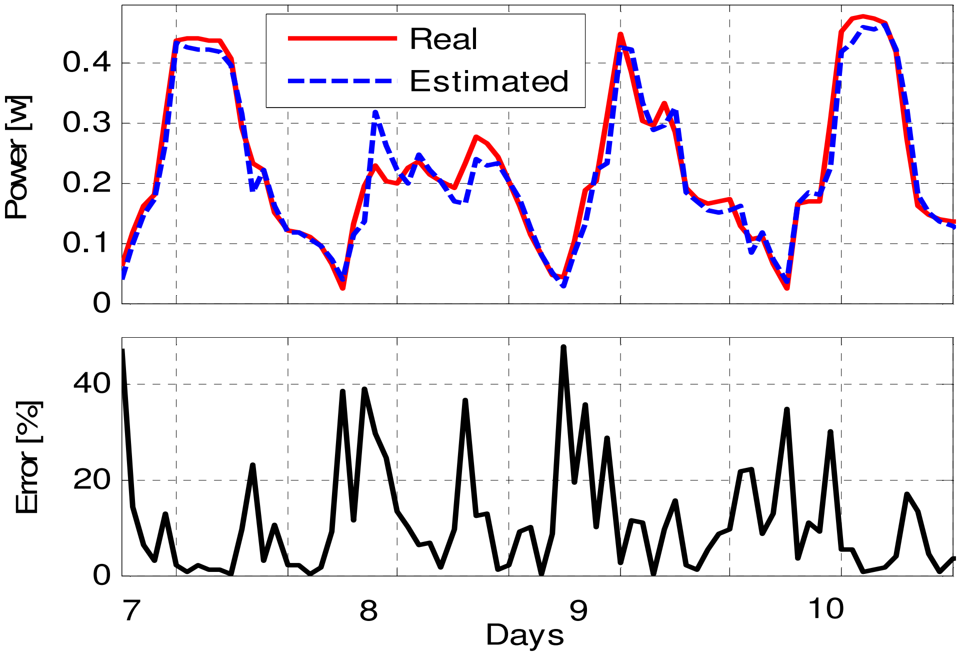


Figure 4.15 Error evolution for N=48 and α=7

Figure 4.15 shows four consecutive days of real solar samples and predicted values. Even for changeable weather conditions, as in days 8 and 9, the predictor rapidly adapts to provide accurate results. An estimate error is shown in Figure 4.15 below the current power values. We found that an average error is only 9.8% over all the 45 days of the collected data, reading and waking up the system just one real value every half an hour.

4.4.1 Comparison with EWMA

To prove the reliability of the algorithm we next compared it to a commonly used algorithm that can predict results for similar applications: the Exponentially Weighted Moving Average (EWMA) [9].

Other researches have used EWMA for solar evolution prediction [75], connecting it to an adaptive duty cycle that reflexes the environmental condition due by the presence of the sun.

The principle of the EWMA is to apply a weighting factor to the previous value that is constantly decreasing. In this way a greater distance in time from the current value corresponds to a lower importance in the prediction but, at the same time, the prediction takes into account every single value ever read with different relevance.

Calling $EWMA_{PRED}(x+1)$ the new predicted value, $EWMA_{REAL}(x)$ the last read value and ρ the weighting factor, the EWMA algorithm follows the following equation:

$$EWMA_{PRED}(x+1) = \rho \cdot EWMA_{REAL}(x) + (1-\rho) \cdot EWMA_{PRED}(x) \quad (4.12)$$

Basically to predict the new value, the algorithm sums the last read value to the previous value predicted, both weighted with ρ and $1\text{-}\rho$. If the factor ρ is high (close to 1) the last read value maintains more importance in the sums and vice versa.

To apply the EWMA algorithm to an application similar to ours, an array of predictions must be saved in memory and it must contain a number of values equal to the number of samples collected in a day.

Considering $E_t(d)$ as the energy available by the solar panel at a time t on the day d, $E_t^{EWMA}(d)$ is the predicted energy and ρ is the weighting factor. therefore the previous equation changes:

$$E_t^{EWMA}(d+1) = \rho \cdot E_t(d) + (1-\rho) \cdot E_t^{EWMA}(d) \quad (4.13)$$

To be sure to have an appropriate comparison we used the Matlab Optimization Tool to test the average error of the application using 10 different values for the weighting factor ρ from 0.1 to 1. We used the 45 days of values previously collected to test the algorithm using t from 1 to 48 and we found that with ρ=0.5 we obtained the lower average error.

We then compared the results of the prediction of our optimized algorithm with the EWMA algorithm using ρ=0.5.

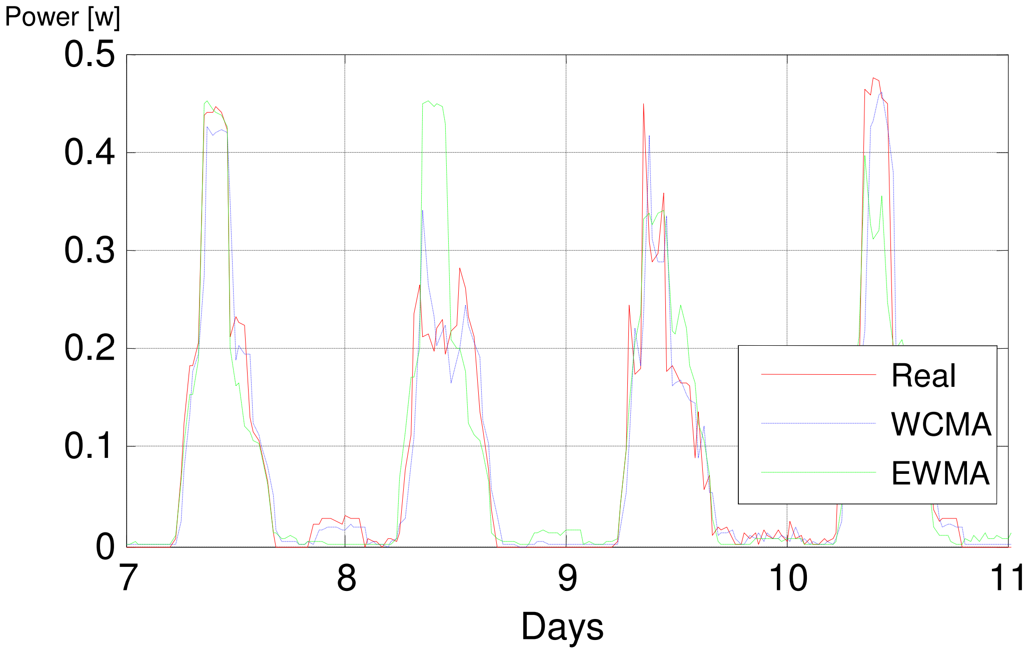


Figure 4.16 Prediction Algorithm vs. EWMA

The figure shows four consecutive days of the actual measurements in different weather conditions, and predicted values using WCMA and EWMA predictors. The first and the fourth days correspond to sunny conditions and the second and third are cloudy. Since EWMA only uses values from previous days at the exact same time period, if the weather condition changes from one day to another, this method presents a large error in prediction. On the other hand, our prediction performs much better since it not only uses the previous day's past values as at the same hour, but also the past values from the same day which contain the actual weather condition information. Looking at Figure 4.16 it can be easily noticed how the EWMA algorithm is not able to adapt to changes in weather condition and in presence of changes in consecutive days it produces a high error. EWMA gives an average error of 28.6% compared to 9.8% obtained by our new algorithm [1].

4.4.2 Comparison with Solar Predictor developed at the TIC laboratory of ETH of Zurich

We also compared the algorithm to another predictor developed at the Swiss Federal Institute of Technology used to contribute to adaptive power management techniques with the same target [76].

The predictor receives tuples *$(t,E_S(t))$* and delivers *N* predictions on the energy production of the energy source. The prediction intervals are of equal size denoted as *L* (in units of the basic time interval *T*). At the time *t*, the predictor produces an estimation $\tilde{E}_S(t+kL, t+(k+1)L)$ for all $0\leq k<N$ that is $\tilde{E}_S(t,k)$in short notation and represents (k+1)th prediction interval after *t*.

The estimation considers the periodicity of the solar power in $D=N*L$ such as the length of a day. The predictor collects information about the received energy in the current unit time interval and combines it with previously received information whose age is a multiple of days. It uses an exponential decay of old data with factor $0<\alpha<1$ and, in order to predict the desired time interval of length *L*, the predicted energy values for the corresponding intervals of length 1 are added. The long term prediction can also be supported by a short term predictor that directly uses the information of the received power $P_S(t)$.

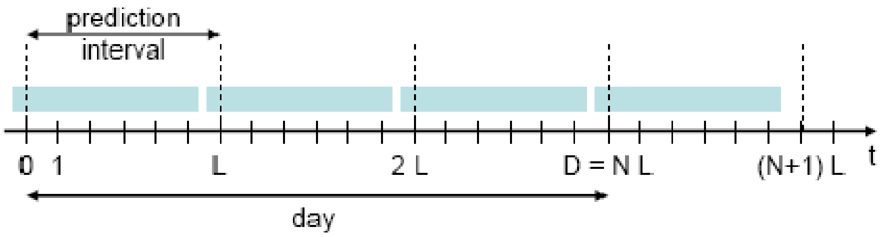


Figure 4.17 Prediction intervals in ETH's predictor

To be able to compare the two algorithms we just considered the short term part of this predictor because only the prediction of the next value is required.

Doing this the prediction can be simplified in the following equations:

$$E_P(t) = \min\big((1-\gamma) * E_N(t) + \gamma * R(t)\ ,\ E_N(t)\big) \tag{4.14}$$

$$R(t) = \beta * R(t-1) + (1-\beta) * E(t-1) \tag{4.15}$$

$$E_N(t) = \alpha * \ E_N(t-1) + (1-\alpha) * E(t-1) \tag{4.16}$$

Where $E_P(t)$ is the predicted value that directly derives from the previous predicted values and from the coefficients α, β and γ.

The 45 days of collected data have been used again to optimize the coefficients with the target of minimizing the error value using the error function in the equation (4.11). The values that we obtained are: $\alpha=0.3$, $\beta=0.4$ and $\gamma=1.0$.

Running a simulation with Matlab we realized that the two algorithms have a comparable computational cost. In fact, they both perform 19000 multiplications for the whole dataset. In terms of memory usage, the WCMA uses more memory slots in case of *D>1*, but, since it saves only short integer values it uses no more than few hundreds bytes. Due to the memory access time the memory usage of WCMA causes a higher execution time that in average microcontrollers is calculated to be 51µs compared to 14 µs of the predictor of ETH.

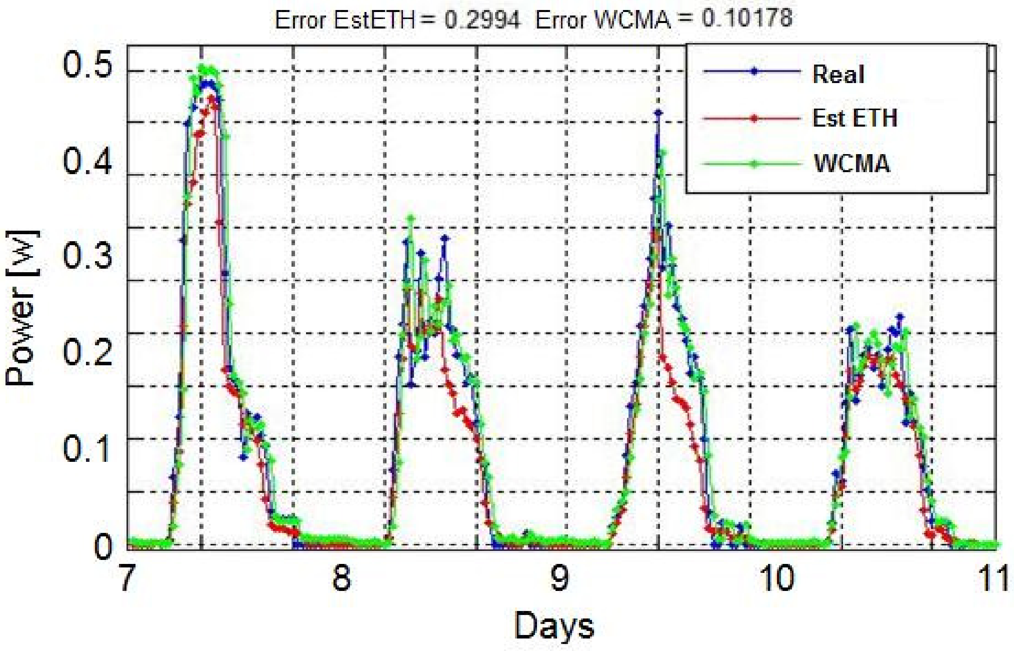


Figura 4.18 Comparison between WCMA and the predictor used at ETH

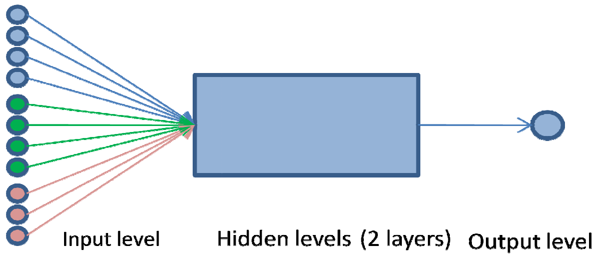
The Figure 4.18 shows the differences between the algorithms reported. We can notice that the predictor developed in Zurich presents a frequent underestimation of the predicted value that causes as a result a higher error value with an average of 29.9% compared to 10.1% of the Weather Conditioned Moving Average.

4.4.3 Comparison with Neural Network

Neural networks are special artificial intelligence structures based on the functioning of human brain and emulate the behavior of biological neural structures. The network is built from the union of many neurons whose connections are opportunely weighted. The weights that connect the neurons determine the relations between the input and the output levels. This is very useful to realize complex relations which analytical functions cannot realize, such as stock exchange trends or weather forecast.

Because of the high unpredictability of the solar energy prediction we decided to create a neural network to be able to evaluate if the results can produce a reliable prediction.

The network is realized using a supervised learning with error back propagation that means that the system needs a training where the input values are strictly connected with the required values. In this way the network uses the required values to modify the weights that connect the neurons. The training presents a computational cost that derives from the average error required. The lower is the average error in the training set, the higher is the reliability of the results in the simulation obtained but also the computational cost.



0	2	4	6	8	10	12	14	16	18	20	22	
0.1	0.1	0	0	1.6	2.5	3.3	1.9	0.8	0.1	0.1	0.1	0
0	0	0	0.1	1.3	4.5	4.3	1	0.1	0	0	0	1
0	0.1	0.2	0	0.9	3.2	3.4	1	1.1	0	0	0.2	2
0	0	0	0.1	1.8	4.6	4.4	1.3	?	0.2	0.2	0	3
0	0	0	0	2.3	4.5	4.3	1.9	1.0	0	0	0.1	4
0	0	0	0.4	2.7	4.3	4.3	1.9	1.0	0	0	0	5

Figure 4.19 Choice of input values

The input values are chosen to be coherent with the values saved in the matrix of the WCMA such as a vector of *d* correspondent values of the previous days, a vector of *k* previous values in the same days and a vector of *k-1* differential values that give information about the trend of the last values.

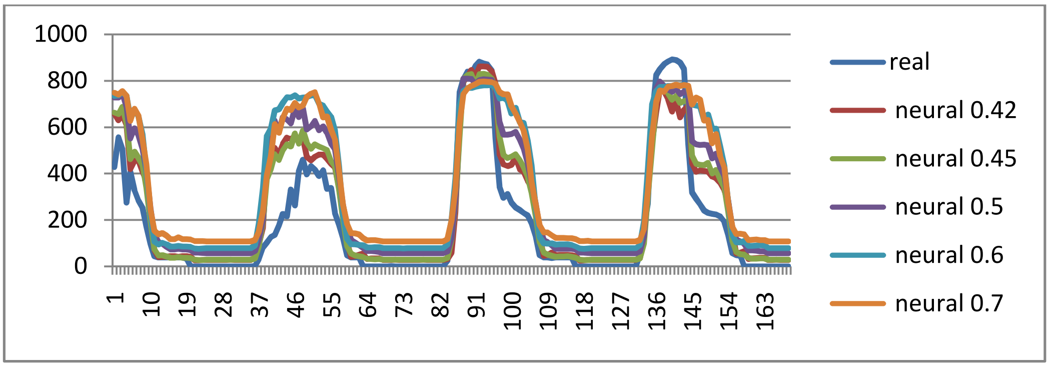


Figure 4.20 Prediction of neural network with different average error during training

The figure shows the different results of the neural network prediction in dependence on the error in the training. The changes of required error affect the computational cost that can vary from 7 cycles to obtain an error of 7% to almost 600 to obtain an error of 0.42.

We then compared the prediction of the neural network to the WCMA results.

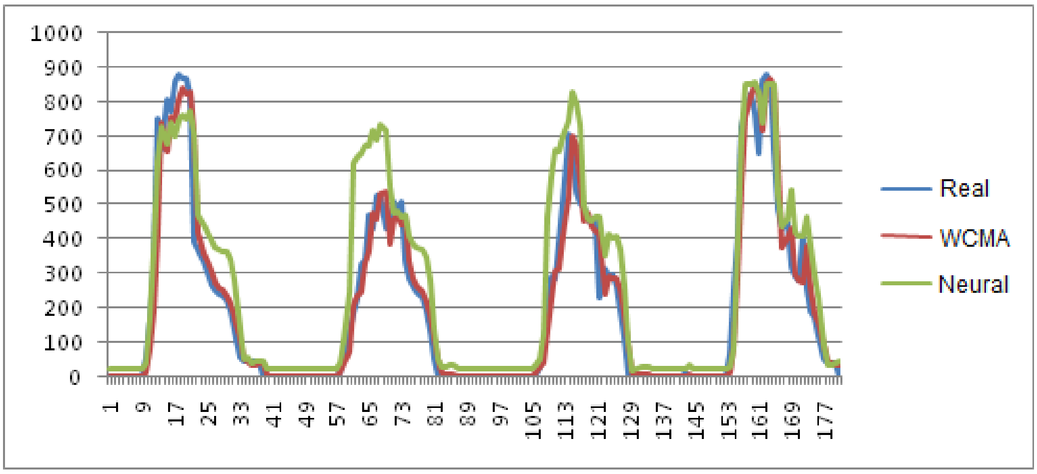


Figure 4.11 Comparison between WCMA and Neural Network prediction

We can notice that the Neural Network provides a prediction that follows the main shapes of the real energy values, but the WCMA presents much higher performances both in terms of prediction and computational costs.

Using the error function in equation (4.11) the Neural Network reports an error of 67.2% compared to 10.2% of the WCMA. The neural network algorithm is also hardly implementable into a low power system or microcontroller because it needs high computational capabilities and is not able to adapt to changes in the environment because it has not recalibration capabilities and it would need another training phase. This does not happen using the WCMA predictor that is able to recalibrate the system and produce a perfect prediction after only *d* days if a change in the environmental condition occurs.

4.5 Experimental setup

We assessed the performance of WCMA in a real-life energy harvested application related to SHiMmer platform.

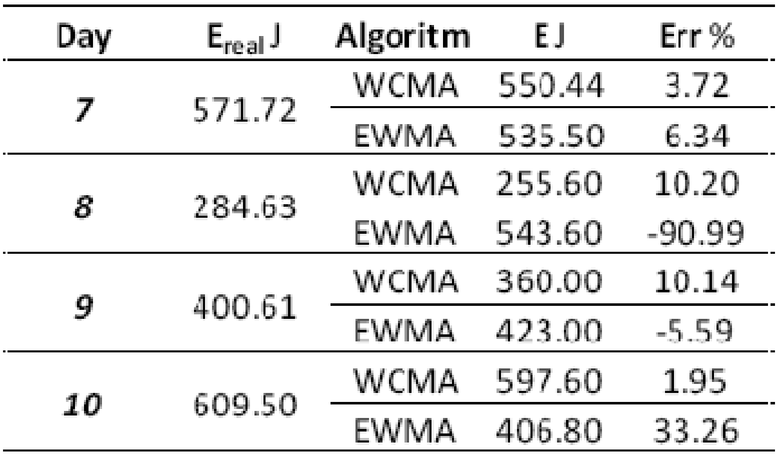
The types of task that can be executed on the platform are:

- **Actuate/Sample**: SHiMmer has 16 PZTs which result in 240 different paths that can be tested. This task is characterized by a mean power consumption of 1027mW and an execution time of 1.1ms, producing 20Kbytes of data.
- **Process**: Different algorithms can be employed to process the sampled data; they can vary from simple time domain patterns matching to complex filtering and frequency domain analysis. Thus, this task has a

mean power consumption of 680mW and an execution time of 55ms (light process) or 3470ms (complex process).

- **Send**: When no damage is detected, little data need to be sent over the radio, but in other situations the whole data record gathered is necessary to perform an analysis of the damage evolution over long periods. This task has a power consumption of 165mW and requires 6ms to send a packet of up to 255B.

Shimmer is usually placed in hard-to-reach locations, with limited wireless connectivity. It needs to execute the damage detection process daily. Thus, an energy estimator is mandatory to carefully choose when the damage assessment should be performed, and what kind of processing can be done with the currently available energy and likely to be harvested in near future.



Day	E_{real} J	Algoritm	E J	Err %
7	571.72	WCMA	550.44	3.72
		EWMA	535.50	6.34
8	284.63	WCMA	255.60	10.20
		EWMA	543.60	-90.99
9	400.61	WCMA	360.00	10.14
		EWMA	423.00	-5.59
10	609.50	WCMA	597.60	1.95
		EWMA	406.80	33.26

Table 4.1 Solar energy prediction results

The table shows the values of predicted and current energy available at noon during a period of 30min for the time record depicted in Figure 4.16. We focus on these specific times to illustrate the difference between EWMA and WCMA when it comes to intraday prediction.

As Table 4.1 shows, WCMA has a maximum energy prediction error of only 10%, while EWMA can have errors of up to 90%. This large EWMA energy estimation error can cause incorrect energy management decisions by SHiMmer. SHiMmer needs to Actuate/Sample, perform full Processing and Send the data obtained from all 240 PZT paths, in order to identify and localize damages in the structure. When there is not enough energy to perform this whole analysis, the complexity of processing task has to be reduced to ensure that at least some feedback is obtained on the current structural health. Thus, in the following experiments we evaluate how well WCMA and EWMA perform as energy availability estimators in situations where energy resources are at a premium and not all tasks can be executed. Checking the whole structure means that all 240 different paths are actuated/sensed, with the total energy cost of 240·1.2mJ=0.27J. Sending all these processed data requires 240·77.6mJ=18.6J. The rest of the available energy can be used to perform processing by the DSP. While the goal is to run full processing at 3.47s per path each day, when there is not enough energy available, then a fraction of paths runs only a light processing, resulting in the overall lower average processing time per path. We next report this DSP time per path assuming the perfect prediction of energy harvesting capabilities (oracle), or the prediction when using either WCMA or EWMA to predict the amount of energy harvested over the next 30min period. To illustrate our ideas we focus on the 30 min period around noon for each of the days outlined in Table 4.1.

- Day 7: Both WCMA and EWMA perform great results predicting for consistently sunny days resulting in the average DSP time per path of 3.26s and 3.17 respectively, compared to 3.39s by oracle predictor.
- Day 8: When a sudden change in weather occurs, WCMA is able to adapt quickly resulting in a much better processing allocation relative to EWMA. Average DSP per path time for WCMA=1.45s, while oracle is only slightly higher at 1.63s. In contrast, EWMA significantly overestimates the energy entering the system thus scheduling too many full processing tasks with average DSP time of

3.22s. As a result, SHiMmer runs out of energy before finishing the whole structure scan, resulting in a significantly worse result.

- Day 9: Both predictors offer a good approximation of the DSP per path since the weather is similar to the previous days. Oracle comes in at 2.34s per path, WCMA: 2.09s, EWMA: 2.48s.
- Day 10: EWMA underestimates the DSP time since the weather is better than the previous days; oracle got 3.62s, WCMA: 3.55s, EWMA: 2.38s. As a result, much less processing is done if EWMA is used for prediction, leading to suboptimal results relative to WCMA.

Overall, WCMA correctly predicts all the variations of harvested energy so that SHiMmer can fully utilize it to estimate the structural damage, while EWMA has significant difficulties during variable weather conditions.

Our solar energy harvesting predictor has shown to be very accurate and reliable in days of variable weather. This is extremely necessary so that our recharge estimator can accurately determine how long to sleep before the energy storage units are full as it will be explained in the next chapter. We have shown that if using an inaccurate predictor, the system energy storage unit will be unable to store more energy because they are full while, with our innovation, the system is able not to waste energy but to use it all for its target.

5. Recharge estimation

5.1 Introduction

Once we have predicted the amount of energy likely to be available from the solar cell, we next need to find out how long it will take to maximize our energy storage. Taken SHiMmer platform as our reference system we firstly considered to have supercapacitors as unique storage energy unit, but the following ideas could possibly be generalized to rechargeable batteries.

The amount of solar energy that enters the system strictly affects the rate at which the energy storage unit recharges.

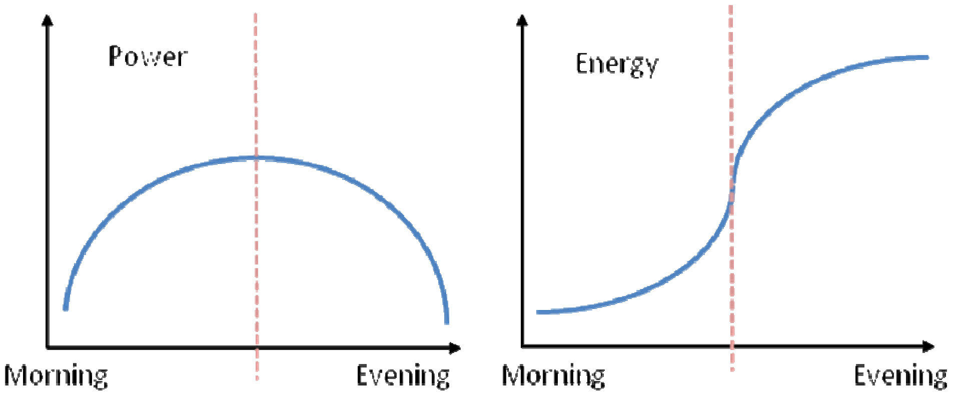


Figure 5.1 Ideal power and energy during a day

Figure 5.1 simplifies the solar energy changing during the day in a semicircular manner. As the solar energy increases at the beginning of the day, the charge rate also increases and as the solar energy decreases toward the latter part of the day, the charge rate also decreases.

Our goal is to track how much energy is stored in the system and to provide that information to our energy manager. If the energy manager is able to have a realistic information of the energy level in the system it then can decide when and if to execute some tasks, and in what order. The main target is still the same as explained in previous chapters that is that the system can switch to a low power state and wait until the energy level in the storage unit is high enough to perform the required tasks.

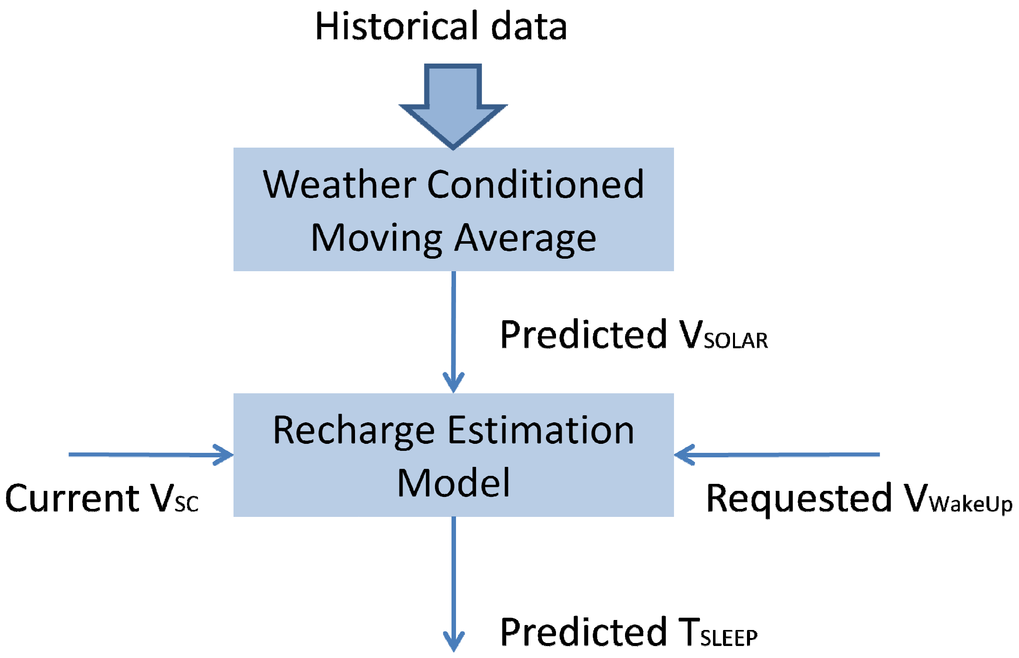


Figure 5.2 Recharge Estimator model

As explained in Figure 5.2 we want to create a Recharge Estimator that is able to receive the predicted value from the solar energy predictor, together with the current value of the energy storage unit and the value requested at the wake up. Given these inputs the estimator will provide a correspondent sleeping time.

In order to estimate how much energy is going to be stored in the system, we assume that the solar energy behaves as a current source. If we use a super-capacitor as an energy storage unit, the voltage in the super capacitor increases linearly during static solar conditions, so the charging rate can be defined as:

$$R = \frac{dE}{dt} \tag{5.1}$$

We decided to calibrate the maximum possible storage rate measuring the voltage of the supercapacitor during a sunny time period and use it to calculate the overall energy stored $E=CV^2/2$.

We then defined T_{Sleep} as the time the system needs to stay in a sleep state in order to recharge. T_{Sleep} is calculated by the following equation:

$$T_{Sleep} = \frac{E_{wUp} - E_{Act}}{R} \tag{5.2}$$

E_{wUp} represents the energy level at which the system is requested to wake up while $E_{present}$ is the present energy level. To determine T_{Sleep} taking the difference between the 2 energy levels E_{wUp} and $E_{present}$ is essential. We have now obtained the sleeping time for the system in order to charge the energy storage unit. It is calculated by using the solar energy harvesting predictor. If the sleeping time calculation is accurate, then the system will be able to harvest a maximum amount of energy from the solar panel and then use it effectively with the energy management strategies to maximize the performances, as will be shown in the following sections.

5.2 System calibration

The target of this section is to suggest some strategies to calibrate a single system in order to obtain fundamental values that permit to run the solar energy predictor WCMA obtaining from it reliable results.

First of all, we want to determine the rate of charge at any given solar intensity level, to predict the energy increasing produced by different energy levels. To do this, we program a calibration function that measures the voltage of the supercapacitor as it has changed during a small window of time.

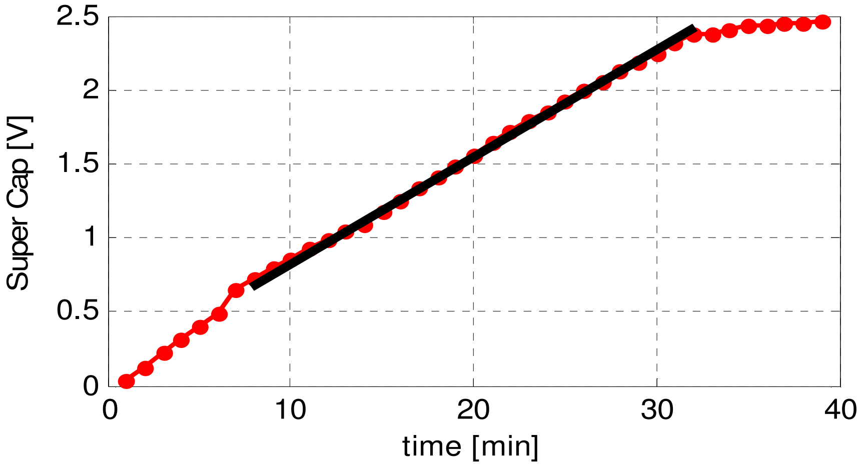


Figure 5.3 2.5V 50F super cap charging during noon

From Figure 5.3 we can see that the curve is very similar to a linear function in the main part of the voltage values but in the higher values. The linear function represented by the black line, in fact, fully covers the red line represented by the values we obtained from the system. This is acceptable because, most of the time, the supercapacitor is operating in the linear region.

Since the charge rate is higher in the central values than in the region where it is almost full (above 2.4V), if estimated linearly, then we can overestimate the voltage level which is beneficial because we avoid the region at which the supercapacitor recharges slowly. We then want to maximize the duty cycle of the system while permitting the supercapacitor to work in the efficient range of voltage.

5.2.1 Calibration algorithm

The calibration algorithm that we defined measures the supercapacitor voltage every minute for a period of ten minutes. Then, calculating the difference between each two consecutive voltages it is possible to find the change in voltage per minute. Finally, an average is taken over the set of differences resulting in the charge rate at a measured solar intensity as shown in the following equation.

$$R = \sum_{i=0}^{N-1} \left(V_{SC,i+1} - V_{SC,i} \right) \Big/ N - 1 \tag{5.3}$$

We ran the calibration algorithm every 30 minutes (corresponding to the duty cycle of the predictor) for three days resulting in a table of solar intensities with corresponding recharge rates.

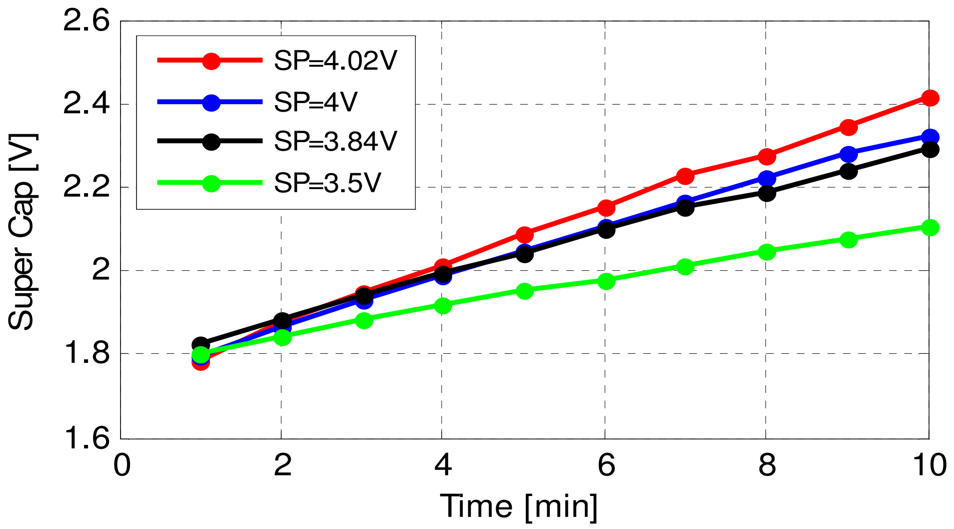


Figure 5.4 Supercapacitor charge for different solar conditions

The Figure 5.4 shows a sample of the data taken during the calibration algorithm showing how the voltage of the 2.5V 50F supercapacitor fluctuates as time passes for different solar intensities. We collected the data to fill the graph putting the solar panel under the direct sunlight during different periods of the day and some different artificial light sources in order to be able to cover the highest number of possible values of the input voltage.

We then collected all the data obtained in a wide range of light situations and we plotted all these points in a unique graph. The target was to obtain an equation that can in some way cover all the points in the graph and connect the value of the solar cell to the charging rate of the supercapacitor.

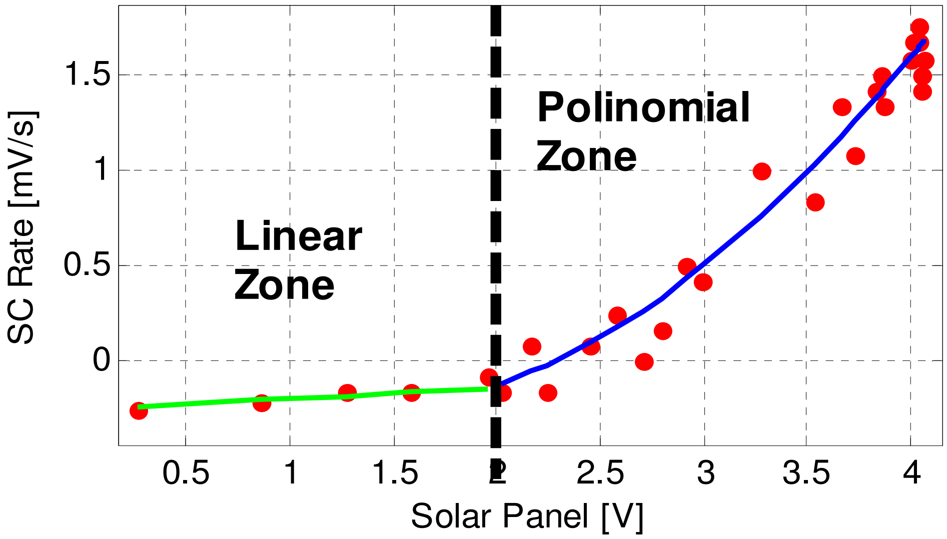


Figure 5.5 Charging rate estimation

Every red dot in the figure represents a value charging rate collected running the calibration for a solar panel voltage value. We realized that with our particular platform it was hard to find an equation that can fit close to all the data points. We decided to divide the graph into 2 different zones related to 2 equations: a linear zone with lower energy values and a polynomial (or exponential) zone for higher energy levels. We can notice that the whole linear zone presents negative values. That depends on the energy consumption present in the platform also in sleeping mode and some probable leakage current that together are higher than the energy harvested by the system during cloudy or night conditions.

We can finally relate all the data points with two equations. A second order polynomial R_P, is used to fit values greater than 2V and a linear equation R_L is used for values between 0 and 2V. The fitting is performed using least means squares. The results are shown in the equations below:

$$R_P = 0.23x^2 - 0.53x + 0.01 \tag{5.4}$$

$$R_L = 0.06x - 0.25 \tag{5.5}$$

We simplified the data point positions into equations. This is very important because using these two equations, the values can be easily adapted to many different microcontrollers without requiring any complex mathematical libraries.

5.3 Simulation and tests

After we obtained the equations that link the input energy to the charging rate, we set a simulation using the benefits of the equations with the solar energy predictor WCMA. Knowing the expected energy that will enter the system and the relative charging rate, we calculate the sleeping time T_{Sleep} to wait before the energy reaches a required level. We decided that 2.4V is our required level because we noticed that, when the supercapacitor has a higher voltage, the charging rate is lower with the same energy input (Figure 5.3). To simulate we utilized the collected data of the 45 days we used to calibrate the WCMA and we obtained the results of the following figure.

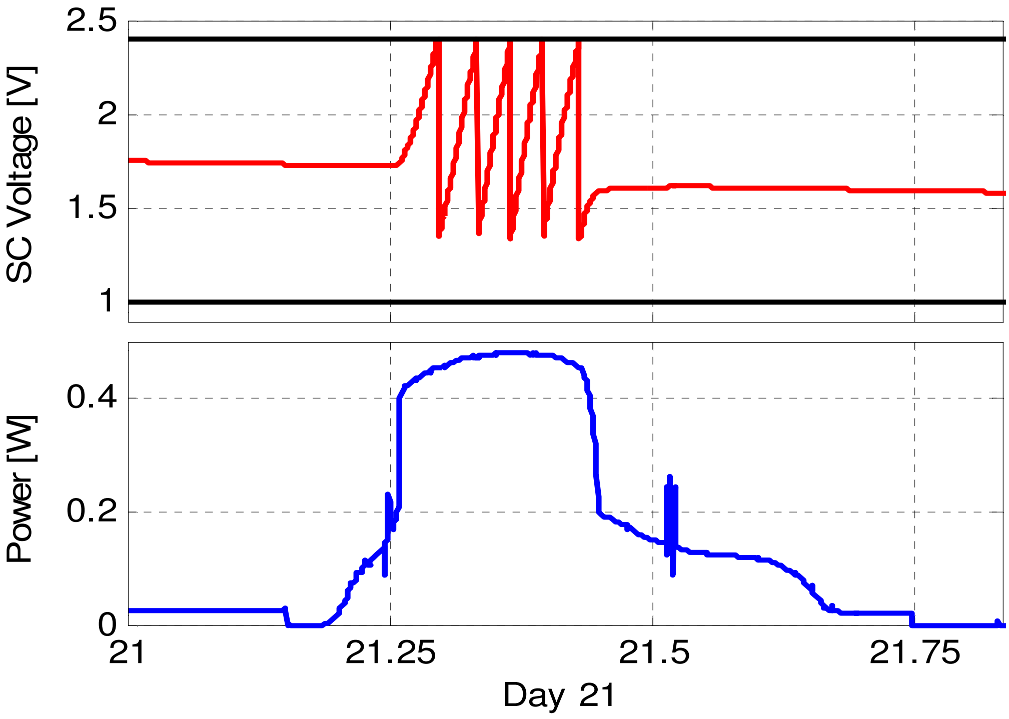


Figure 5.6 Simulation of the energy harvesting prediction

The figure shows the supercapacitor charging and discharging according to this estimation during a window of a day with good weather conditions. Equation(5.2) is used to estimate the length of time to recharge the supercapacitor up to 2.4V. When it wakes up, tasks are executed to consume energy so that more energy may be harvested because if the supercapacitor is full, then energy could not be added. To set up the simulation we considered the SHiMmer platform constraints. In Figure 5.6 it is evident that the algorithm works properly as it very rarely wakes up after the supercapacitors are above 2.4V. Thus, not much energy is wasted in the “slow” charging zone or at full capacity.

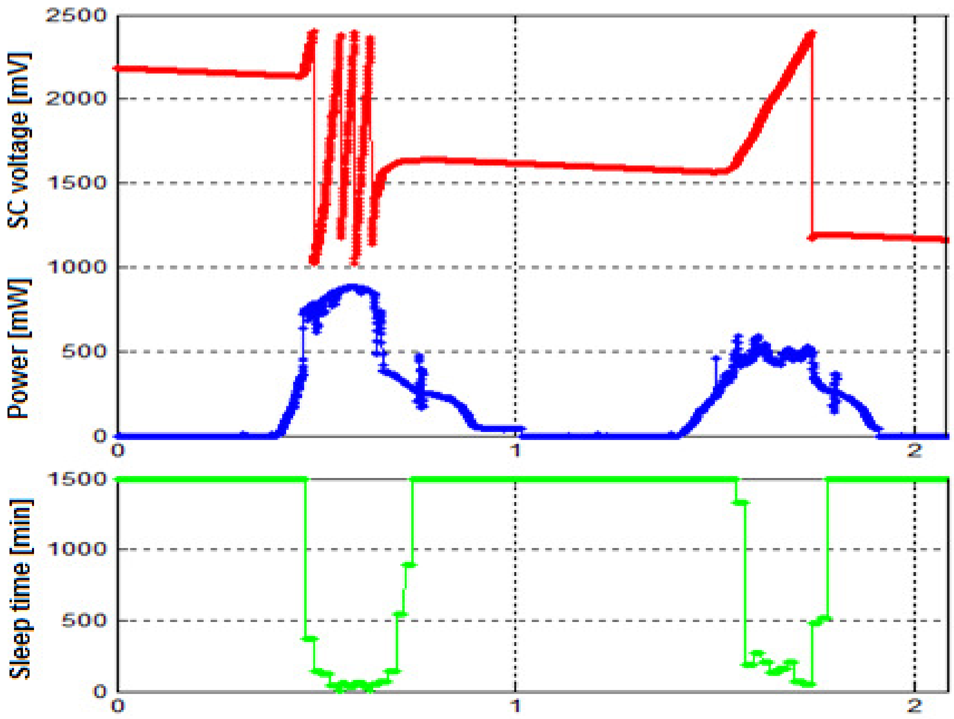


Figure 5.7 Simulation of the supercapacitor duty cycle related to the sleeping time

Figure 5.7 shows how the duty cycle of the supercapacitor load is able to adapt to the different solar condition. A two days' window is represented and we can notice that on the first day (sunny) the frequency is increasing and the shape of the sleeping time calculated by the recharge estimator can be considered dual compared to the energy input from the solar panel. The second day presents worse solar conditions that cause the sleeping time to be higher during the whole day. The time calculated is changing depending on the solar conditions, but also on the proximity of the current supercapacitor voltage to the requested value.

After the simulation we ran the algorithm in the real platform obtaining very good results.

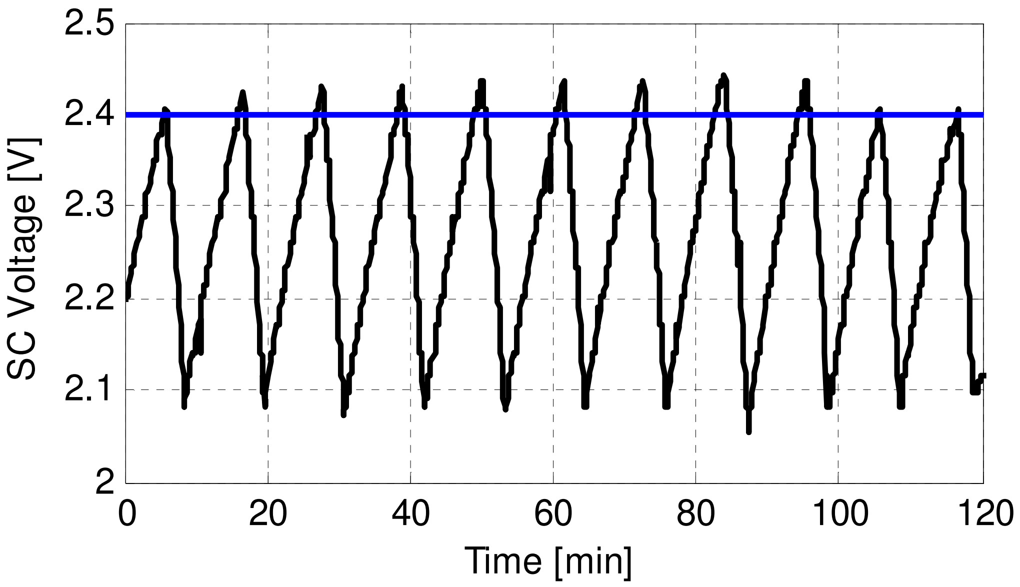


Figure 5.8 Shimmer test of the wake up estimator

The figure represents the charging and discharging of the supercapacitor in a range of about 2 hours in SHiMmer. The supercapacitor charges using the solar energy from a solar panel while the system is in a sleep mode. The system, before falling asleep, sets a wake up time calculated by the WCMA together with the Recharge Estimation Algorithm. To discharge the supercapacitor we used a task that discharge rapidly and constantly in the time. In order to run the test we simply used a high resistor. As we can see in Figure5.8, the algorithm is able to predict quite accurately when it reaches 2.4V and never reaches full capacity at 2.5V. This means that all available solar energy is harvested so that the node can use it.

5.3.1 Comparison with binary search algorithm

To compare the recharge estimator algorithm, we used a simple binary search algorithm [36] to also determine the recharge time. We ran the same experiment with SHiMmer platform using the same input values and we obtained different results.

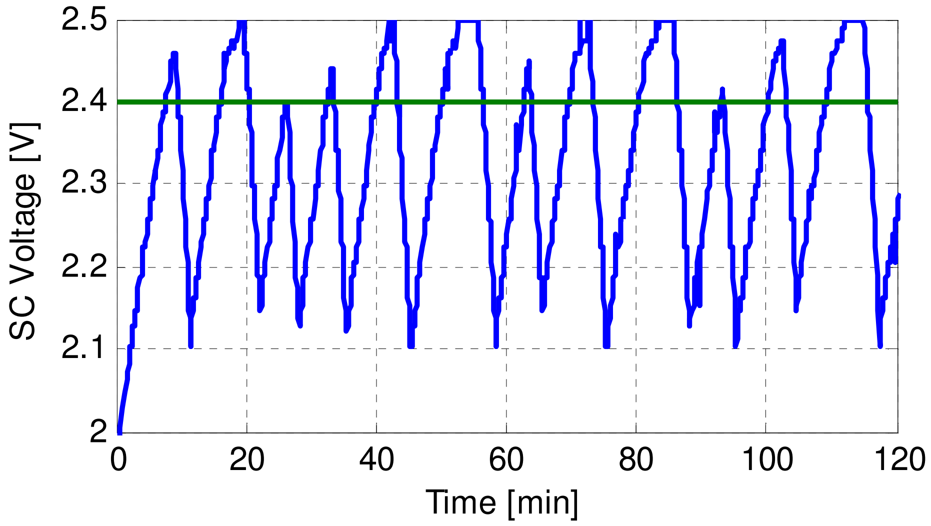


Figure 5.9 Shimmer test of the binary search wake up

The figure shows the results of this algorithm as also implemented on SHiMmer. It is possible to notice that the node usually wakes up after the supercapacitor reaches 2.4V and sometimes even reaches 2.5V. This means that when it is at 2.5V, then it is unable to harvest more energy, thereby wasting available harvestable energy. Using our recharge estimator, the remains above 2.4V (region of slow recharge) for 12.6 minutes as opposed to 36.6 minutes using binary search. This shows that our recharge estimator is more than 3 times more efficient in harvesting energy than binary search.

6. Theoretical analysis

6.1 Introduction

This section explains the main issue that brings us to focus our attention on the evaluation of an energy manager that is able to help the system to schedule and prioritize different tasks with low power constraints. We took into account the SHiMmer platform to define the main constraints and the main problems that our typical system must be able to solve. In this chapter, in fact, we describe an energy manager that is able to schedule tasks of different priority while taking into account their energy needs. We first look at the constraints of the system to see how they affect the energy manager so that it can operate with maximum efficiency. Then we analyze it to determine the average queue time for a task. Finally, we look at different scenarios in which the energy manager can operate to show its flexibility for maximum performance.

6.2 System Constraints

In this work we assume that the sensor node has a way to both store and harvest energy, and uses that energy to perform a combination of actuation, sensing, processing and communication tasks. Furthermore, in contrast to previous work [33], we do not assume hard deadlines, since many sensor node applications work in more flexible regimes. A block diagram of the class of sensor nodes we consider in this work, is shown in the next figure.

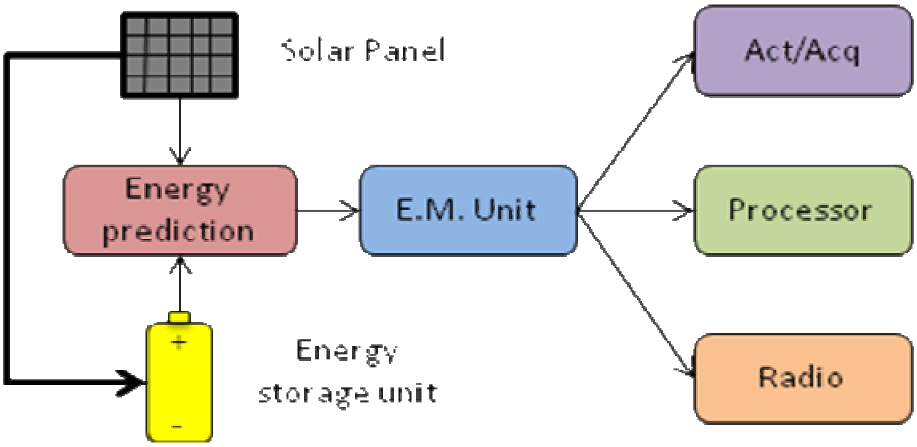
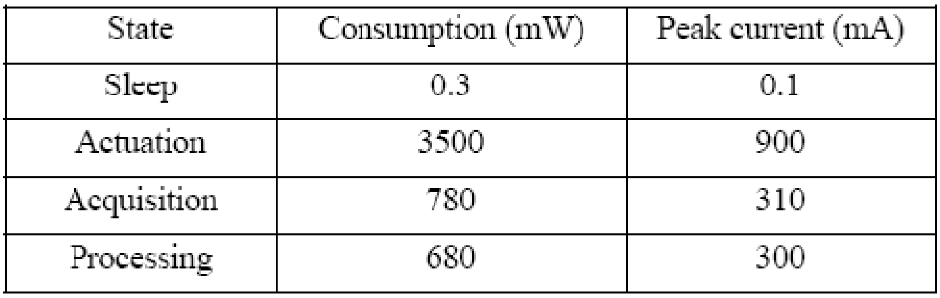


Figure 6.1 System Model

In Figure 6.1 the sensor node is powered by a solar panel and an energy storage unit. The Energy Predictor, the WCMA in our case, uses both the solar panel value and the amount of energy available in the storage unit to determine how much energy can be harvested during a future period of time. The Energy Manager Unit then takes that into account and schedules tasks of type actuation/acquisition, process, and transmit accordingly.

This particular compliance is needed in many different applications of many relevant fields in nowadays research. Considering the SHiMmer case, the platform must be able to perform an actuation when it creates the waves through the structures and it acquires it from the PZT sensors. It also must be able to perform some processing to evaluate and transform the results sending the results and receiving the command from the external devices by radio communication. The energy consumption of these three main tasks can be variable and can affect the general performance and accuracy of the system. The actuation/acquisition task, for example, must be repeated multiple times in order to have reliable results and it is obvious that the number of waves sensed is strictly proportional to accuracy but also to energy consumption. At the same time, different strategies of processing can be chosen and the radio communication can be delayed or different paths of data can be sent in order to reduce both the number of communications and their length.

We denote with E_{Ti} the energy consumed by task i and with E_{Hi} the energy harvested during the execution of task i. In most previously studied systems [10][33] the amount of energy stored is comparable to the amount needed for processing considering the Energy Neutral Operation paradigm, but in high computing and active sensing systems, the difference in energy storage capacity versus energy needs can be quite large: $E_{Ti} >> E_{Hi} \ \forall i$. This particular constraint imposes an extremely low duty cycle in order to operate in an energy neutral state, and motivates us to use the extra available energy when the energy storage is at capacity. Also the current used by tasks in similar systems is usually much higher than the average current that a small solar panel for sensor nodes can provide.



State	Consumption (mW)	Peak current (mA)
Sleep	0.3	0.1
Actuation	3500	900
Acquisition	780	310
Processing	680	300

Table 6.1 Power and peak current dissipation in SHiMmer

Table 6.1 shows the peak current of the main tasks performed by SHiMmer. We can notice that with the actuation it can reaches a peak of 900 mA that is higher than the current that a solar panel can provide if we respect the dimension constraint of the typical applications.

With this assumption we will consider that the system is not able to use the energy directly from the solar panel but rather must need some kind of energy storage unit. Clearly, the amount of energy we are able to store is limited by the capacity. Therefore, at time t, $E_A(t) \leq E_{max}$ where $E_A(t)$ represents the energy available at t in the energy storage unit, and E_{max} is the maximum energy it is able to store. Since we can only harvest energy during periods of light (assuming solar harvesting), we must have a minimum amount of energy saved that ensures the system will be alive during periods of darkness. We denote $E_{min}(T_{max})$ as the minimum amount of energy that the system needs to stay alive during a period T_{max} when energy is scarce.

Next, we assume that a sensor node needs to be able to respond to queries for data upon a trigger (event-triggering) node, so it needs to keep a minimum amount of energy to reserve. So, let E_{trig} be the amount of energy necessary for the node to execute the necessary tasks when triggered. Thus, the new minimum energy required is $E_{Tmin}(T_{max}) = E_{min}(T_{max})+E_{trig}$.

The energy manager must maintain the system energy level below E_{max} and above E_{min}. If the amount of energy in the storage unit reaches E_{max} and if more energy can be harvested, then energy should be spent by executing tasks. Otherwise the extra available energy would be wasted since the system is unable to harvest more energy.

In order to save as much energy as possible, the system must enter into a low power sleep state. Furthermore, if we can estimate how much energy is likely to be harvested over the next period of time, then we can determine when we can wake up the node to execute tasks. Our energy prediction unit uses present and past values from the solar panel, as it is explained in

Chapter 4 and 5, to compute the time necessary to recharge the energy storage unit. The prediction is then used by our energy management system in order to schedule different types of tasks so that the system maximizes its performance and energy availability.

6.3 Energy Management

The block diagram in the following figure presents the organization of the development of our energy management system.

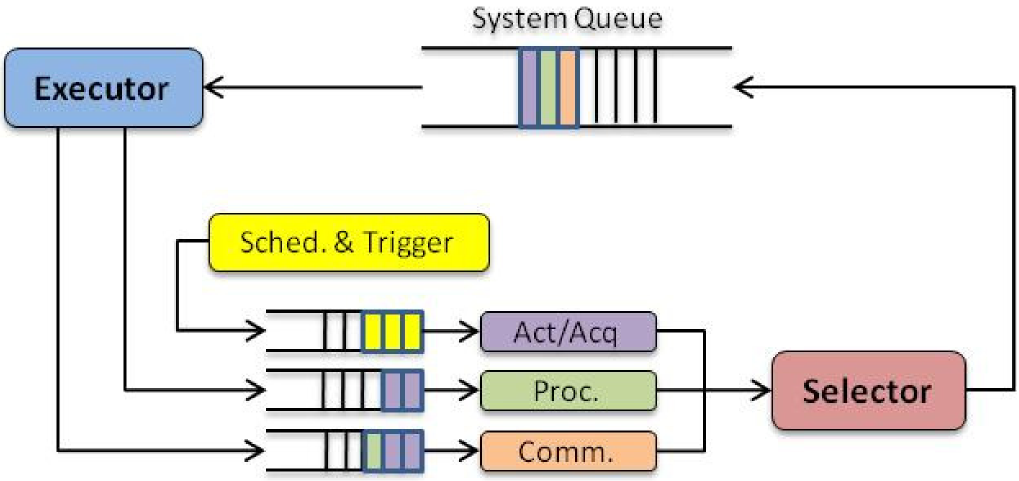


Figure 6.2 Energy management unit

6.3.1 Energy Manager Description

The system manages two subsections of the design: the energy supply and sink. The energy supply, in our particular case, consists of energy harvesting via solar cells, storage and prediction algorithms related to those described in Chapter 4. The energy sink is divided into three separated parts: actuation and sensing, processing and communication. The first task usually executed is the acquisition and sensing of data which may involve actuation, either by mobilization or by active sensing and it can be very expensive in terms of power consumption. After the acquisition, some performance processing may be necessary, for example using a DSP to compute a FFT to process the acquired data can be useful for detecting damage in structures in SHM applications. The processing part can be repeated many times to produce more accurate evaluations of the damages, but can also be possibly excluded because it can be preferable to just send the acquired values instead of processing them. Finally, the results may need to be transmitted by a wireless radio and this can involve many different evaluations about the communication path than can have better performance concerning higher energy costs. The manager has the target of adding different types of tasks to the system and to put them into their respective queues depending on the current needs. The task selector selects tasks from the heads of the three queues and puts them into the system queue. It determines the execution order by considering priority, memory availability, and the application objectives of the sensor node.

The task executor executes the task at the head of the system queue if there is energy available. If not, then it will estimate the length of time it will take to recharge the energy storage unit and enter the sleep state for the estimated time. After each task is executed, the executor determines which queue the results should enter next. For example, after executing an actuation/acquisition task, the executor can put the results into the process or transmit queue for either processing or transmitting respectively.

The energy management system we are considering faces the problem of how to spend the energy harvested in the correct way. Various tradeoffs may be selected and explored. For example, sometimes a system will decide to only gather then send data to save energy since more energy may be consumed during processing versus transmitting. On the one hand, this kind of approach may save energy, but the energy consumption during transmission may increase because the data length may be larger than expected. Another consideration that the scheduler must take into account is how to process the collected data. Sometimes it is better to process every time the system senses new data; other times processing more data together can provide more accurate results at the expense of memory usage.

In a node that performs actuation, processing and radio transmission, different patterns can be applied. Let us call actuation *A*, processing *P* and transmission *T*.

- ***Act-Process-Send***: This is the most general pattern which sequentially performs: *A P T A P T A P T…*

- ***Act-Send***: If the consumption of the radio is much lower than the processing, then the pattern: *A T A T A T A T…* may be used. This is useful when we need real-time data streaming.

- ***Multiple Act***: The scheme: *A A A A P T…* could be used to actuate multiple times before processing. This could be useful in considering many samples together, for example taking the set average.

- ***Event Triggering Sending***: In some systems the radio may not always be available. In this case the pattern: *A P A P A P…* and eventually *T* when the radio becomes available.

Taking into account these different tradeoffs and patterns, we will introduce some possible scenarios to apply the energy management system. All the scenarios consider wireless sensor nodes with high performance, active sensing, and computation powered by energy harvesting.

The first scenario we consider is an Electrocardiogram (EKG) monitoring system, using the reconfigurable embedded sensor node SunSPOT. Methods of EKG health monitoring involve having patients visit their doctors where they will then be monitored for a short period of time. However, such monitoring can not accurately assess the complete health of the patient, nor is it able to predict infrequent occurrences of abnormalities. For periodic EKG analysis, the Multiple Act profile could be used to acquire data for a certain length of time, process it by selecting an algorithm depending on the energy state of the node and finally transmit the results when a connection is available. The Act-Send profile may be used if real-time streaming data is needed.

Another important and complicated scenario that could exploit more possibilities of the energy manager, would be for Structural Health Monitoring applications. SHM, as it has been explained in Chapter 2, is the process of observing a structure over time, identifying a damage sensitive feature in the observations and performing a statistical analysis of these features to determine the health of the observed structure. The Act-Process-Send profile may be used to periodically check the health of a structure. If a SHM node is placed in an inaccessible location, (the top of a bridge where a radio link may not always be available), or the node is used for the rapid assessment of structural state after an extreme event, (earthquake), the Event Triggering Sending profile can be used. Since some SHM algorithms require high accuracy by actuating and measuring the same area many times to compute an average, the Multiple Act profile should also be used.

Now, we must determine how long tasks will wait to be executed. In the next section, we will analyze this queuing system so that we can determine the average queue time for a given task.

6.4 Queue model

In this section we will present a prioritized queue model, developed in collaboration with Joaquìn Recas Piorno, that gives the expected residence time of the tasks taking into account the energy requirements of the system.

6.4.1 Delay in priority queue systems

Due to the new low power and high process capabilities microprocessors, present embedded systems are capable to execute multiple tasks so, in order to maximize the performance of the node, a priority system can be applied. We are trying to analytically describe the mean delay systems with priority service discipline.

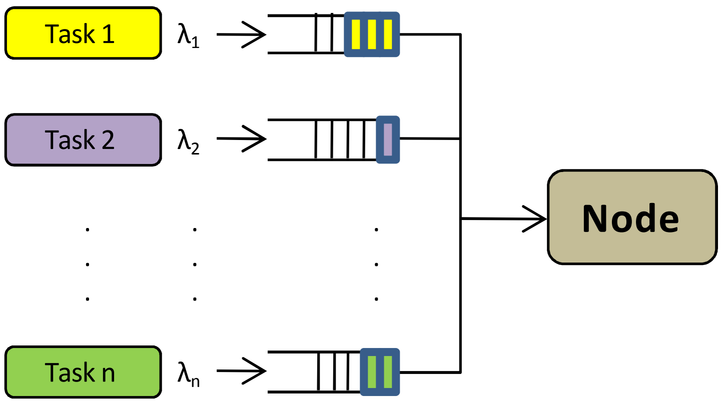


Figure 6.3 Multi task general system

The figure shows a general system organization that presents many different queues for every required task in the application.

We can model the system as a general *M/G/1* queue in which the arrival time follows an exponential random variable and the execution time follows a general random variable distribution. In our model we use separate queues for different priority tasks and non preemptive tasks. When the node becomes available, it selects a task. In the next section we will refer to a general task as T_k, to be executed from the highest non empty queue. This discipline is referred to as *head-of-line priority service* in reference [77]. In this model, the inter arrival time of a task T_k follows a Poisson process of rate λ_k and it has an expected execution time $E[\tau_k]$. We define the server utilization of T_k as:

$$\mu_k = \lambda_k E[\tau_k] \qquad (6.1).$$

In order to compute the estimated waiting time in queue for a T_k, $E[W_k]$, let us assume that T_{k-1} has bigger priority than T_k. In this case the estimated waiting time in queue for the highest priority task, T_1, can be computed as (see [77]):

$$E[W_1]=E[R^T]+E[N_{q1}]E[\tau_1]=\frac{E[R^T]}{1-\mu_1} \quad (6.2).$$

where *E[R^T]* is the residual service time of a task found in service, *E[N_{q1}]* is the expected number of tasks of type 1 found in the queue. T_2 has less priority than T_1 so, for computing the queue waiting time for T_2, we will take into account the number of elements in T_1 queue and the expected T_1 arrivals during the waiting time, so the queue waiting time will be as follows:

$$E[W_2]=E[R^T]+E[N_{q1}]E[\tau_1]+E[N_{q2}]E[\tau_2]+E[M_1]E[\tau_1] \quad (6.3).$$

where *E[N_{q2}]* is the expected number of tasks of type 2 found in the queue and *E[M_1]* is the number of expected arrivals of T_1 during the waiting time. It can be proven [77] to be equal to:

$$E[W_2]=\frac{E[R^T]}{(1-\mu_1)(1-\mu_1-\mu_2)} \quad (6.4).$$

And more generally for a *Task k*:

$$E[W_k]=\frac{E[R^T]}{(1-\mu_1-...-\mu_{k-1})(1-\mu_1-...-\mu_k)} \quad (6.5).$$

The task found in service by an arriving task can be of any type, so *E[R^T]* is the residual service time of tasks of all types:

$$E\left[R^T\right] = 1/2 \sum_{i=1}^{n} \lambda_i E\left[\tau_i^2\right] \qquad (6.6).$$

where $E[\tau_i^2]$ is the second moment of the service time of T_i.

With this model, knowing the inter arrival time of the different tasks λ_k, the expected execution time $E[\tau_k]$, and the second moment of the expected execution time $E[\tau_k^2]$, it is possible to compute the total expected waiting time, time spent in queue plus execution time, as follows:

$$E\left[T_k\right] = E\left[W_k\right] + E\left[\tau_k\right] \qquad (6.7).$$

Example 1**:** Let us study the example shown in Figure 6.4, a node with three tasks, *Sense*, *Process* and *Transmit*, with the *Sense* and *Transmit* as the highest and lowest priority task respectively.

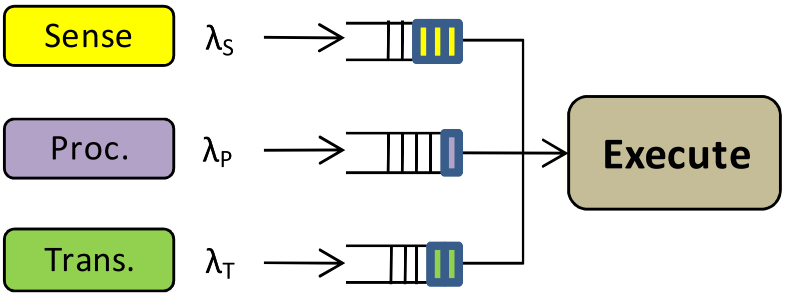


Figure 6.4 Residence time example

	λ_S	$E[\tau_A]$	P
Sense	0.1 req/s	2s	0.5W
Process	0.1 req/s	1s	1W
Transmit	0.05 req/s	5s	0.2W

Table 6.2 Priority Task Execution example Data

If we assume that the execution time follows an exponential random variable, i.e. the system can be described as a *M/M/1* queue model, the second moment of the execution time, [77], will be $E[\tau_k^2] = 2\cdot E[\tau_k]^2$ and so, the residual service time using the Table 6.2 values for the tasks, will be:

$$E\left[R^T\right] = \sum_i \lambda_i E\left[\tau_i\right]^2 = 1.75s \tag{6.8}$$

Using equation (6.5) we can compute the estimated queue time for a new *Transmit* task that arrives to the system:

$$E\left[W_T\right] = \frac{E\left[R^T\right]}{\left(1-\mu_S-\mu_P\right)\left(1-\mu_S-\mu_P-\mu_T\right)} = 5.56s \tag{6.9}$$

Finally, the mean residence time in the system for a new *Transmit* task will be the queue waiting time plus the execution time, which is 10.56s. The same study can be done for *Sense* and *Process*, obtaining an estimated waiting time of $E[W_S]$ = 2.19s and $E[W_P]$ = 3.12s and residence time of $E[T_S]$ = 4.19s and $E[T_P]$ = 4.12s. Notice that the residence time for *Sense* is slightly bigger than *Process*, this is because of the execution time that is double for *Sense* task. This kind of studies gives important system behavior characteristics, which are not obvious in some cases.

6.4.2 Delay in energy harvesting systems

In a system without energy constraints, when the node becomes available, it is always possible to execute a task. But in systems with energy constraints, even if the node is free to execute a task, it is possible that there is not enough energy available to do it, so the actual waiting time must take into account the energy availability.

Let us define $E[E_k]$ as the total estimated energy needed to execute a new task T_k that enters the system taking into account prioritization. This amount of energy will be the sum of the energy needed to execute all the tasks of higher or equal priority that are waiting in the system $E[J_k^W]$, plus all the higher priority tasks that arrive during the waiting time $E[J_k^A]$, plus the energy needed to execute the task $E[J_k]$, plus the residual energy needed to finish the task already in service at the arrival $E[R^J]$:

$$E[E_k] = E[J_k^W] + E[J_k^A] + E[J_k] + E[R^J] \qquad (6.10).$$

Assuming the same criteria used in the previous section about priority assignation, i.e. a lower task number corresponds to a higher priority, $E[J_k^W]$, $E[J_k^A]$ and $E[R^J]$can be computed as follows:

$$E\left[J_k^W\right]=\sum_{i=1}^{k} E\left[J_i\right]E[N_{qi}] \qquad (6.11).$$

$$E\left[J_k^A\right]=\sum_{i=1}^{k-1} E\left[J_i\right]E[M_i] \qquad (6.12).$$

$$E\left[R^J\right]=1/2\sum_{i=1}^{n} \lambda_i E\left[J_i^2\right]P_i \qquad (6.13).$$

But, how much energy does it correspond to? If we have enough energy, the waiting time defined in equation (6.5) can be directly computed and, applying Little's formula, we have that $E[N_{qi}]=\lambda_i E[W_i]$ and also the expected arrival tasks of higher priority are $E[M_i]= \lambda_i E[W_k]$, replacing them in (6.11) and (6.12) we obtain:

$$E\left[J_k^W\right]=\sum_{i=1}^{k} E\left[J_i\right]\lambda_i E[W_i] \qquad (6.14).$$

$$E\left[J_k^A\right]=\sum_{i=1}^{k-1} E\left[J_i\right]\lambda_i E[W_k] \qquad (6.15).$$

Now we are able to compute $E[E_k]$ only knowing the expected queue waiting time, arrival rate and expected energy consumption of T_1 to T_k.

Example 2**:** Continuing with the example described in Table 6.2. How much energy do we expect to be necessary to finish a new *Transmit* task, the lowest priority one? Using (6.11), (6.12) and the waiting time computed in the previous section, we find $E[W_S]$ = 2.19s, $E[W_P]$ = 3.12s, we obtain $E[J_T^W]$ = 0.79J and $E[J_T^A]$ = 0.94J. If we assume that the execution time follows an exponential random variable, as explained in the previous section, we obtain $E[R_J]$ = 0.44J. Finally the total energy needed will be $E[E_T]$ = 3.18J

6.4.3 Energy Neural Operation

Due to the variation of the environmental energy availability in energy harvesting systems, to ensure energy neutral operation (ENO), that is that the system is able to run perpetually, it becomes necessary to have an accurate model of the system and adapt the consumption parameters to make the mean energy consumption during time smaller or equal to the energy entering the system. The total estimated energy consumption of the system, *E[E]*, will be the sum of the power consumption of each task:

$$E[E] = \sum_{i=1}^{n} \mu_i P_i \qquad (6.16).$$

In [29] the authors introduce a model to characterize the energy consumption and availability taking into account the time variations. A general function of time *f(t)* is a $(\rho_s,\sigma_1,\sigma_2)$*-function* if the next inequality is true for all real number *T*:

$$\rho_s T - \sigma_2 \ \leq \ \int_{\tau}^{\tau+T} f(t)dt \ \leq \ \rho_s T + \sigma_1 \qquad (6.17).$$

A (ρ_c,σ) *– function* is also defined as:

$$\int_{\tau}^{\tau+T} f(t)dt \ \leq \ \rho_c T + \sigma \qquad (6.18).$$

If we denote $P_s(t)$ as the energy delivered by the harvesting unit during time, and identify it with a *(ρ_s,σ_1,σ_2) – function*, ρ_s becomes the mean energy rate available, and σ_1,σ_2 covers time variations of the energy availability. We can identify a consumer $E_c(t)$ with a *(ρ_c,σ) – function* in which the mean energy consumption rate *E[E]*, is ρ_c, and the variations are covered by σ.

In [11], the authors give a detailed explanation of this model and postulate that, assuming those models for energy harvesting and consuming, if we have an energy buffer characterized by parameters η for storage efficiency, and ρ_{leak} for leakage, the following conditions are sufficient for the system to achieve energy neutrality:

$$\begin{aligned} &\rho_c \leq \eta\rho_s - \rho_{leak} \\ &B \geq \eta\sigma_1 + \eta\sigma_2 + \sigma \\ &B_0 \geq \eta\sigma_2 + \sigma \end{aligned} \qquad (6.19).$$

where B denotes the capacity of the energy buffer and B_0 is the initial energy stored in the buffer.

Intuitively those equations say that the mean rate of consumption should be less or equal to the mean rate of storage plus the leakage loss. And also that the storage capacity of the system should be able to store energy enough to cover three special cases:

1. There is a burst of energy available that should be completely stored in the energy source, modeled by $\eta\sigma_1$.
2. There is a lack of energy available, so in the system there should be at least $\eta\sigma_2$ energy stored.

3. There is a period of high energy demanding by the consumer, modeled by σ. We can identify the energy consumption computed in (6.16) with ρ_c.

The waiting time for a task T_k, entering the system at time t, should not be affected by the harvesting unit if the energy needed to execute T_k is less than the energy present in the system when T_k arrives plus the minimum energy the system is able to store during the waiting time:

$$E[E_k] \leq E_t + \eta(\rho_s E[T_k] - \sigma_2) - \rho_{leak} E[T_k] \qquad (6.20).$$

where E_t is the energy stored in the system in t. If this inequality is true for all T_k entering the system, the mean waiting time is only dependent on the arrival rate and the mean execution time.

Let us consider the case when the necessary energy is strictly equal to the stored energy plus the harvested energy; if there were more energy available the restrictions would be less strict. Considering an initially empty system at t_0 and an arrival of T_k, the total estimated waiting time $E[T_k]$ due to other tasks, will be 0, so the energy stored in the system has to be equal to the energy needed to execute the task plus the minimum energy harvested during the execution minus the leakage, so $E[J_k] = E_t - \eta\sigma_2 + \rho_{leak} E[\tau_k]$ is true for an empty system.

If we now consider an arbitrary state at t_n when all the tasks have been added fulfilling (6.20), a new task T_k can be added if there is enough energy to execute all the tasks with the same or more priority, plus the energy necessary to execute higher priority tasks that arrive during the execution of the last task in the k queue, T_k'. If the previous task added in the system fulfills (6.20), there is enough energy to finish T_k', so we only need to add the energy necessary to execute the extra high priority tasks that arrive during the execution of T_k', that is covered by the factor $E[M_i]$

by (6.10). Equation (6.20) is in particularly useful to study the residence time of the tasks in nodes in which the execution time and energy consumption of the tasks are known.

But, how can we estimate the waiting time in systems when (6.20) is not true? In these cases, the waiting time of an incoming task will be affected by the harvesting system and the task will remain in the system until enough energy is harvested. This waiting time can be computed using (6.20), so:

$$E[T_k] = \frac{E[E_k] - E_t + \eta\sigma_2}{\eta\rho_s - \rho_{leak}} \qquad (6.21).$$

Example 3**:** Considering that we need to execute a *Transmit* task seen in the previous section, $E[E_T]$ = 3.18J, and we only have E_t = 1.5J in the system, assuming that the store efficiency is η = 1 and ρ_{leak} in null, the maximum expected waiting time will be only dependent on the lower bound of the harvest unit. Let us suppose that the power entering the system is ρ_s = 0.5W and σ_2 = 5J, then the expected residence time will be $E[E_T]$ = 13.35s.

6.5 System Queue Evolution

In this section we show how the energy management system selects and executes tasks in order to efficiently use energy. We decided to simulate the evolution of the queues inside the system using a simple Actuate-Process-

Send profile, with a task energy consumption of 25, 10 and 5 Joules respectively, based on the measurements on the SHiMmer sensor node.

In the first scenario that we considered, ten actuation tasks are added to the system everyday to perform actuation, including sensing, processing and transmission. These actuation tasks will be executed depending on energy and memory availability. After each actuation is completed, it will be sent to the processing queue and then to the transmitting queue. By adding 10 tasks to the system each day, that is a high load for the considered application, the system is usually unable to execute all tasks in the same day unless the whole day is sunny. Thus, we would be able to evaluate and correct the system behavior when the tasks remain in the system for several days.

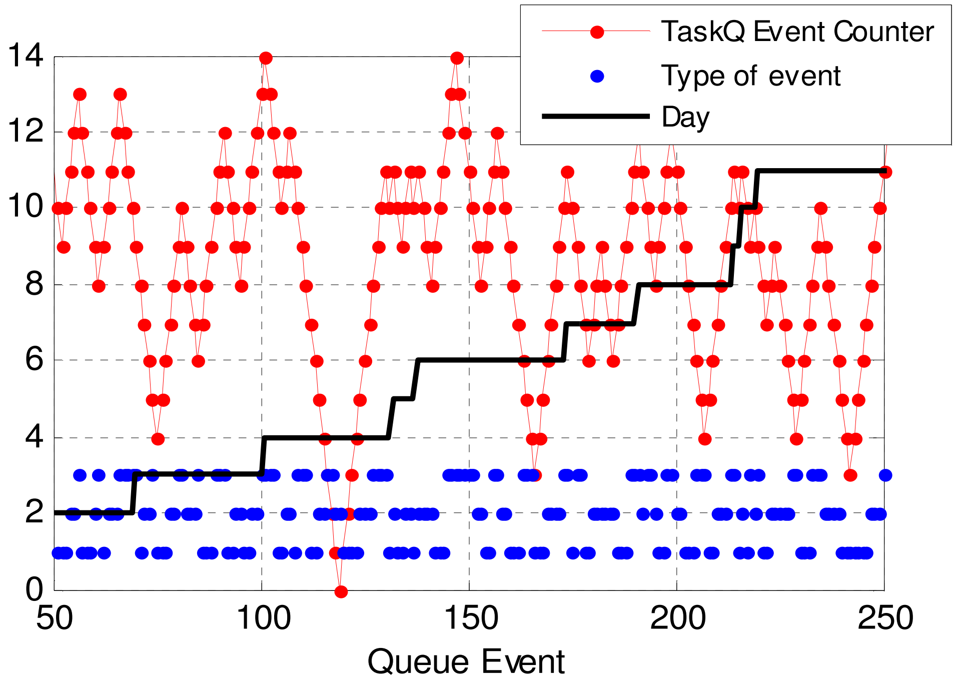


Figure 6.5 System queue evolution

The figure shows the results of the simulation of the System Queue evolution over 9 days. The red dotted line is the task queue event counter which keeps track of the number of elements in the system task queue. We consider that the tasks enter the queue at the beginning of the day, so every day the value increases of ten units. The blue dots are the type of tasks that enters or exits the system queue where type 1 is actuation, type 2 is processing, and type 3 is transmitting. The black continue line represents the days' evolution, when it increases it means that the day has changed. We can see in Figure 6.5 how the number and type of tasks in the system queue change during the day due to solar energy and memory availability. On most days there is not enough solar energy available to execute all tasks in the System Queue. However, on day 4, all the tasks are executed (the system queue becomes empty) because enough energy has been harvested. On the contrary, when the solar energy is scarce, few tasks can be executed and that produces an increasing of the number of tasks in queue. For example on day 8, that was cloudy, only one task is executed because of solar energy scarcity. We also took into account the memory usage. We fixed a maximum amount of memory available and we considered that sensing data on SHiMmer requires 8000 data points. It is easy to understand that if the manager executed only sensing tasks, the memory would soon be full. To reduce memory usage we must consider that process and transmit tasks must be executed before executing other sensing tasks when the memory is going to be full.

The simulation results show that during the 45 days the total amount of energy that can be used by the task is 11650J.

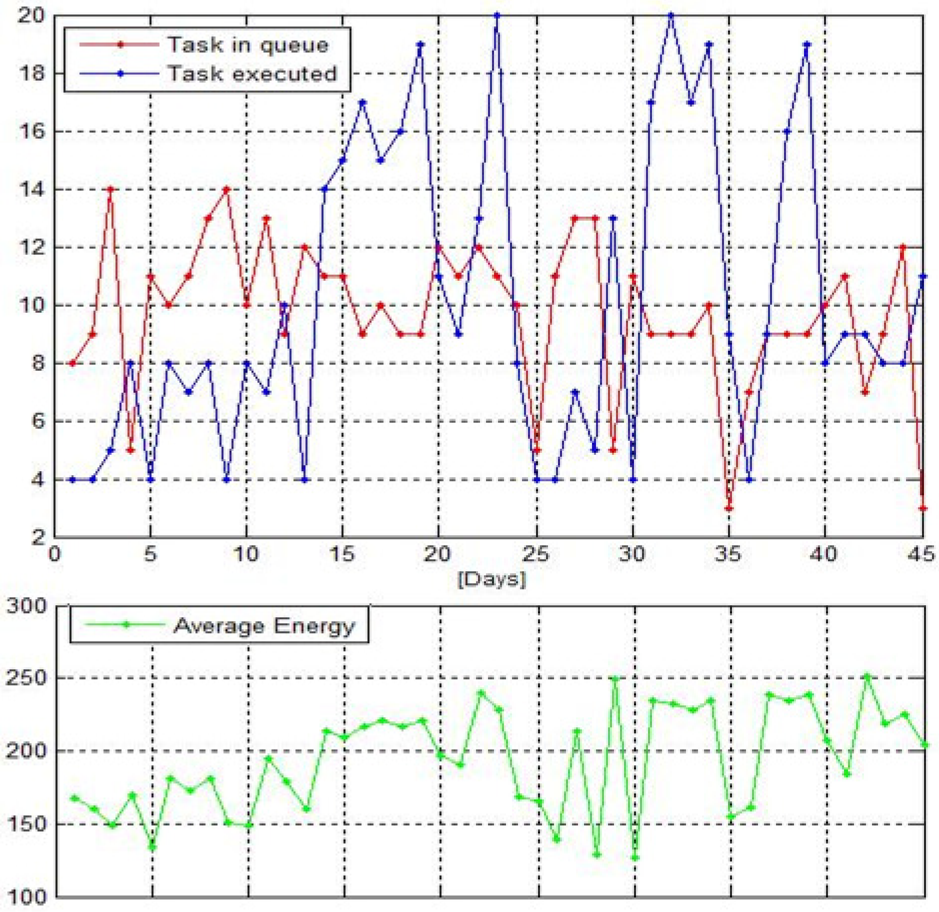


Figure 6.6 Queue evolution related to the input energy during 45 days

Figure 6.6 shows the evolution of the system queue during the whole period of 45 days considered in the simulation. The lower line represents the average energy during the day and we can notice the strict relation between the input energy and the number of complete tasks ended during the day. Even if on some days the system is not able to compute a high number of tasks, the system queue never explodes and maintains a number of total processes in the queue always lower than 14.

We also compared the evolution of the system queue produced by the developed algorithm to the one produced with the binary search. The following figure shows the system queue using the same profile as before except that the binary search recharge estimator is used to estimate how long it will take for the supercapacitors to recharge.

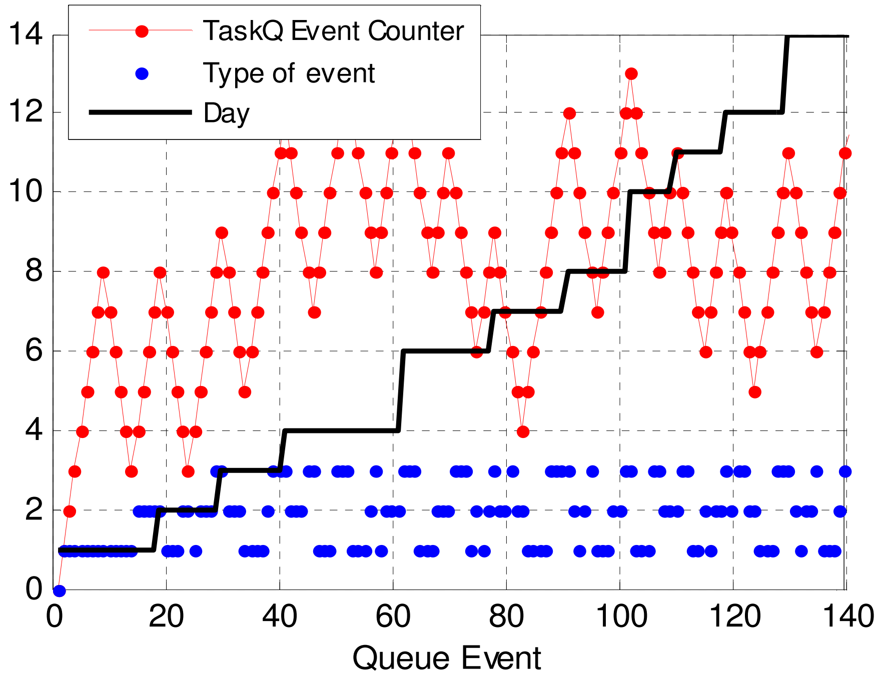


Figure 6.7 System queue evolution with binary search

We can see that in the same period of time as before, the number of executed tasks is lower. Also, there are several days (5, 9, 13) where the system is unable to execute any tasks. This shows that if an imprecise algorithm is used to estimate the recharge time, then a non-optimal number of tasks will be executed. The total amount of energy used during the 45 days is 5225J, which is less than half of the amount of energy used when our estimator was utilized. This shows the accuracy of our predictor and estimator.

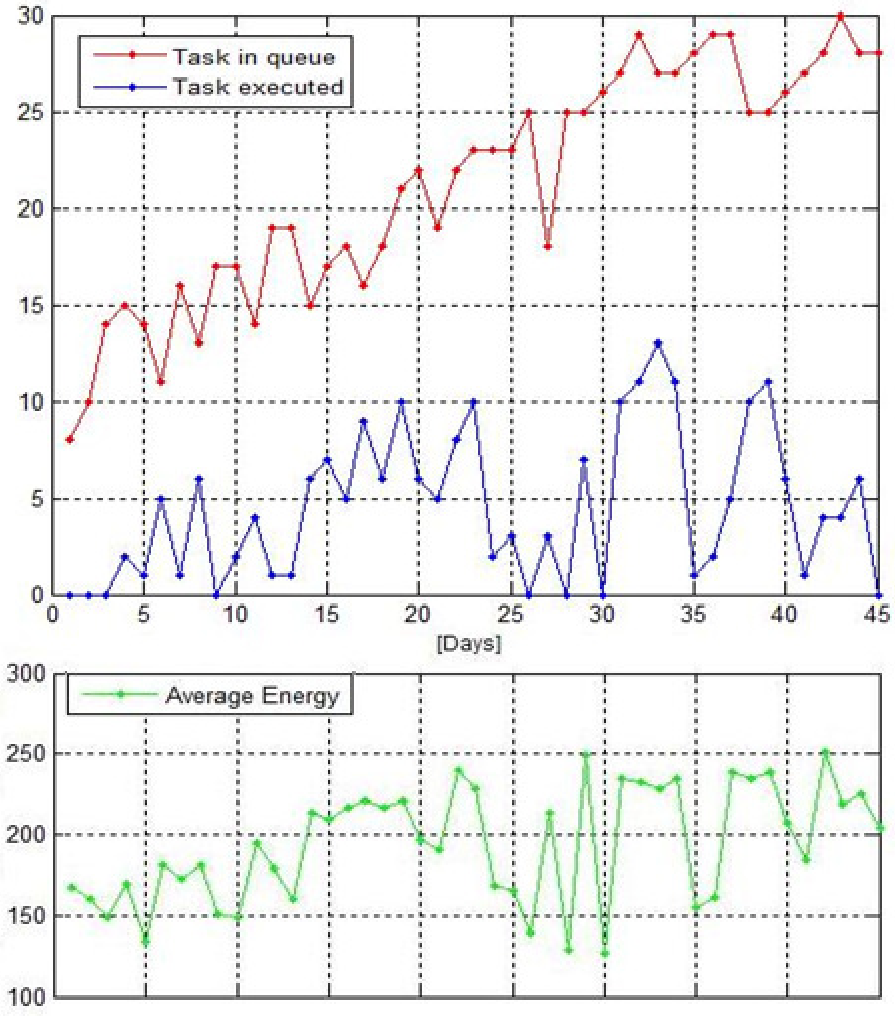


Figure 6.8 System queue evolution with binary search during 45 days

Figure 6.8 shows the evolution of the queue during the whole period of 45 days. In this case the system is not able to optimize the use of energy and runs a lower number of tasks. The number of processes in the system queue increases during the whole period and this will cause it to grow to infinite. At the same time the number of processes completed on each day never exceeds the number of 13, while in Figure 6.6 we can notice that it reaches the value of 20.

6.6 Queue Waiting Times

Now we will observe how long a complete process composed by the pattern A, P, T tasks takes to execute using the energy manager.

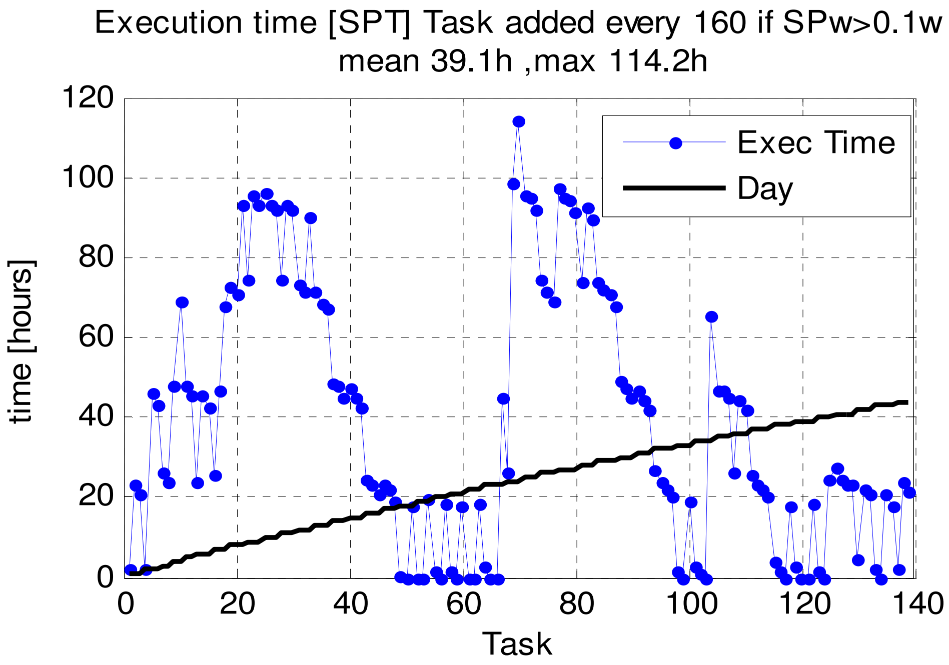


Figure 6.9 Low arrival rate

In Figure 6.9 an acutation task arrives every 160 minutes and the tasks consume the same amount of energy as specified in the previous section. All the 45 days of data are used to run the simulation. Notice that the queue time increases between days 1 and 12 because there is insufficient solar energy. However, as more solar energy becomes available, the delay decreases as seen on days 18 to 24.

To compare these results to the theory discussed before, let us determine that at a certain time, the state of the system is as follows: 5 actuation tasks, 2 process tasks, and 2 transmit tasks in their respective queues. The total energy required to execute all these tasks to completion, using the measurements given, is 240J. At this time, a new actuation task enters the system. The amount of available energy for execution, if the supercapacitor is full, is 135 Joules, and T_H is 120 minutes if the day is sunny. If the day is cloudy, then T_H may even be greater than 24 hours. Given a sunny day and the supercapacitor empty, then it will take a mean of 120 minutes to recharge and begin execution. Calculating $k = 1$ as the number of recharge cycles, the total time before the new task is fully completed in the A, P, T order would be 240 minutes. However, during this time, another new task would arrive due to the average arrival rate of 3.2 tasks per day. Since priority is considered, 25 more Joules would be consumed before our task is fully complete. This would mean that another recharge cycle of 120 minutes is necessary. Taking all things into consideration, the waiting time for our new task to be fully completed would be 6 hours if in sunny conditions. We can see that this time would dramatically increase during cloudy days since the mean time is 5 hours.

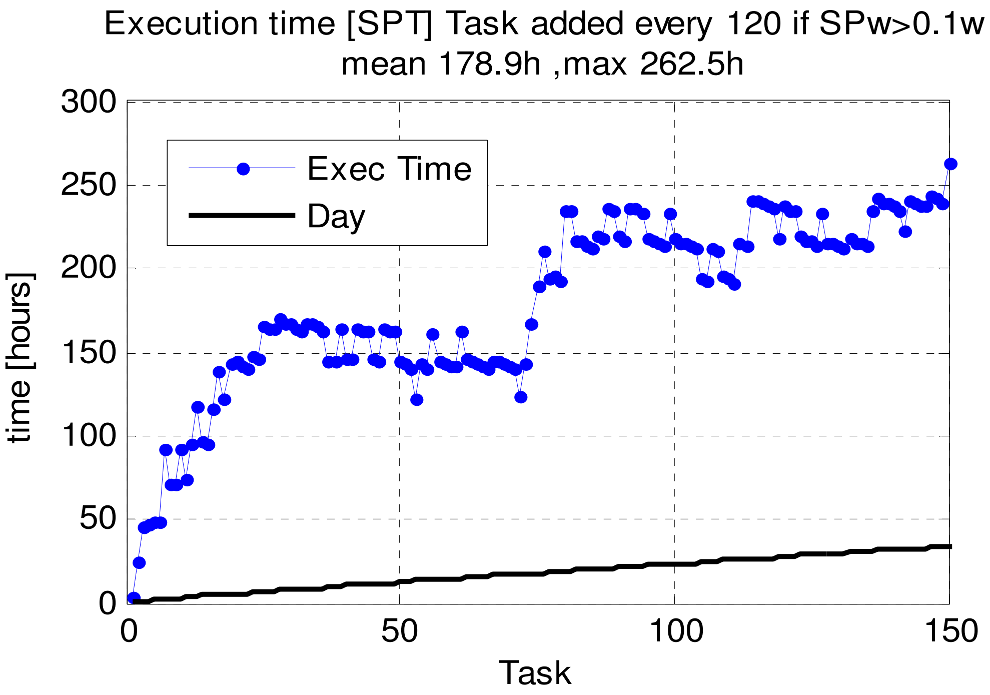


Figure 6.10 High arrival rate

Figure 6.10 represents the same situation with an increased arrival rate of 120 min per actuation task. Since there are cloudy days and the arrival rate is higher than before, the delay increases significantly to the point where the system is unable to handle so many tasks. The delay will eventually go to infinite. If we compare the mean queue time of 39 hours in the previous figure, to the mean queue time of 178 hours in this figure, we can see that by increasing the arrival rate just by 40 minutes, the queue time will eventually go to infinite. Thus, we can clearly see that the average queue time is solely based on the time it takes to recharge the supercapacitors and that task execution time has now become insignificant.

6.7 Energy profiles

We will now demonstrate how the energy manager dynamically utilizes two different energy profiles. The first is a high energy profile that actuates several times, runs a complex signal analysis algorithm, and then transmits both data and results. This will use the original SHiMmer task energy values defined in Section 6.4.1 and will be run in times when energy is abundant. The second is a low energy profile that actuates once, executes a simple processing algorithm, and only transmits the result. The energy consumption for this is 10J, 1J, and 0.5J respectively and will be used when energy is scarce.

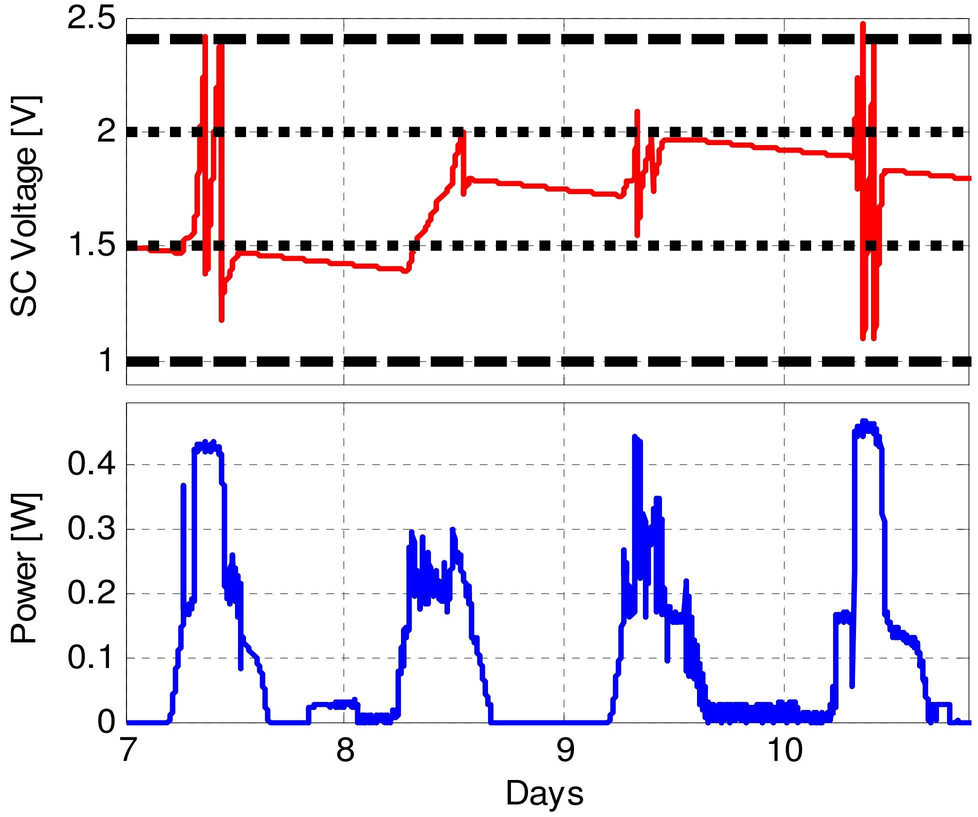


Figure 6.11 Simulation with two energy profiles

Figure 6.11 shows a period of four days, the first and the last being sunny. We can notice that the system uses the second profile during cloudy days and the first during sunny days. The red line is the supercapacitor voltage, and the blue is the amount of power that the solar panel provides throughout the day. The long, broken lines are the upper and lower bounds of the supercapacitor voltage during cloudy conditions and the short broken lines are the bounds during sunny conditions. Also the working range of the supercapacitor is, in fact, adapted to energy availability switching to a 1.5V-2V range during energy scarcity. This occurs in order to be able to execute some tasks anyway and also to save more energy for “surviving” during the night or future expected cloudy conditions.

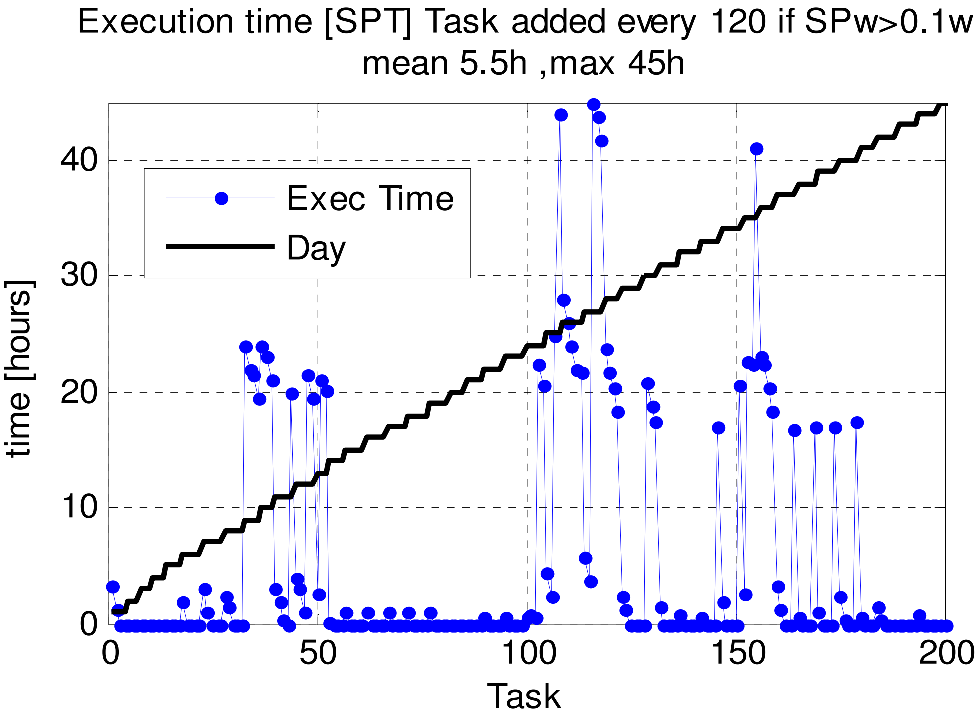


Figure 6.12 High arrival rate with two profiles

This figure shows how long tasks wait in the queue while the energy manager switches between high and low energy profiles. We can see that the mean queue time is only 5.5 hours because the system does not execute energy-expensive tasks when solar energy is scarce. Longer waiting times occur if solar energy is especially scarce for several days at a time, as seen by the maximum queue time of 45 hours. Therefore, by utilizing different energy profiles in our energy manager, we can minimize queue time for energy-expensive tasks.

Conclusions

The development of this work was born from the analysis of a wireless platform powered by solar harvested energy for structural health monitoring. The first part of the work included software evolution to processors code and the graphic user interface. After the analysis of the power management strategies adopted in SHiMmer platform we have determined the formulation of many algorithms and strategies in order to optimize the solar energy usage in every actuation-based system with similar characteristics.

First of all we have focused our attention in deleting the dependence of the system on radio triggering and provide it to the capability of performing tasks every time the energy storage unit reaches the maximum capacity. This permits to utilize a greater quantity of energy that would be wasted using the previous architecture if the UAV was not triggering the system for long time.

In order to obtain the requested target an energy prediction algorithm has been developed in order to provide a prediction of the future value of solar irradiation. The Weather Conditioned Moving Average takes into account the seasonal variation of the hours of dawn and sunset, but also of the weather conditions compared to the previous days' ones. Because of the strict energy and memory constraints of the reference systems, the prediction computation has been limited to simple calculation and the memory usage is just few hundreds of bytes. The algorithm has been compared to different algorithms with similar targets and it has produced a much lower error using the same computational costs.

Once obtained the energy prediction we have created a model in order to connect the input energy level to and estimation of the recharging rate of the energy storage unit. We developed an *ad hoc* methodology tested on the SHiMmer platform. The Recharge Estimator permits to create equations that calculate a sleeping time from the current state of charge of the supercapacitor and the future energy prediction. During this time the system remains in a low power state and, once awake, is able to perform the required tasks because the energy storage unit has probably reached the state of charge needed.

After the development of these algorithms, we evaluated the scheduling of the different tasks present in the reference system such as Actuation/Acquisition, Processing and Communication. After a theoretical analysis of the queues a prioritization strategy has been developed in order to support the scheduling of the 3 tasks. We have also defined 2 different energy profiles which provide a various precision level of the results, but also a various energy consumption. The current system energy, together with the future prediction, permit the system to switch from a default higher energy profile to a lower profile in case of energy scarcity.

In conclusion, we have developed different techniques that, implemented in the same system, provide an optimization of the energy consumption permitting the system to perform a bigger number of tasks during the same time slot. Simulations and tests on the platform have shown that the energy wasted is minimized also obtaining benefits on performances and precision of results. The different algorithms have been compared to algorithms developed with the same target and they have determined a much higher precision and a lower error.

Conclusioni

Questo lavoro è stato svolto a partire dallo studio approfondito di una piattaforma wireless alimentata ad energia solare per applicazioni di Structural Health Monitoring. La prima parte del lavoro ha contribuito ha portare dei miglioramenti al software utilizzato dai processori ed all'interfaccia del sistema con l'utente. In seguito l'analisi dei problemi di gestione dell'energia raccolta sulla piattaforma SHiMmer ha portato ad una serie di valutazioni che hanno determinato la formulazione di diversi algoritmi e strategie per ottimizzare l'utilizzo dell'energia solare in tutti i sistemi attuativi che presentano caratteristiche simili.

In primo luogo si è cercato di cancellare il legame diretto tra la piattaforma ed un agente esterno tramite radio triggering e si è data la possibilità al sistema di svolgere i compiti ogniqualvolta l'energy storage unit avesse raggiunto la sua capacità. In questo modo si è data la possibilità al sistema di utilizzare una grande quantità di energia che, con l'architettura precedente, sarebbe stata sprecata in assenza di agente esterno.

Per ottenere il risultato richiesto è stato sviluppato un algoritmo di predizione in grado di fornire una previsione dell'irradiazione solare in un futuro prossimo. Il Weather Conditioned Moving Average tiene conto della variazione dell'orario stagionale di alba e tramonto, ma anche delle condizioni atmosferiche comparate con quelle dei giorni precedenti. Dati gli alti vincoli energetici e di memoria dei sistemi di riferimento, i calcoli del predittore sono stati limitati a calcoli di alta semplicità e lo spazio in memoria occupato si limita a qualche centinaio di byte. L'algoritmo è stato confrontato con diversi algoritmi precedentemente utilizzati allo stesso scopo, fornendo un errore decisamente più limitato a parità di costo computazionale.

Una volta ottenuta una previsione dell'energia solare entrante nel sistema è stato creato un modello per poter collegare questo livello di energia ad una stima della velocità di ricarica che essa può fornire all'energy storage unit. È stato sviluppato un metodo di realizzazione ad hoc testato poi sulla piattaforma SHiMmer. Il Recharge Estimator consente di creare delle equazioni in grado di fornire un tempo di ricarica ottenendo come input lo stato di carica attuale e la predizione dell'energia solare futura. Nel corso del tempo ottenuto, il sistema è quindi in grado di rimanere in uno stato di basso profilo energetico e, una volta risvegliato, è in grado di compiere i task richiesti dal momento che l'energy storage unit dovrebbe aver raggiunto lo stato di carica necessario.

In seguito è stato valutato l'ordinamento dei diversi task presenti nei sistemi di riferimento: Attuazione, Elaborazione e Comunicazione. A seguito di un'attenta analisi delle code è stata sviluppata una strategia di prioritizzazione variabile dei 3 compiti del sistema in grado di dare un supporto allo scheduling di essi. Sono anche stati creati 2 differenti profili energetici che presentano un diverso livello di precisione del risultato, ma anche di costo energetico. L'energia attuale del sistema in combinazione con la previsione futura permettono al sistema di migrare dal profilo alto (di default) al profilo basso in caso di scarsità energetica.

In conclusione sono state sviluppate diverse tecniche che, implementate nello stesso sistema, hanno fornito un'ottimizzazione generale dell'utilizzo dell'energia consentendo al sistema stesso di svolgere una maggiore quantità di task nella stessa unità di tempo. Simulazioni e test sulla piattaforma hanno dimostrato come l'energia sprecata venga ridotta al minimo, migliorando l'efficienza e la precisione dei risultati. I diversi algoritmi sono stati comparati con algoritmi utilizzati per scopi simili ed hanno determinato l’ottenimento di un errore notevolmente inferiore.

References

[1]. J. Recas Piorno, C. Bergonzini, D. Atienza, T. Simunic Rosing, "Prediction and management in energy harvested wireless sensor nodes". Wireless VITAE 2009.

[2]. D. Musiani et al. "Active Sensing Platform for Wireless Structural Health Monitoring". IPSN 2007.

[3]. Mica Motes, Crossbow, http://www.xbow.com/Home/wHomePage.aspx

[4]. V. Raghunathan et al, "Energy-Aware Wireless Microsensor Networks," IEEE Signal Proc. Mar 2002.

[5]. R. Pon et al. "Networked infomechanical systems: a mobile embedded networked sensor platform". IPSN 2005.

[6]. Gabriel T. Sibley et al. "Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Sensor Networks". ICRA 2002.

[7]. V. Raghunathan et al. "Design considerations for solar energy harvesting wireless embedded systems". IPSN 2005.

[8]. D. R. Cox, "Prediction by exponentially weighted moving averages and related methods". Journal of the Royal Statistical Society. Series B (Methodological) 23, 2, 414–422 1961.

[9]. J. Stuart Hunter. "The Exponentially Weighted Moving Average" Quality Technology, Vol. 18, No. 4, pp. 203–207, 1986.

[10]. M. Christopher et al. "Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks". SECON 2007.

[11]. Aman Kansal et al. "Power Management in Energy Harvesting Sensor Networks". TECS 2007.

[12]. Records of Solar Panels, http://www.powerfilmsolar.com/index.htm

[13]. H. Suehrcke et al. “A performance prediction method for solar energy systems”. Solar Energy ISSN 0038-092X, 1992 vol. 48.

[14]. R. Iqdour et al. “Prediction of daily global solar radiation using fuzzy systems”. International Journal of Sustainable Energy 2007.

[15]. J. Hsu, S. Zahedi, A. Kansal, and M. B. Srivastava. “Adaptive duty cycling for energy harvesting systems”. Networked and Embedded Systems Laboratory, UCLA, April 2006.

[16]. Vijay Raghunathan, Papleologos Spanos, Mani B. Srivastava. “Adaptive Power-Fidelity in Energy-Aware Wireless Embedded Systems”. RTSS '01.

[17]. Savvides, A. Srivastava, M.B. “A Distributed Computation Platform for Wireless Embedded Sensing”, Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '02).

[18]. R. Min et al., “An architecture for a power aware distributed microsensor node”, in: Proceedings of the IEEE Workshop on signal processing systems (SIPS '00).

[19]. D. Brunelli, S. Raggini, L. Benini, C. Moser and L. Thiele, “An efficient solar energy harvester for wireless sensor nodes.”, International Symposium on Low Power Electronics and Design (ISLPED), 2007.

[20]. Aman Kansal, Mani B. Srivastava, “An environmental energy harvesting framework for sensor networks”, Int. Symp. Low power electronics & design, 2003, Seoul, Korea

[21]. Igino Folcarelli, Alex Susu, Ties Kluter, Giovanni De Micheli, Andrea Acquaviva, “An opportunistic reconfiguration strategy for environmentally powered devices”, Computing frontiers, 2006, Italy.

[22]. Luca Benini , Alessandro Bogliolo , Giovanni De Micheli, “A survey of design techniques for systemlevel dynamic power management”, Very Large Scale Integration (VLSI) Systems, 2000.

[23]. Chou, P.H., “Challenges on Low-Power Platform Design for Real-World Wireless Sensing Applications”, VLSI Design, Automation and Test, 2006.

[24]. V. Raghunathan and P. H. Chou. “Design and power management of energy harvesting embedded systems”. ISLPED '06: Low power electronics and design, New York, USA, 2006.

[25]. Chulsung Park Qiang Xie Chou, P.H. Shinozuka, M. “DuraNode: Wireless Networked Sensor for Structural Health Monitoring” Sensors,2005.

[26]. V. Raghunathan, C. Schurgers, S. Park and M.Srivastava, “Energy-Aware Wireless Sensor Networks,” IEEE Signal Proc. 2002.

[27]. C. Rusu, R. Melhem, and D. Mosse. “Multi-version scheduling in rechargeable energy-aware real-time systems”. EuroMicro Conference on Real-Time Systems, 2003.

[28]. R. Pon, M. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. Kaiser, M. Srivastava, G. Sukhatme, D. Estrin: “Networked infomechanical systems: a mobile embedded networked sensor platform”. Information Processing in Sensor Networks, 2005.

[29]. Aman Kansal , Dunny Potter , Mani B. Srivastava, “Performance aware tasking for environmentally powered sensor networks”, Measurement and modeling of computer systems, NY, USA, 2004.

[30]. Xiaofan Jiang , Joseph Polastre , David Culler, “Perpetual environmentally powered sensor networks”, Information processing in sensor networks, Los Angeles, California, 2005.

[31]. Gabriel T. Sibley, Mohammad H. Rahimi, and Gaurav S.Sukhatme. “Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Sensor Networks”.Robotics and Autom.(ICRA), Washington D.C.2002.

[32]. M. Rahimi, H. Shah, G. S. Sukhatme, J. Heidemann, and D. Estrin. “Studying the feasibility of energy harvesting in a mobile sensor network”. Robotics and Automation, 2003.

[33]. Clemens Moser, Davide Brunelli, Lothar Thiele, Luca Benini "Lazy Scheduling for Energy Harvesting Sensor Nodes" Distributed and Parallel Embedded Systems DIPES 2006, Braga, Portugal, 2006.

[34]. J. Stuart Hunter. “The Exponentially Weighted Moving Average” J Quality Technology, 1986.

[35]. S. Singh and C.S. Raghavendra, “PAMAS—Power Aware Multi-Access protocol with Signaling for ad hoc networks”, ACM Comput. Commun. Rev.28, 1998.

[36]. L. Kleinrock, Queueing Systems Vol. II: Computer Applications, Wiley-Interscience, N.Y. 1976.

[37]. V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. “Design considerations for solar energy harvesting wireless embedded systems”. Information Processing in Sensor Networks, 2005.

[38]. ZigBee Radio, Digi, http://www.digi.com/products/wireless/zigbee-mesh/

[39]. P. Alleyne, D.N.; Cawley. “The interaction of lamb waves with defects”. Ultrasonics, Ferroelectrics and Frequency Control, 1992.

[40]. A. Anandarajah, K. Moore, A. Terzis, and I.-J. Wang. “Sensor networks for landslide detection”. SenSys ’05: Embedded networked sensor systems, New York, USA, 2005.

[41]. ATMega128. http://www.atmel.com.

[42]. D. Backman, E. Flynn, R. Swartz, R. Hundhausen, and G. Park. “Active piezoelectric sensing for damage identification in honeycomb composite panels”. International Modal Analysis Conference, 2006.

[43]. CC1100. http://www.chipcon.com.

[44]. K. Chintalapudi, J. Paek, O. Gnawali, T. S. Fu, K. Dantu, J. Caffrey, R. Govindan, E. Johnson, and S. Masri. “Structural damage detection and localization using netshm”. In IPSN ’06: Processing in sensor networks, NY, USA, 2006.

[45]. P. Enjeti, J. Howze, and L. Palma. “An approach to improve battery run-time in mobile applications with supercapacitors”. Annual Power Electronics Specialists Conference, 2003.

[46]. FIRE. Fire project: Smokenet. http://fire.me.berkeley.edu/smokeNet.html.

[47]. D. Giraldo, J. Caicedo, and S. Dyke. “Experimental phase of the structural health monitoring benchmark problem”. EM ’03: ASCE Engineering Mechanics Conference, 2003.

[48]. L. Gu and J. A. Stankovic. “Radio-triggered wake-up for wireless sensor networks”. Real-Time Syst, 2005.

[49]. N. Guo and P. Cawley. “The interaction of Lamb waves with delaminations in composite laminates” . Acoustical Society of America Journal, 1993.

[50]. B. Hailu, G. Hayward, A. Gachagan, A. McNab, and R. Farlow. “Comparison of different piezoelectric materials for the design of embedded transducers for structural health monitoring applications”. Ultrasonics Symposium, 2000.

[51]. J.-H. Huang, S. Amjad, and S. Mishra. “Cenwits: a sensor-based loosely coupled search and rescue system using witnesses”. SenSys ’05 Embedded networked sensor systems, NY, USA, 2005.

[52]. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. “Directed diffusion for wireless sensor networking”. IEEE/ACM Trans. Netw. 2003.

[53]. X. Jiang, J. Polastre, and D. Culler. “Perpetual environmentally powered sensor networks”. IPSN ’05: Information processing in sensor networks, 2005.

[54]. S. Legendre, D. Massicotte, J. Goyette, and T. Bose. "Wavelet-transform-based method of analysis for lamb-wave ultrasonic node signals". Instrumentation and Measurement, 2000.

[55]. E. Lehfeldt and P. Holler. "Lamb waves and lamination detection". Ultrasonics, 1967.

[56]. P. Levis, D. Gay, and D. Culler. "Active sensor networks". Network Systems Design and Implementation (NSDI), 2005.

[57]. J. Lynch, K. Law, A. Kiremidjian, T. Kenny, E. Carryer, and A. Partridge. "The design of a wireless sensing unit for structural health monitoring". International Workshop on Structural Health Monitoring, Stanford, CA, 2001.

[58]. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. "Tag: a tiny aggregation service for ad-hoc sensor networks". SIGOPS Oper. Syst. Rev., 2002.

[59]. S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. "Tinydb: an acquisitional query processing system for sensor networks". ACM Trans. Database Syst., 2005.

[60]. D. L. Mascarenas, M. D. Todd, G. Park, and C. R. Farrar. "A miniaturized electromechanical impedance-based node for the wireless interrogation of structural health". SPIE – Health Monitoring and Smart Nondestructive Evaluation of Structural and Biological Systems, 2006.

[61]. D. Montalvao, N. M. Maia, and A. M. Ribeiro. "A Review of Vibration-based Structural Health Monitoring with Special Emphasis on Composite Materials". The Shock and Vibration Digest, 2006.

[62]. S. Pakzad, S. Kim, G. Fenves, S. Glaser, D. Culler, and J. Demmel. "Multi-purpose wireless accelerometers for civil infrastructure monitoring". International Workshop on Structural Health Monitoring, 2005.

[63]. C. Park and P. H. Chou. Ambimax: "Efficient, autonomous energy harvesting system for multiple-supply wireless sensor nodes".

SECON ’06: Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks, 2006.

[64]. V. Raghunathan and P. H. Chou. “Design and power management of energy harvesting embedded systems”. ISLPED ’06: Low power electronics and design, 2006.

[65]. V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. “Design considerations for solar energy harvesting wireless embedded systems”. IPSN ’05: Information processing in sensor networks, 2005.

[66]. F. Simjee and P. H. Chou. “Everlast: long-life, supercapacitor-operated wireless sensor node”. ISLPED ’06: Low power electronics and design, New York, NY, USA, 2006.

[67]. J. Simmers, Garnett E., J. R. Hodgkins, D. D. Mascarenas, G. Park, and H. Sohn. “Improved piezoelectric self-sensing actuation”. Journal of Intelligent Material Systems and Structures, 2004.

[68]. T. Smith, J. Mars, and G. Turner. “Using supercapacitors to improve battery performance”. Power Electronics Specialists Conference, 2002.

[69]. K. S.S., S. Spearing, and C. Soutis. “Damage detection in composite materials using lamb wave methods”. American Society for Composites, 2001.

[70]. A. B. Thien. “Pipeline structural health monitoring using micro-fiber composite active sensors”. 2006.

[71]. TMS320R2811. http://www.ti.com.

[72]. Y. Wang, J. P. Lynch, and K. H. Law. “Design of a lowpower wireless structural monitoring system for collaborative computational algorithms”. Health Monitoring and Smart Nondestructive Evaluation of Structural and Biological Systems SPIE, 2005.

[73]. N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. “A wireless sensor network for structural monitoring”. In SenSys ’04: Embedded networked sensor systems, 2004.

[74]. Y. Zou, L. Tong, and G. Steven. “Vibration-based model-dependant damage (Delamination) indetification and health monitoring for composite structures” A review. Journal of Sound and Vibration, 2000.

[75]. M. J. Ren, J.A. Wright. “Adaptive diurnal prediction of ambient dry-bulb temperature and solar radiation”, HVAC&R Research 2002.

[76]. C. Moser, L. Thiele, D. Brunelli, L. Benini. “Adaptive power management in Energy harvesting systems”. DATE ’07.

[77]. Alberto Leon-Garcia, “Probability and Random Process for Electrical Engineering. Second Edition” Addison-Wesley Publishing Company 1994.

Ringraziamenti/*Acknowledgements*

Innanzi tutto non posso fare altro che ringraziare i miei genitori che mi hanno supportato ma anche sopportato in questo lungo periodo di studi dal primo esame fino alla revisione di questa stessa tesi rendendo poi possibile la mia esperienza all'estero che ha dato il via a questo lavoro. Ringrazio anche mio fratello Andrea con cui ho condiviso non solo la camera ma altre mille esperienze universitarie in questo lungo periodo solo un anno l'ho abbandonato dandogli l'opportunità di occupare la singola prima del mio ritorno. Ringrazio anche gli zii e i miei migliori cugini Lucia, Rita e Mauro.

Non posso non ringraziare il professor Luca Benini, che ancora una volta mi ha appoggiato in un lungo lavoro di tesi e mi ha dato l'opportunità di vivere un'esperienza unica di studio all'estero. Con lui ringrazio Davide Brunelli che, nonostante fosse sempre impegnatissimo, mi ha ascoltato ed aiutato nella redazione di questa tesi ed anche Michele che mi ha aiutato a capire dei modelli che hanno portato a ottimi risultati. Un ringraziamento anche ai Professori Salmon e Chiari che mi hanno permesso di effettuare la valutazione pre-tesi del mio lavoro. *I want to thank Professor Tajana Simunic Rosing. Without her I could never reach such a great staff to work with. I must thank Joaquin and Ben, together with them I spent hours, days and nights having many trouble with our work but finally we realize a great work together. I can't miss Edoardo and Shervin for great foosball games during the breaks, but also all the other guys of the professor staff Ayse and Gaurav from the beginning, Jamie, Junjee and Priti later helped* *me and followed my works. With David and Eric I shared the laboratories, where many things happened, looking them working at their great helicopter.*

Come era altamente prevedibile, anche questa volta, mi sono trovato all'ultimo a scrivere questi ringraziamenti. Sono tantissime le persone che hanno fatto parte di questa mia carriera universitaria che si sta concludendo e sono sicuro che mi dimenticherò di ringraziare tanta gente non perché non siano stati importanti per me ma per via del mio stato confusionale di questo momento. Ringrazio Gabri, in primis perché senza di lui non saprei da dove partire a scrivere questi ringraziamenti, con i suoi sott'occhio il compito è decisamente più semplice, grazie anche alle sue reti neurali. Insieme con lui devo ringraziare il Pol. Con loro due ho trascorso centinaia di ore a studiare… o meglio centinaia di ore a fare cavolate tra una studiata e l'altra, ancora più ore in giro per Bologna, con loro è nata la Globo, ho condiviso l'esperienza Rosso ed una serie infinita di ricordi che non riesco neanche ad elencare; non ultimo il Pol mi è stato anche utile per monitorare la presenza di Davide e Luca in fase di tesi. Ringrazio Albe, tra tutti quelli con cui ho studiato è quello che conosco da più tempo e con cui ho i più grandi ricordi da condividere. Ringrazio Ricx, per questa laurea ma anche per la precedente tesi dove me lo ero dimenticato. Lo ringrazio per tutto il tempo che mi ha fatto perdere con i suoi giochini su internet e per tutte le ora di cazzeggio tra facoltà e campi da squash. Ringrazio Ale, anche lui fedele compagno di squash e di studio indispensabile per il suo supporto in uno degli ultimi grandi ostacoli di questa mia carriera universitaria. Con Axl, Luca e Azzurra si è conclusa la prima tappa della mia vita universitaria ma restano per me grandi amici con cui condividere grandi divertimenti… e ora anche un pecorino. Ringrazio anche Acqua ed Ema con cui invece concluderò questa volta e ancora non so cosa ci potrà succedere e con Ema ringrazio tutti gli altri siculi che stanno invadendo terre Irlandesi (Mario e Fulvio). Ringrazio

Tara per la sua amicizia e per la fantastica vacanza berlinese. Ringrazio Daniele sempre presente nei momenti importanti, così come Martina immancabile presenza nelle serate bolognesi. Ringrazio Fede per le super vacanze a casa sua e per le super feste a casa mia. Ringrazio Anna, amica ritrovata dopo tanto tempo è già parte integrante nel nostro gruppo. Nico è stato mio compagno a numerosi aperitivi di questo o di quel baretto, numerosi kebab notturni ed un viaggio indimenticabile attraverso gli USA. Con lui ricordo tutti gli argentini, Elena, Gioia e Carna, ma anche la Stefi l'Elena C. e l'Elena R. con il suo background. Ringrazio Fabri, e con lui Beppe. Anche con loro ho condiviso un fantastico viaggio oltreoceano e spero vivamente ti ripetere questo tipo di avventure. Ringrazio Teo che ci ha abbandonato senza neanche dire niente ma che andrò presto a trovare a Cipro, come lui la Serena che è scappata a Parigi e, visitarla, è già uno dei miei piani del futuro prossimo. Ovviamente ringrazio anche la Barbara che invece è rimasta qua da noi. Ringrazio Bucci e tutti i compagni di Ferrata e con lui la Sara. Edo, la Gaia, Pallot, Tullio, Ziz e la Silvia e tutto il gruppo S. Lazzaro e ovviamente con loro anche il Matte. Ringrazio le super psicologhe Laura ed Elena. Ringrazio anche la novelle psicologhe marchigiane Laura e Cri e con loro la Sarina e la Stefania. E poi tutti gli altri con cui ho condiviso momenti fantastici all'Università: Il Blond, Alagna, Patch, Marco, la Sarina, la Martina e Diana e tutti gli altri che in questo momento mi sto sicuramente dimenticando.

Ringrazio i 5 Zoccoli originali e tutti quelli che sono subentrati al loro posto. Walter anche lui sotto laurea sono sicuro che aspetterà me a festeggiare in maniera adeguata… Sandrino che stiamo attendendo per il suo ritorno in patria, ma anche casa e in squadra, Marco e Ludovica. E poi le new entries Mario e Massimo e tutti gli altri che prima o dopo sono passati nella mitica nella mitica via Zoccoli 5… tranne le pernacchie, loro no.

Ringrazio il Rosso della Casa (rossodellacasa.wordpress.com), eredità della Globo F.C. per avermi riaccolto in squadra dopo che, un anno di mia assenza, ha comportato la vittoria del campionato. Vorrà mai dire qualcosa? Comunque ringrazio il Capitano, i Burgi, Ale&Franz, Tommy, Monda, Dario, Giorgio, Giovanni, Pallot e tutti quelli che hanno fatto parte di questa grande squadra sperando in un futuro ancora vincente.

Ringrazio tutti gli amici di Crevalcore, per primo mio fratello con cui, suo malgrado, ho passato ogni genere di avventura. Ringrazio Ricky e Giò che presto andrò a trovare a Trieste, Tix con le sue raccolte di foto, Bobo il Boss e Guiz sperando di tornar presto insieme sulla neve, Alle e Vent, il futuro sindaco, Gemex per i futuri weekend a Milano. Ringrazio poi Paz, Ste, Leo, Albi e poi Filippo, Ciaccia, Mazza e tutti gli altri. Infine ringrazio Comme, che non mi ha picchiato dopo che ho ionizzato una weiss inondandogli il bar.

Ringrazio tutti gli alpinisti e arrampicatori dei CAI che mi hanno sostenuto e soprattutto assicurato durante le nostre avventure e che presto tornerò ad affiancare.

Ringrazio tutto il gruppo del liceo, di recente ci siamo riuniti per una fantastica cena. C'è chi ha già figli e chi si sposa ma il gruppo è sempre unito come una volta. Ringrazio il Toro e anche la Martina a questo punto, Mizio, Ranz, Jan e poi Luca che attendo per una sciata come si deve, Barbi, Mitch, Ricky, Umbi e tutti gli altri .

Infine, dal momento che questa tesi è stata svolta in California vorrei ricordare tutte quelle persone che hanno reso la mia esperienza oltreoceano un'esperienza unica. Elide, la mia più fedele compagna di viaggi, è stata parte integrante di tutta la mia California. Con lei ho condiviso pranzi, cene, giornate di studio ed esperienze di ogni tipo e non finirò mai di ringraziarla per avermi reso disponibile casa sua come fosse stata mia e sopportato e supportato in tutto questo tempo. Con Broc sono nate le migliori idee per tutti i nostri viaggi passati e futuri che senza di lui non sarebbero mai esistiti. Marco Borri mi ha accompagnato il lungo e in largo per l'america, ma anche per i migliori bar La Jolla, P.B, Downtown e TGI Frydays… Con David e Laura è iniziato tutto, sono stati i miei primi veri amici oltreoceano, con loro ho visto per la prima volta tutto, ho condiviso i primi interminabili viaggi in autobus, le prime cene ai fast food messicani e soprattutto ho condiviso l'esperienza archstone a partire dalla prima notte dormita sul tappeto. Vorrei ringraziare tutti quelli che sono passati dal mitico appartamento #355 dell'Archstone La Jolla. Con Eugenio non ho solo condiviso la camera ma anche mille ore di cazzeggio, di studio, di viaggio e di festa dal primo all'ultimo giorno americano. Senza Trulli non saprei come comportarmi dopo aver investito un cinghiale. Che dire dei Coviello's? se non ci fossero bisognerebbe inventarli, ringrazio anche la madre che mi spingeva a mettere ordine in camera e la nonna di Schkauri. *I want to thank Andrew for the precious suggestions in squash but also because, together with Ryan, they survived to the Italian invasion and I don't think it's easy. I can't forget Jia, the best roommate ever to have in the living room. I want to thank also my pseudo-roommate Amy, from Taiwan, that I really would like to visit soon.* Ringrazio Alberto per i tanti campetti a sfidare i cinesi, il Dirty per le serate in disco a Vegas e SD, Pit con i suoi preziosi consigli sulla pasta salmone e zucchini e i libri sulle filippine che però non sono riuscito a mandare in porto. Con Filippo ho condiviso ore di studio e stress sui maledetti libri di Ingolf *and together with Ahmed I spent hours and nights of great "high" level studies.* Ringrazio tutti gli altri italiani che mi hanno fatto sentire a casa: Livia, Paola e Gioele, Ale, Marcello, Rom che è ormai più californiana di tutti gli altri, la Chiara che non sono mai riuscito a vedere all'opera, Niccolò che mi ospiterà molto presto in terre orientali. Con lui ringrazio tutte le squadre di calcio e calcetto ed i loro capitani Tom e Mujaindin. *I want to thank Mulloy, Tremors and Ara who always made me happy with her* *sentences Jake and Cassiopee.* Devo anche ringraziare tutti gli altri ragazzi che, sparsi per gli altri campus, mi hanno ospitato e guidato nelle città della California. Matteo e Giacomo a LA, non dimenticherò mai le notti alla coop o alla tri-house. Anna, Carlotta, Silvia e Caterina nelle fantastiche feste di Santa Barbara, Carolina, Valentina, Chiara e Francesco nelle serate a San Francisco, Federico e Simonetta in giro per O.C.

Quando tutto questo è iniziato, ormai troppi anni fa, non avrei mai immaginato tutte le esperienze che avrei potuto fare durante questo tempo. A partire dalle prime lezioni nella scomoda 2.8 si sono avvicendate così tante cose che hanno reso questa esperienza unica e sono ormai quasi impossibili da ricordare tutte. Le ore di cazzeggio in palafitta, la provola in mensa, le festa da me o da Martina. I primi viaggi Ryanair da Bruxelles a Stoccolma, da Lisbona a Dublino, da Parigi a Villarosa, il viaggio di laurea a Praga, ma anche quelli devastanti di sci in Francia e quelli estivi in Spagna, Grecia e Irlanda. Il college, il corto, gli irish pub, spaghetti notte, la bruschetteria e i cicchettari e poi gli happy hours a Marina. Tutti calcetti fatti in campo o in facoltà e tutte le squadre ed i tornei, le partite di basket al campetto, tennis e squash, l'alpinismo e le arrampicate in palestra, in montagna o sulle torri. Le cene e le feste di carnevale alla Ragnatela e al Cheek. Le ore di "studio" in palafitta, il porto di Ancona, le chiacchiere al posto della lezione e poi l'aperitivo al number 10. Via Bergami in toto, via delle Fragole al freddo, Petrarca e Pasubio. E poi è arrivata la California con i viaggi, le feste, i viaggi ancora, e il surf, e un altro po' di viaggi, le Hawaii e poi il campus con tutte le palestre e gli sport fatti, con le partite di basket, di beach volley e i tornei maschili e co-ed, la geysel, la biblioteca dei nerd con il serpente e i canyon, la libray walk e l'i-walk con lo skateboard. MollyAnn con tutte le ore di litigi nel suo ufficio per inutili pratiche burocratiche. Le feste all'i-house e le dormite nelle lounge room e non. E poi il caffe V, l'OVT, il sierra summit, il warren ed Earl's place. Come non dimenticare il Porter's pub, e Jose's, il

Bar and Grill, il Tavern, il Cabo Cantina, on Broadway, le feste in piscina e da Gregoire e le spiagge dove non si può bere… no, non si può. Come non ricordare i community service e i poliziotti californiani… e quelli di Utah, Connecticut, Massachussets, Indiana e Texas? E poi le serate a Las Vegas, Los Angeles, New York… E quando sono tornato a Bologna tutto è ricominciato, con le feste, le uscite, il college e il pepita. Le nuove entrate del gruppo e chi ci ha abbandonato, e poi Birmingham e Berlino non si dimenticano.

Non so quante cose mi sono dimenticato, quello che è certo è che è grazie a tutte queste esperienze e a tutte queste persone che ora sono quello che sono. È grazie a loro se sto terminando questa esperienza nel modo migliore, con un voto sicuramente superiore a qualunque più ottimistica aspettativa di chiunque mi abbia visto lavorare, il sottoscritto per primo. Ma non era certo per il voto che mi sono impegnato in questi anni, quello che volevo era solo vivere al meglio questi anni di Università e l'ho ottenuto grazie a tutti voi. Grazie!