Università degli Studi di Bologna

FACOLTÀ DI INGEGNERIA

DOTTORATO DI RICERCA IN INGEGNERIA ELETTRONICA,
INFORMATICA E DELLE TELECOMUNICAZIONI

Ciclo XIX

ELECTRONIC SYSTEMS FOR AMBIENT INTELLIGENCE

Tesi di Dottorato di:

DAVIDE BRUNELLI

Relatori :

Chiar. mo Prof. Ing. **LUCA BENINI**

Chiar. mo Prof. Ing. **BRUNO RICCÒ**

Coordinatore:

Chiar. mo Prof. Ing. **PAOLO BASSI**

Settore Scientifico Disciplinare: ING/INF01 Elettronica

Anno Accademico 2006/2007

ELECTRONIC SYSTEMS FOR AMBIENT INTELLIGENCE

A dissertation submitted to the
DEPARTEMENT OF ELECTRONICS, COMPUTER SCIENCE AND SYSTEMS
OF UNIVERSITY OF BOLOGNA

for the degree of Doctor of Philosophy

presented by
DAVIDE BRUNELLI
born March 26, 1977

March 2007

Keywords

Ambient Intelligence

Wireless Sensor Networks

Energy Scavenging

Power Management

Scheduling Algorithms

To my wife Francesca,
my sister Elisa and
my parents.

Acknowledgements

This thesis would not have been possible without Francesca, my patient and loving wife, who has been a great source of strength and encouragement all through this work.

During these years I have had the opportunity to work, collaborate, and discuss with a wide range of people. These interactions have been the most rewarding part of my research work. In particular, I would like to express my gratitude to my academic advisor, Prof. Luca Benini, whose expertise, understanding, and patience, added considerably to my PhD experience. I appreciate his vast knowledge and skill in many areas, his visions and support throughout my research activity. I want to thank him for highly stimulating discussions (which generally took place walking to the train station for his coming back home) that have led to the discovery of many results in this thesis.

I am also grateful to my co-supervisor, Prof. Bruno Riccó for his excellent questions and valuable discussions which have been both inspiring and motivating.

I would like to thank Prof. Dr. Lothar Thiele for giving me the opportunity to work at ETH and to live in Zurich for seven months. I have appreciated very much his guidance and advice throughout my visit.

I am also grateful to Clemens Moser for his friendship and the pleasant research environment that he arranged. Much of the work on algorithms is based on his collaboration.

I would like to express my deepest gratitude to my parents, Libero and Nadia for their never ending support and understanding, and in particular my sister Elisa, which is one of the persons I mostly admire.

Thanks also to my colleagues of Micrel Lab Research Group and all my friends who helped me get through these years of PhD course.

My warmest thanks and kisses go to my nephews Alessandro, Beatrice and Giorgia for the smiles and laughter gifted to me.

Finally, I thank all who assisted and/or supported me in one way or the other. Your contributions and assistance were very much appreciated.

Thank you so much to all of you.

Bologna
March 6, 2007

Contents

List of Figures

List of Tables

About the author

Davide Brunelli has worked for his PhD program in the Microelectronic Research Group under the supervision of Prof. Luca Benini and Prof. Bruno Riccó. His research interests concern design and analysis of wireless sensor networks for ambient intelligence and ubiquitous computing. Particular emphasis is addressed on analysis of energy scavenging techniques, and optimization of harvesting sensor networks.

The author has also been visiting researcher at the Computer Engineering and Networks Laboratory of the Swiss Federal Institute of Technology - ETH (Zurich, Switzerland) under the supervision of Prof. Dr. Lothar Thiele

Introduction

The term Ambient Intelligence (*AmI*) has been coined to describe systems, which are embedded in the user's surroundings and are context aware, adaptive and proactive. Thus the environment serves and helps people by becoming sensitive and reactive to their presence, since electronics is ubiquitous and deployed everywhere. This visionary concept was forged by Weiser in 1991 [1, 2] and it is certainly one of the most profound visions in the mobile computing community.

> *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*
>
> — Marc Weiser, "The Computer for the 21st Century"

This promotes the creation of pervasive environments improving the quality of life of the occupants and enhancing the human experience. Moreover, it enables the interaction with users in a natural way: passively, by observing and trying to interpret the actions and intentions of people, but also actively, by learning their preferences and adapting the system parameters. This augmented space can be an open or a close environment, constrained in a physical location, or spread across a large distance, where technologies are distributed, not intrusive and always present. (A clear analogy is electricity: it is everpresent and widely used, but we do not think about it, and most of the time we are not aware of it.)

Ambient intelligent systems are basically very heterogeneous and require an excellent cooperation between several hardware/software technologies and disciplines, including signal processing, networking and protocols, embedded systems, information management, and distributed algorithms. Thus, the design of such systems is interdisciplinary by nature and requires an intense dialog between specialists of multiple domains. In particular, an *AmI* system requires the adoption of distributed sensors and actuators to create a technological framework, able to interact transparently with the users. Since the vision

is to integrate electronics into everyday objects (clothing cars houses public places, ...) these *Pervasive Devices* are quite different from current personal electronics such as hand-held, mobile or stationary computing systems.

The dependence on the existence of large quantities of wireless sensors deployed throughout the environment makes Wireless Sensor Networks (*WSN*) one of the most important enabling technologies which contribute to build the paradigm of *AmI*. Such micro-systems typically incorporate both sensors and communication functions. Each sensor node monitors its local environment, processes and stores the collected data so that other nodes can use them spreading the information through a wireless link. For this reason pervasive networks of wireless sensors nodes have the potential to significantly impact society and to create large market opportunities.

Unfortunately, while advances in IC fabrication technologies, circuit designs and networking techniques have greatly reduced the size and power consumption of potential wireless sensor platforms, the amount of available on-board energy still limits severely the operating lifetime of distributed embedded systems. Although research continues to develop higher energy-density batteries and supercapacitors, power sources are rapidly becoming the bottleneck which limits the widespread deployment of wireless sensor networks. For applications where many autonomous devices are distributed throughout the environment it would clearly be impractical to replace batteries frequently. This leads to the concept that the *AmI* devices must harvest their energy directly from the surroundings (exploiting light, EM radiation, thermal gradients, vibration, air flow, etc.). This method of powering is called *Energy Scavenging* and introduces great challenges that cover a wide spectrum of hardware technologies.

So far much attention has been paid to application scenarios that often look like dreams in a far future. In this thesis we pay attention to the realities of hardware technologies and the design needed to realize these visions, attempting to propose solutions on open issues and filling the gap between *AmI* scenarios and hardware reality.

The physical implementation of an individual wireless node is constrained by three important metrics which are outlined below.

Small Form-Factor

Embedding the wireless devices into the infrastructure of daily life (walls, furniture, lighting, etc.) requires a small form factor for each node. For many scenarios, sizes smaller than 1 cm^3 are necessary. A very high level

of integration is mandatory if such small dimensions have to be achieved.

Low Cost Implementation

For commercial viability, the cost of a sensor node must be very small, since it is designed to be part of a large mesh. Moreover for network reliability, the node density (# nodes/area) must be high. Thus, the cost of each node must be extremely small (<1 €).

Power

To reduce installation costs and to allow flexible deployment, most nodes must be untethered and have their own energy source. Frequent replacement of the energy source is unfeasible because it would result in a high maintenance cost. Thus, the nodes must scavenge their energy from the environment. This requires the average power dissipation of the node to be extremely low ($< 100mW$).

During the PhD research activity, we have tackled all the constraints listed above and have proposed solutions to overcome the current *WSN* limits. The specific contributions are presented and discussed in the following:

- We have designed and implemented a wireless sensor node focusing on the small size and the flexibility of reuse in different applications. The main idea was to develop a simple and general purpose architecture for rapid analysis, prototyping and deployment of *AmI* solutions. In particular our vision pointed to develop applications for wearable computing and human computer interaction (HCI). Moreover, the system built is a suitable platform for testing and measuring the features and the constraints of a sensor network (radio communication, network protocols, power consumption and autonomy);

- To prove the flexibility of the architecture designed, we have implemented a wearable system for human posture rehabilitation. The application was developed in conjunction with biomedical engineers who provided the algorithms for signal processing.

- Exploiting the developed sensor node as testbed platform, we focused also on the problem of autonomy and energy management, using systems and techniques which harness energy form the surroundings environment. A number of different methods to harvest environmental energy have been analyzed and have been successfully tested to power wireless sensor devices;

- We have principally investigated the design of photovoltaic (*PV*) harvesters, converting outdoor and indoor light intensity into electrical po-

wer. We have contributed to develop a scavenger that exploits miniaturized *PV* modules performing a highly efficient conversion at a minimum energy cost. Much attention has been paid to achieve a small size factor for the harvester circuit and for storage devices;

- Although harvested energy seems to be virtually inexhaustible, it is characterized by significative and unpredictable variations. In this thesis we show that conventional scheduling algorithms (like e.g. EDF) are not suitable for this scenario. Based on this motivation, we have constructed optimal scheduling algorithms that jointly handle constraints from both energy and time domain;
- Since some sensor nodes are not equipped with any operating system, scheduling of activities is not practicable. We have adapted results of the well-established framework of multiparametric programming to the emerging area of energy harvesting systems. The result is the development of a predictive controller to tune the parameters of the sensor node (such as rate of activity, transmission power ...) that optimize the long term system performance;

The remainder of the thesis is organized as follows. Chapter 1 gives an overview about Wireless Sensor Network (*WSN*), illustrating the principal features of this technology and the key challenges still open. It concludes with a short list of the real solution and prototypes proposed by academic research and manufacturers. The sensor node developed in the first part of my research activity is presented in Chapter 2. Initially designed for applications of Motion Capture, it embeds an accelerometer and can be exploited also for HCI and wearable applications. The example of biomedical application for the optimization of human posture and movement is discussed in Chapter 3. In this thesis we have adopted the sensor node also as testbed for research activity on perpetual powering systems. In fact to introduce energy harvesting techniques, Chapter 4 gives a survey of the principal methods of converting energy from the environment in order to power sensor nodes. Opportunities and limitations are also discussed. In Chapter 5 we discuss about the design of a high efficient photovoltaic harvester for *WSN*, while algorithms of energy managements for *WSN* are proposed in the next Chapters. In particular Chapter 6 tackles the problem of how to schedule tasks on a sensor node. For applications which do not require a scheduling activity, Chapter 7 discusses a method to determine decisions that optimize the long term system behavior (such as reactivity, aver-

age throughput, minimal sensing rate). Finally, conclusions are drawn, reporting the main research contributions that have been discussed throughout this dissertation.

Chapter 1

Overview of Wireless Sensor Networks

A wireless sensor network consists of many energy-autonomous small embedded systems equipped with sensor and radio-transmitter. The sensor nodes are distributed throughout an area of interest and each one monitors its local environment, processes and stores the collected data so that other nodes can use them. Network nodes share this information via a wireless link.

Since these networks often are deployed in regions that are difficult to access, the nodes should not require maintenance. They must be energetically autonomous, using batteries that do not need to be replaced or recharged. Moreover, since network may be have hundreds or thousands of nodes, sensor nodes should be cheap.

The main purpose of this chapter is to overview the state of the art in sensor node design, and to present the main constraints in wireless sensor networks. To achieve these goal, the chapter begins describing emerging sensor network applications (section 1.1). Section 1.3 describes characteristic, requirements and functional components of a Sensor Node, highlighting key research challenges due to the use of constrained resources. The chapter concludes with an overview of current sensor network solutions proposed by universities and manufacturers.

1.1 Wireless Sensor Network Applications

Although the most part of today sensors are still wired, wireless sensors offer significant advantages over wired ones, can reduce the cost of the system and enable applications that are otherwise unimaginable. Nevertheless today's WSN market is still fragmented, largely due to the unique characteristics of

each application space. Unlike the Internet, where TCP/IP is the standard transport protocol upon which all the internet applications can be built, sensor networks have yet to define common standards on which most application can be implemented. In the following, we discuss a list of applications areas that are expected to be early adopters of wireless sensor networks and that can give the first impulse to a standardization process.

Asset and wharehouse management

Wireless sensors may be used to monitor and track assets such as trucks or other equipment, especially in an area without a fixed networking infrastructure. Sensor Networks may also be used to manage assets for industries such as oil, gas and utility.

Automotive

With the features of a wireless sensor network, car will soon be able to talk to each other and to roadside infrastructures. Sensor may be applied on wheels for alert messages and driver safety assistance. During an emergency brake, an alert message from the breaking car can be broadcast to nearby cars so that preventive measures may be taken. Another interesting application is in the collection of real-time traffic. Given a vehicle, cars coming from the opposite direction may have sensed timely and valuable information. As the vehicles pass each other, they exchange summary information on the location of traffic jams and the speed and density of traffic or information that may be generated by ground sensors. These summaries propagate from vehicle to vehicle and can be used by drivers to avoid traffic jams and plan alternative routes.

Building environment and control

Sensor embedded in a building can drastically cut down energy costs by monitoring the temperature and lighting in the building and regulating the heating and cooling system and lights. Coupled with the security systems, the wireless sensor network may detect unauthorized intrusions or unusual patterns of activity in the building.

Environmental monitoring

Environmental monitoring is one of the earliest application of sensor networks [3]. Sensor can be used to monitor conditions and movements of animals in wildlife habitat where minimal disturbance is desired. Sensor can also monitoring air quality, track pollutants and discover biological and chemical hazards to provide early warnings.

Health Care

Elderly care can greatly benefit from using sensor networks that monitor

vital signs of patients and are remotely connected to doctor's offices, or simply alert when a patient falls or feels loneliness. Wireless sensor network may be also used as a personal area network (PAN) equipped with sensor able to understand the movement and the behavior of the people. This kind of system can become a valid interpreter of human gesture, supporting people who suffer of physical disability and providing imminent feedbacks when occurs.

Industrial process Control

A typical equipment manufacturer spends thousands of euro in services and maintenance every year. The costs can be reduced introducing wireless sensor networks that may be used to monitor manufacturing process or the condition of industrial equipment and provide dynamic alert for imminent failures.

Military battlefield awareness

Military supporting was the first purpose that gave life and money to wireless sensor network research. Real-time battlefield intelligence is an essential capability of modern command, control and communication systems. Wireless sensors can be rapidly deployed, either by themselves, without an established infrastructure, or working with other assets such as radar arrays and collect immediately important information. Moreover in military applications, the cost factor is not as dominant as in commercial solutions.

Security and surveillance

An important application field of sensor networks is in security monitoring and surveillance for buildings, airports, subways, or other critical infrastructure such as power and telecom grids or roads and motorway. Imager or video sensor can be very useful in identifying and tracking moving entities, although the require higher-bandwidth communication links. Heterogeneous systems that comprise lower-cost sensors, such as motion or acoustic sensors, that can act as triggers for imagers, can be very cost effective and efficient. Since it is rather difficult for security guards to continuously watch a set of video monitors when most of the time nothing occurs, the overall security effectiveness will be improved when motion and/or acoustic detectors generate an alert. These wireless devices equipped with acoustic sensors can locate a sound wave (by determining the magnitude of the sound and the angle of arrival, and performing primitive frequency analysis). The nodes organize themselves dynamically and convey the location information periodically, or

on-demand, to controller nodes, which then take appropriate actions under real-time constraints.

Traffic Control

Sensor networks have been used for vehicle traffic monitoring and control for quite a while. Most traffic intersections have either overhead or buried sensors to detect vehicles and control traffic lights. Furthermore, video cameras are frequently used to monitor road segments with heavy traffic, with the video sent to human operators at central locations. However, these sensors and the communication network that connect them are costly; thus, traffic monitoring is generally limited to a few critical points. Inexpensive wireless ad hoc networks will completely change the landscape of traffic monitoring and control. Cheap sensors with embedded networking capability can be deployed at every road intersection to detect and count vehicle traffic and estimate its speed. The sensors will communicate with neighboring nodes to eventually develop a "global traffic picture" which can be queried by human operators or automatic controllers to generate control signals.

1.2 Requirements and components of a Sensor Node

Sensor node hardware can be grouped into three categories, each of which entails a different set of trade off in design choices.

Augmented general-purpose computer

Examples includes low-power PCs, embedded or custom designed PCs, personal digital assistant (PDA). These nodes typically run off-the-shelf operating systems such as Windows CE, Linux, or real-time operating systems and use standard wireless communication protocols such as - IEEE802.11 or Bluetooth. Because of their relatively higher processing capability, they can accommodate a wide variety of sensors ,ranging from simple microphones to more sophisticated video cameras. Compared with dedicated sensor nodes, PC-like platforms are more power hungry. However, when power is not an issue, these platforms have the advantage that they can leverage their availability of fully supported networking protocols, popular programming languages, middleware, and other off-the-shelf software.

Dedicated embedded sensor nodes

Examples includes the Berkeley mote family [4], Ember nodes [5] and Tmotesky [6, 7]. These platforms usually use commercial off-the-shelf

chip set with emphasis on small form factor, low-power processing and communication, and simple sensor interfaces. Because of the simple microcontroller used, these platform typically support few programming languages (asm, C). However, in order to keep the program size small to accommodate thy small memory space, programmers of these platforms are usually given full access to hardware.

System-on-chip nodes

Examples of SoC hardware include smart dust [8] and the BWRC picoradio node [9]. Designers of these platforms try to push the hardware limits by fundamentally rethinking the architecture of a sensor node at chip design level. The challenge is to find new ways of integrating CMOS, MEMS, and RF technologies to build extremely low-power and small footprint sensor nodes. Since these platforms are currently in the research pipeline with no instruction set, there is no software platform support available.

This thesis focuses on the second category shown above. In next paragraphs we will discuss only on embedded sensor node calling them simply *sensor node*.

A sensor node is usually composed of a power unit, a processing unit, a sensing unit and a communication unit. The processing unit is responsible to collect and process signals captured from sensors and to transmit them through the network. Sensors produce a measurable response to a change in a physical condition. The wireless communication channel provides a medium to transfer signals from sensor to exterior world or to a computer network.

The major characteristics and requirements of a sensor node can be listed in the following:

Energy-efficiency

Power consumption is and will be the primary metric to design a sensor node and reducing power consumption requires optimization. Nodes have a limit amount energy resource that determine their lifetime. Since it is unfeasible to recharge thousands of nodes, each nodes should be as energy-efficient as possible. In many application scenarios, the target node lifetime typically spans from 1 to 5 years. This imposes drastic constraints on power consumption. With a single 1.5-V AA alkaline battery, the average power consumption ranges from 100 to 10 microwatts, for a node lifetime ranging between two and seven years. Considering that today's commercially available radio transceivers typically consume tens of milliwatts, keeping all part of the node constantly active is clearly im-

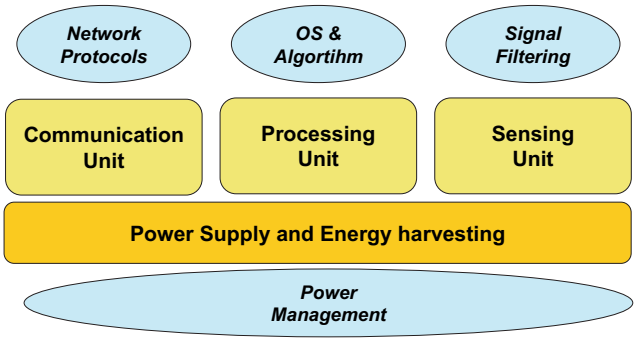


Figure 1.1: Architecture of a generic sensor node

possible. One solution may be introducing duty cycling on the order of 0.1 percent to 1 percent.

Low-cost

Since network may be have hundreds or thousands of nodes, sensor nodes should be cheap.

Wireless

The sensor nodes need to be wireless. In many applications, the environment being monitored does not have installed infrastructures for communication. Thus, the nodes should use wireless communication channels.

Multi-hop

A sensor node may not reach the base station. The solution is to communicate through multi-hop. Another advantage is that radio signals power are proportional to r^{-4}, where r is the distance of communication. Thus it can be more energy economic transmit many short-distance messages than one long-distance message.

Distributed processing

Each sensor node should be able to process local data, collect data from environment, aggregate it, and transforming it in information.

Fig. 1.1 presents the system architecture of a generic sensor node, that is composed of four major blocks: power supply, communication, processing and sensors. The figure also illustrates some challenges for wireless sensor network. A power management layer is necessary to control the energy level and the main resources of a sensor node, even when other energy sources are dynamically added or the energy is harvested from the environment. New network protocols are necessary, including link, network, transport, and application layers to solve problems such as routing, addressing, clustering and

synchronization, and they have to be energy efficient. A micro-kernel for sensor node should help in developing applications. Many operating system exists for small devices (PDAs, cell phone) but not as simple as a sensor node. Algorithm for filtering and sensor signal processing are also needed.

In the next subsections we investigate and analyze the main characteristics of the processing unit, the power supply layer, and the communication unit.

1.2.1 Processing unit

Since the sensor node is expected to communicate, to process and to gather sensor data, it must have processing units. The central processing unit of a sensor node determines to a large degree both the energy consumption and the computational capabilities. Many different types of commercially available products can be integrated in a node: microcontrollers, microprocessors or FPGAs, which allow a big flexibility for CPU implementations.

FPGA

Nowadays FPGA present some disadvantages, the major of them is the consumption. Although there is at the market low-power FPGAs, their power consumption is not as low as a sensor node should be. It does not mean that FPGAs are not good solution for the near future. May be with the development of ultra- low power FPGAs, these devices will be a good solution for sensor node.

Microcontrollers

Microcontroller includes not only memory and processor, but non-volatile memory and interfaces such as ADCs, UART, SPI, counters and timers. There are many types of microcontrollers ranging from 4 bits to 32b bits, varying the number of timers, ADC resolution, power consumption, etc.

1.2.2 Power supply unit

So far, The power supply unit usually consists of a battery and a dc-dc converter and has the purpose to power the node. It is important to chose the battery type since it can affect the design of sensor node. Recently alternative solutions for powering a sensor node has been introduced and have been also the most important subject of thesis. Energy scavenging techniques, that are methods to extract and convert environmental energy (thermal variations, vibration, solar irradiance, ...) into electrical power for sensor nodes, seems to be the a valid solution to achieve perpetual WSN applications.

1.2.3 Communication

Sensor node must communicate among them and to a base station using a wireless communication channel. The networking capability of WSNs is built up in layers. The lowest layer controls the physical device. We explore three possibilities, optical, infrared and radio frequency.

Optical communication - laser

The advantages of optical communication are:

- Security since there is no broadcast and if the channel is intercepted it would interrupt the signal;
- High bandwidth communication;
- No need for antenna.

The disadvantages are:

- Needs line of sight, since the laser beam of the transmitting device must be line up to the optical receiver;
- Sensible to atmospheric conditions;
- The communication is directional and due to the fact that sensor nodes could be deployed, this is not an attractive solution.

Infrared

Infrared communication is usually directional with a limited field of view. To enlarge the field of view, transmitting diode should be replicated on the board increasing the power consumption. Infrared systems advantage is no need for antenna. Moreover the designer must also know the short-range of infrared communication (few meters).

Radio-frequency

RF communication is based on electromagnetic waves and it is the most common wireless channel used in sensor network systems. One of the most important challenges in RF is to reduce the energy consumption. Several aspect affect the power consumption of a radio, including the type of modulation , scheme used, data rate, transmit power. In general radio can operate in four distinct modes of operation: transmit, receive, idle and sleep. Most radios operating in idle mode result in high power consumption, almost equal to the receive mode, thus it is important to shutdown the radio. The First solutions adopted very simple radio transceiver and he development of ad-hoc and low-power network protocols for medium access control (MAC) and routing was necessary. One of these radio devices is TR1001 and it is still used.

Recently some network standards for personal area network (PAN) and wireless sensor networks, such as Bluetooth [10] and Zigbee [11], have been proposed to community. Bluetooth is a standard that realizes small form factors, low-cost, short-range radio links. The Bluetooth standard provides specifications for the radio link, baseband link, and link manager protocol.

Bluetooth performs small star network composed by 8 members at maximum, this increase the sensor network complexity. Bluetooth is designed mainly for computer cable replacement, so it is not the best choice for a sensor network. Nevertheless the high data rate of Bluetooth suggests to use it for gateways or sensor nodes that need to transmit at higher bandwidth such as a video.

Zigbee and IEEE802.15.4 are standards developed for wireless sensor networking. IEEE802.15.4 is proposed by the IEEE and performs low data rate solution with multi-month to multi-year battery life and very low complexity. It is intended to operate in an unlicensed, international frequency band. IEE802.15.4 defines the physical layer (PHY) and the media access controller (MAC). For these optimized short-range wireless solutions, the other key element above the Physical and MAC Layer is the Network/Security Layers for sensor and control integration. The ZigBee Alliance is in the process of defining the characteristics of these layers for star, mesh, and cluster tree topologies.

1.3 Overview of state of the art

Modern research on sensor networks started around 1980 with the Distributed Sensor Networks (DSN) [12] program at the Defense Advanced Research Projects Agency (DARPA). Technology components for a DSN included sensors (acoustic), communication (high-level protocols that link processes working on a common application a resource-sharing network), processing techniques and algorithms (including self-location algorithms for sensors), and distributed software (dynamically modifiable distributed systems and language design).

Current sensor networks can exploit technologies not available 20 years ago and perform functions that were not even dreamed of at that time. Sensors, processors, and communication devices are all getting much smaller and cheaper. Commercial companies such as Ember [5], Crossbow [13], and MillenialNet [14] are now building and deploying small sensor nodes and systems. These companies provide a vision of how our daily lives will be en-

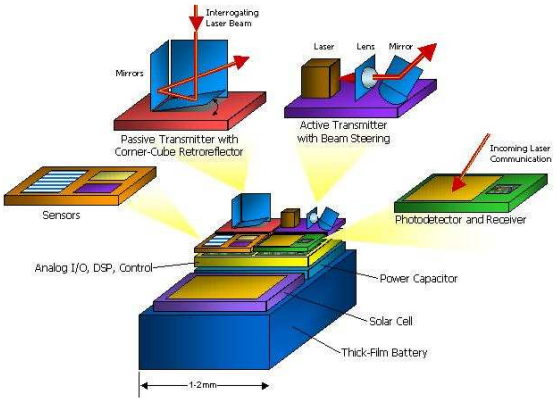


Figure 1.2: Smart Dust architecture

hanced through a network of small, embedded sensor nodes. In addition to products from these companies, commercial off-the-shelf personal digital assistants (PDAs) using Palm or Pocket PC operating systems contain significant computing power in a small package.

In the following we present a series of commercial and academic solutions of wireless sensor network and their main features.

Smart Dust

The Smart Dust [15, 8] is a project of Berkeley university and aims at the develop sensor node of millimetric size. Their focus is on miniaturization of sensor node so that it has a dust size. Since this is a long time project, the first step was the development of the a node family called WeC.

The dust nodes are comprised of various subsystems from different fabrication technologies (Fig. 1.2). Many sensors, including temperature, pressure, and acceleration sensors, from MEMS and CMOS processes can be attached to a mote. An ASIC handles measurement recording, data storage, and system control. A receiver circuit converts photocurrent from an incoming laser into a data stream to be used to interrogate or reconfigure the mote. Several transmission systems can also be utilized, such as a passive corner cube reflector (CCR) for communication to a base station, or an integrated laser with beam steering MEMS structures for inter-mote communication. Finally, all of the components are mounted onto a thick-film battery charged by a solar cell.

iBadge

iBadge [16] is also a solution from University of California, Los Angeles (UCLA). It is equipped with sensors such as microphone, localiza-

tion sensor and temperature sensor. The iBadge is not only a collection of sensors; it possesses enough computational power to process the sensed data. Two processors, Atmel ATMEGA and TI DSP C5416, share the workload according to their computational ability (Fig. 1.3). The DSP processes the computational more expensive speech data and the magnetic sensor data, whereas ATMEGA is interfaced to sensors with simpler data structure and the Bluetooth module. The Bluetooth module provides the wireless connection to the rest of the infrastructure, to which all the collected and processed data is sent. It has also a simple RF transceiver to communicate to small nodes. It should be a good solution for gateway.

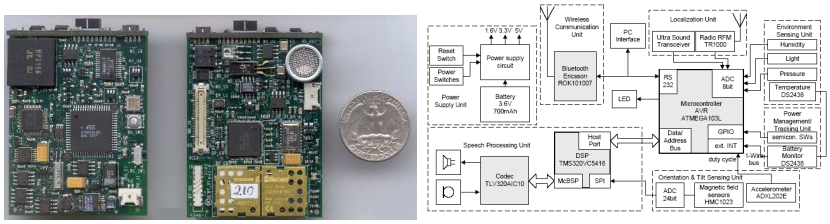


Figure 1.3: iBadge and its architecture

μ Amps

μ Amps [17] (micro-Adaptive Multi-domain Power Aware Sensors) node consists of a stack of three or four printed circuit boards. The top board contains the radio, including the RF circuitry and the FPGA used for digital coding and decoding. The second board contains an Intel StrongARM processor and associated RAM and flash ROM. Also on the processor board are an acoustic sensor (microphone, amplifier, filter, and analog-to-digital converter) and a collection of dc/dc power converters that service the entire node. The optional third board in the stack is an additional sensor module to replace the acoustic sensor on the processor board. Many components incorporate more complex power controls that allow gradual power scaling.

The μ Amps node can be easily adapted to different applications by designing an appropriate sensor board (FIG. 1.4). The bottom board in the stack contains the power source: four AAA batteries. The μ Amps processor block consists of a StrongARM microprocessor along with low-power static RAM and flash ROM. The StrongARM processor was chosen because of its high performance/power ratio and its built-in variable frequency (59-206 MHz) core clock generator. An adjustable clock frequency enables dynamic voltage scaling, in which processor voltage

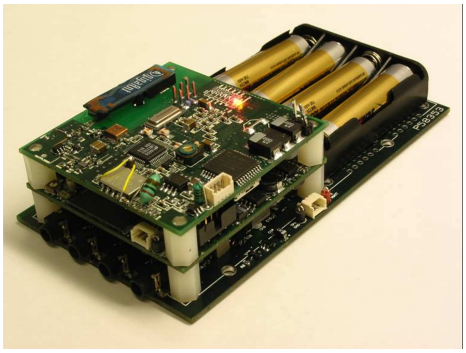


Figure 1.4: μAMPS sensor node

and clock frequency are adjusted together to provide an energy-latency trade off. The μ Amps radio consists of a digital baseband processor (implemented on an FPGA) and an RF frontend. The digital component is responsible for encoding, decoding, and error detection/correction. It also controls the timing of the transmitter and receiver according to the TDMA scheme employed by the network protocol. The core of the RF circuitry is an LMX3162 2.4 GHz radio transceiver chip from National Semiconductor.

Mica motes

The node architecture [13] consists of five major modules: processing, RF communication, power management, I/O expansion, and secondary storage. In the first MICA [4] implementation (Fig. 1.5), the microcontroller is an ATMEGA103L running at 4 MHz. It is an 8-bit microcontroller with 128 Kbytes of flash program memory and 4 Kbytes of system RAM. Additionally, it has an internal 8 channel, 10-bit ADC, 3 hardware timers, and 48 general-purpose I/O lines. It has one external UART and one SPI port. The RF module consists of an RF Monolithic TR1000 transceiver and the set of discrete components required to operate the radio. The radio interface gives direct control over the transmitted signal allowing for the use arbitrary communication protocols.
Currently generation of Mica are equipped with different microcontrollers and transceivers and are compliant with the Zigbee standard.

MillennialNet solution

MillennialNet inc. [14] presents a small size node (iBean) for wireless sensor network. New iBean family is also compliant for IEEE802.15.4/Zigbee protocols. The iBean Endpoint is a tiny microcomputer with analog and digital interfaces and a wireless communications link. The i-Bean antenna enables communications over a range of 20 meters or more to an

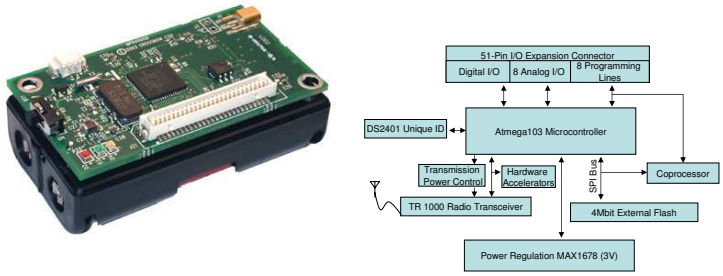


Figure 1.5: MICA node

i-Bean Gateway or i-Bean Router

Figure 1.6: iBeam modules

Intel mote

The Intel Mote [18] hardware is a modular, stackable design that includes the following components in a package that is about half the size of the original Berkeley mote. Intel Research is using the Zeevo module on the main board (containing an ARM1 core, SRAM and Flash memory, and Bluetooth wireless technology), an optional power supply regulator, and sensor boards. The mote platform can accommodate other features as well, such as alternate radio, debug and actuator boards. A backbone interconnect provides power and bidirectional signaling capability.

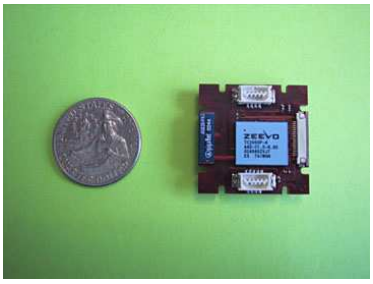


Figure 1.7: Intel mote - stackable node

TmoteSky

Tmotesky has been proposed by Mote*iv* [7]. Tmotesky is a node platform for extremely low power, high data-rate, sensor network applications designed with the dual goal of fault tolerance and development ease. Tmotesky boasts the largest on-chip RAM size (10kB) of any mote, the first IEEE 802.15.4 radio, and an integrated on-board antenna providing up to 125 meter range. Tmotesky offers a number of integrated peripherals including a 12-bit ADC and DAC, Timer, I2C, SPI, and UART bus protocols, and a performance boosting DMA controller. Tmotesky offers a robust solution with hardware protected external flash (1Mb in size), applications may be wirelessly programmed to the Tmotesky module. In the event of a malfunctioning program, the module loads a protected image from flash. Toward development ease, Tmotesky provides an easy-to-use USB protocol for programming, debugging and data collection. Designed with applications in mind, the Tmotesky module is FCC certifiable and may be added into existing products.

Figure 1.8: Tmotesky sensor node

BTnode

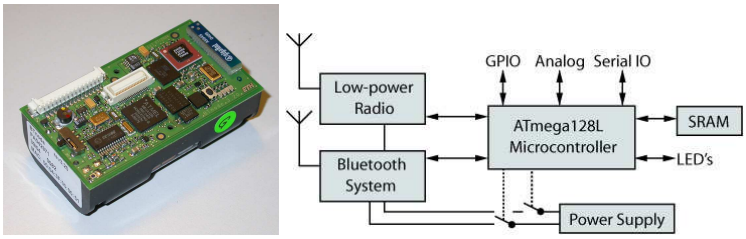


Figure 1.9: BTnode

The BTnode is an autonomous wireless communication and computing platform based on a Bluetooth radio and a microcontroller. It serves as a demonstration platform for research in mobile and ad-hoc connected networks (MANETs) and distributed sensor networks. The low-power radio is the same as used on the Berkeley Mica2 Motes, making the BTno-

de rev3 a twin of both the Mote and the old BTnode. Both radios can be operated simultaneously or be independently powered off completely when not in use, considerably reducing the idle power consumption of the device.

Chapter 2

A small low-power wireless architecture for *AmI* applications

This chapter presents the design and implementation of a wireless sensor node for a Motion Capture system with Accelerometers (WiMoCA). It is composed by a tri-axial integrated accelerometer, a microcontroller and a wireless transceiver. The use of a single integrated tri-axial accelerometer allows to overcome inaccuracies of orthogonally mounted bi-axial accelerometers and to build a smaller device compared to other sensors presented in literature. WiMoCA nodes have been exploited to build a Wireless Body Area Sensor Network (WBASN) that allows to implement a wireless/wearable distributed gesture recognition system where nodes are mounted on many parts of the human body. We describe the hardware architecture and all the software layers supporting the recognition system. We also show characterization experiments on WiMoCA nodes that highlight how their performance and power consumption levels make them suitable to HCI applications.

2.1 Introduction

The ever increasing diffusion of mobile and ubiquitous computing is pushing the development of a new generation of human-computer interfaces (HCIs) to provide natural and context-aware access to personalized services. Moreover, mobile and ultra-portable devices such as cell-phones and PDAs ask for the design of alternative input systems to overcome the limitations of traditional touch-based interfaces. Nowadays, motion tracking and gesture recognition

technologies are gaining momentum and will likely play an important role in the coming decade. At the same time, sensor technology is enabling the development of small form-factor devices that can be connected together to form a wireless sensor network (WSN). Putting it all together, the technology is mature to develop natural human-computer interfaces by exploiting sensors networks for detecting user movements. However, several challenges are to be faced. For example, wearability, flexibility and low power consumption are key requirements that are hard to jointly satisfy.

Accelerometers are effective for recognition purposes because of their small size and because they are immune from electromagnetic interferences and obstacles [19]. In this work we present the design and implementation of a wireless sensor node called WiMoCA (Wireless Motion Capture with Accelerometers) equipped with a tri-axial accelerometer. We also present the wireless body area sensor network made by WiMoCA nodes that we exploited to implement a distributed gesture recognition system for HCI applications. Furthermore, we describe all the layers needed to implement the body area network: the hardware of the nodes, the network and the application layer.

The gesture recognition system makes a step forward with respect to traditional gesture recognition systems by exploiting wireless sensor networking technology to detect postures and gestures of different parts of the human body. To the best of our knowledge, this is the first implementation of a wireless sensor network for gesture recognition applications.

Applying sensor networking technology to this purpose imposes original design requirements compared to traditional sensor networks. First of all, nodes must be extremely light and wearable to avoid obstructing user movements. Second, the software on the nodes must be designed to handle a continuous real-time data stream composed by acceleration values. This is different from networks designed for environmental monitoring that are commonly designed to handle sporadic events. From a network perspective, nodes are all visible and directly connected to a base station (installed as an extension board on a palmtop PC) that represents also the gateway that connects the network to a PC where is located the user application that makes use of human gesture. Thus, the topology of the body network does not require the implementation of routing algorithms and data aggregation algorithms.

If we refer to previous gesture recognition solutions, our system has many elements of novelty. First of all, being based upon wireless nodes, it overcomes other solutions where sensors are connected through cables. Moreover, being equipped with a single integrated tri-axial accelerometer, WiMoCA has a smaller form-factor compared to previous solutions. Our module provides better accuracy with respect to traditional orthogonally mounted accelerome-

ters [20, 21], thus reducing re-calibration frequency.

Commercial solutions like Mica Motes [13] are designed to handle sporadic or slowly changing events (such as temperature and pressure variations) and to interface with web applications for environment monitoring. As such, they are equipped with embedded operating systems [22], that support a complete network stack. Comparing to these solutions, our system has been designed for real-time interactive applications with low-power requirements and for this reason we focused on minimizing software overhead by implementing our own component drivers and communication layer. Moreover, Mica are equipped with analog bi-axial accelerometers, that require additional ADC conversion and are less tailored to gesture recognition applications compared to digital integrated tri-axial devices equipped by WiMoCA.

The software support we developed to drive the module, exploits the low power consumption of sensing unit to improve battery lifetime and provide a network stack, (physical and MAC layer) to drive the wireless transceiver, interface with other modules and organize accelerometer data in packets to be sent to a remote machine for further processing.

Even if the main target of WiMoCA is gesture recognition, its characteristics make it suitable to be used in industrial and medical applications. The nodes have been designed in modular way, enabling fast replacement. This is of critical importance especially when nodes are placed in dangerous or inaccessible environments.

The contribution shown in this chapter is threefold. First, we describe and characterize (in terms of accuracy, performance and power) the hardware architecture of new sensor module. Second, we describe the software architecture developed for the node and the architecture of the WBASN that connects WiMoCA module together. Finally, we demonstrate the use of the WiMoCA WSN in a vertically integrated application in gesture recognition where nodes are mounted on many parts of the human body.

2.2 Related Work

To develop wearable and unobtrusive interfaces, accelerometers must be mounted on lightweight and tiny boards without wires and cables for power supply and connectivity. To this purpose, some research laboratories recently targeted the designed low-power, wireless and wearable devices. Hand and head motion tracking systems have been designed [23] for Virtual Reality applications. This class of systems are dedicated and optimized for tracking specific body parts and movements. WiMoCA targets general purpose gesture recognition, and for this reason is designed to be installed in any body part.

A consistent amount of work has been done to build absolute orientation detection devices mainly targeted to robot navigation. In these kind of devices, accelerometers are used to compute orientation referred to local vertical direction, while magnetometers are used to compute the direction of gravity with respect to the hearth. Recently, MARG modules have been designed [21], which are equipped with triads of orthogonally mounted accelerometers, magnetometer and angular rate sensors. They target 3 DoF (Degrees of Freedom) orientation measurements of static or slowly moving rigid bodies. Since the module does not have wireless communication capabilities, data must be transmitted over a wired link for further processing. Accelerometers, magnetometers and gyroscopes are exploited in [20] to build a wireless portable absolute orientation estimation device. The device uses a Kalman filter to filter out accelerations due to user movements. Being designed to be part of a sensor network, the device uses a Bluetooth module for network communications.

Comparing this class of sensors to WiMoCA, the main difference is that they are designed to perform tracking of absolute orientation. In our design, we are interested in gesture recognition instead of tracking, and for this reason we do not need complex filtering algorithms to compensate the error introduced by the integration steps. We use a geometric algorithm instead, which uses accelerometers as inclinometers. Moreover, being not designed to detect absolute orientation, we do not use gyroscopes and magnetometers. For this reason, WiMoCA shows a smaller form-factor, making it more suitable to be mounted on various body parts without impacting normal user movements.

Accelerometers and gyroscopes have been used also to build human-computer interfaces [24] [25]. This class of HCI interfaces differs from WiMoCA either because they target a specific movement or because they are not designed to build a distributed general purpose recognition system. In fact, WiMoCA provides advanced wireless networking hardware and software support to communicate with other WiMoCA modules to build a body area network.

Wearable inertial sensors are a low-cost, low-power solution to track gestures and, more generally, movements of a person. The implementation of a body-centric network mounting inertial sensors has been explored in many fields. Examples are in context-aware applications [26] [27] and monitoring of patient activity in the medical domain [28]. A combination of accelerometers, magnetometers, temperature and light sensors to be worn by users has been applied to help indoor navigation [29] and infer user's location. Many research studies focus on hybrid sensor networks or sensor fusion techniques. The sensing elements are distributed along the body and data processed both off-line or in real-time [30], [31]. Even if these solutions are suitable to specific applications, they are not tailored to applications requiring high wearability

and very-low power consumption.

Recently, wearable systems have been implemented to enable input operations through gesture recognition, for example in [32] RFID tags are used to recognize gesture by the location of the wrist relative to a region of the body. A reader antenna is worn on the user's wrist and passive RFID tags are worn on the body. Gestures are mapped among a small number of options putting the wrist near a certain tag along the body. This solution is low-cost and low power but lacks in flexibility and limits the implementation of more natural movements that should be independent from recognition devices.

A range of application integrates many small sensors in clothing [33] [34] [29][35]. Where wireless networks are not supported, embedding sensors in clothes allows a more comfortable deployment of connection cables but severely limit the flexibility of the system. In our system, flexibility and comfortable deployment is achieved through a network of tiny wireless sensors.

2.3 WiMoCA Components

In this section we describe the main hardware components of WiMoCA nodes. We also specify the architecture of the gateway node that performs bridging functions between WBASN and the external world.

2.3.1 Sensing Node Architecture

WiMoCa sensing node is designed to be wearable and low-power. It has a modular architecture to ease fast replacement and update of each component. It is composed by three sections (Fig. 2.1) namely MCU/sensors, RF and power supply. Main features of WiMoCA node are detailed in Table 2.1. In the first row we reported the sensibility of the accelerometers, which depends only on the sensing units. In the second row we reported the maximum sampling rate independently from the maximum transmission rate on the wireless channel, which is 1000 samples per second. Since each sample is composed by an acceleration value of 12 bits along each axis at 1K sampling rate, we have 36 bit samples. Fourth row reports on-board buffering capability in terms of number of samples. In the last we reported the maximum power consumption (i.e. at the maximum output data rate).

Microprocessor. The core of the WiMoCa node is the ATmega8 8-bit microcontroller based on the AVR RISC architecture [36]. The choice of this MCU has been driven by its low-cost and low-power characteristics coupled with a throughput of 1 MIPS per MHz, which is suitable to our purposes. The ATmega8 is equipped with 8K bytes of FLASH, 512 bytes of EEPROM, 1K byte of

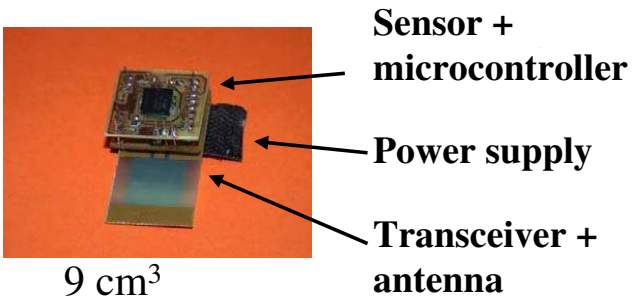


Figure 2.1: WiMoCa system structure

Sensibility	0.001g
Max sampling frequency	1000 Hz
Max throughput (bit/sec)	36000
On board sampling memory	300 samples
Power consumption 3.3V)	68 mW

Table 2.1: WiMoCa features

SRAM, a serial programmable USART, a 8-channel ADC with 10-bit accuracy, an SPI serial port, 23 general purpose I/O lines and five software selectable power saving modes. Only a few discrete components are needed around the MCU, whose values were determined by the specification of the system. In our system the MCU is set to operate with an internal 4MHz clock to reduce board area occupation.

Sensing Unit. Acceleration measurements are performed with LIS3L02DQ sensor [37], a tri-axial digital output linear accelerometer manufactured by ST-Microelectronics. The device includes a MEMS sensing element, three $\Sigma\Delta$ analog-to-digital converters (one for each axis), and a SPI serial interface from which the external MCU reads acceleration values. The inertial sensing element is manufactured using a dedicated process called THELMA (Thick Epi-Poly Layer for Microactuators and Accelerometers).

The LIS3L02DQ has a programmable full scale of 2g or 6g. It is capable of measuring accelerations with 1 mg resolution and provides a maximum sample rate of 2 KHz for each axis. The data rate can be selected according to application requirements. In addition, the device can be configured to generate an inertial wake-up/interrupt signal when a programmable acceleration threshold is exceeded along one of the three axis. The maximum power consumption is 5mW in active mode and 1 μW in sleep mode (at supply voltage of 3.3V).

RF Transceiver. The RF section is a TR1001 transceiver by RFM [38] that operates in the 868 MHz European free bandwidth. It uses OOK modulation with a maximum bit rate of 100kbps. At this rate, multipath fading effects can

be completely neglected up to distances of 50m. Since RF section is the most sensitive to noise, care must be taken when designing board layout. As a consequence, the TR1001 device and the other few discrete components are placed on the upper side of the PCB while a copper ground plate has been placed on the other side to shield the transceiver from MCU/sensors subsystem.

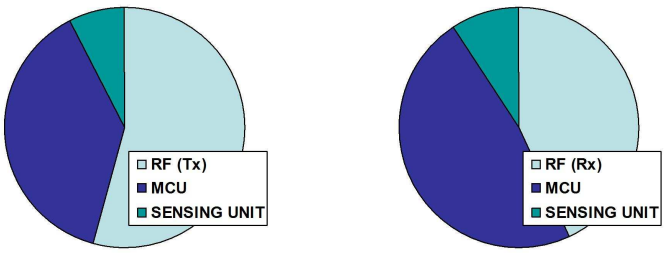
RF interface to MCU is represented by four digital signals: MCU data receive (RX) from TR1001, MCU data transmit (TX) to TR1001 and two state control signals. The RF section has three active states and one sleep state, as shown in Table 2.2. TX_OOK and TX_ASK are both transmission states characterized by OOK and ASK modulation respectively. RX is the receive state and SLEEP is a low-power state where the transceiver is neither transmitting nor receiving data.

Status	Max Data Rate	Power
RX	115.2Kbs	11mW
TX_OOK	30Kbs	36mW
TX_ASK	115.2Kbs	36mW
SLEEP	-	5 μ W

Table 2.2: States for RF section

Power Supply. The system power supply is realized with charge-pump integrated devices to stabilize voltage at 3.3V and by three NiMH rechargeable, serially connected 1.2V batteries.

The power contributions of node's components are shown in Figures 2.2 when the RF section is respectively in transmit (TX_OOK) and receive (RX) state. A complete power characterization of the node is given in Section 2.7.



a) Node is transmitting. b) Node is receiving.

Figure 2.2: Node power contributors.

2.3.2 Gateway Node

As explained later in this chapter, we designed a special node that takes care of interfacing the WBASN with an external processing unit, e.g. a PC or worksta-

tion. This node, namely a gateway, uses the same MCU and RF transceiver but it is not provided with a sensing unit. The gateway interfaces with the external processing unit (PU) through a Bluetooth [10] wireless link. As a consequence, a complete Bluetooth protocol stack has been ported on the gateway node.

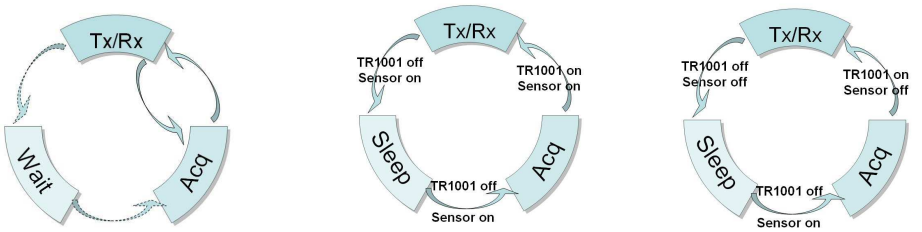
2.4 Sensing and Acquisition Software

The software support of WiMoCa nodes performs pre-processing of acceleration data and coordinate acquisition and transmission tasks initiated by an application running on a remote PC. Sensors are responsible to detect user gestures when he/she interacts with the PC. Depending on the type of movements to be recognized, each sensor can be programmed to provide the required output data rate. The user application configures the acquisition cycle through a set of APIs that interface with the software running on the gateway node. The APIs implements a device driver of the whole acquisition system. The software on the gateway is responsible for programming the sensing units involved in the recognition process, collect their data, compose the message and send it back to the application at the required rate. The APIs provide also methods to start and stop an acquisition session. For example, a data message from the gateway to the application can contain two samples coming from sensors installed on the arms and two other samples coming from sensors on the legs to detect if the user is walking, standing up or sitting down.

2.4.1 Sensor Data Conditioning

The sensing element of the wireless node provides 16 bit acceleration data for each orthogonal axis. The accelerometer can measure both dynamic acceleration (e.g. vibration and in general acceleration due to movements) and static acceleration (e.g., gravity). In steady-state conditions we can assume that gravity is the main acceleration component, thus acceleration value can be easily translated in tilt through simple trigonometric rules. Using a geometrical approach as in [19], inclination can be computed without the need of integration that introduces errors. Since accelerometers are sensitive to gravity acceleration and speed variations due to object movements, in steady-state conditions the main contribution is the gravitational one. The gravity components along the three axis of the sensor can be used to determine the orientation of the sensor and hence of the object where it is mounted.

Data received from sensor are represented in two complement in a range between 0 and $2^{16}-1$. Data are then shifted inside a $[-2g, 2g]$ range to represent both positive and negative accelerations. Since the accelerometers are used in



a) Normal mode. b) Low-Power mode case 1. c) Low-Power mode case 2.

Figure 2.3: State diagram of WiMoCa node in Normal mode (a), Low-Power mode with high acquisition rate (b) Low-Power mode with low acquisition rate (c).

steady-state conditions we expect to collect value inside a $[-1g, 1g]$ range.

2.4.2 Sensing/Transmission Cycle

The software running in the microcontroller drives the acquisition/transmission process in each WiMoCa node. It also implements a handshaking protocol needed to communicate with the gateway node. To start an acquisition session, a user application sends a start message to the microcontroller of gateway node and a stop frame to cause the system to stop the session. The start message contains also information regarding nodes involved in the acquisition session (such as the number of samples for each node and the required data rate). The gateway is then responsible for propagating the command to the interested nodes on the network.

The software running on WiMoCa node can be configured in two modes of operation. First mode is *normal mode*, where all components are always active. This mode provides the highest responsiveness but also greatest power consumption. The second mode is *low-power mode*, where the transceiver and the accelerometer are power controlled.

Normal Mode. Normal mode operations are described as a state diagram in Fig. 2.3.a. During the beginning phase, nodes are listening (RX state) for control messages (start/stop) from the gateway. Upon reception of a start command, each node involved in the acquisition session begins the acquisition/transmission cycle. During this cycle, a node switches between three states:

- **Acquisition (ACQ).** Node software samples data from sensing unit and perform data conditioning.
- **Communication (TX/RX).** Node software prepares the data packet (de-

scribed in Section 2.5) and drives the transceiver to transmit (TX) the packet towards the gateway. Before the actual transmission takes place, the node can experience a delay during which the transceiver waits for the channel to be free. The node can also receive data (RX) from the gateway during the beginning phase. Here the node also receives synchronization information required by the MAC protocol. Coordination between RX and TX as well as other MAC-level issues will be covered in Section 2.5. The node stays in the TX/RX state until the communication completes.

- **Waiting (WAIT).** After transmission/reception, the node can be idle for a while before to begin a new acquisition. The duration of this waiting state depends on the acquisition rate that can be programmed by the user.

The cycle is broken upon reception of a stop message from the gateway. The stop message triggers the end of the acquisition session.

Low-Power Mode. In the current software implementation, we support a low-power mode of operations. We focused on the transceiver and the accelerometer to reduce overall power consumption thus increasing battery lifetime. Playing with these components allows to implement a power management strategy that does not affect acquisition performance. In the next release of the software we plan to add a power/performance control strategy for the MCU. In low-power mode, described in Fig. 2.3.b the transceiver can be turned off when it is not communicating (i.e. out of TX/RX state). The time needed to turn it on and off is negligible compared to the duration of the cycle. We called *sleep state* (SLEEP) the state in which the transceiver is shut-off and the MCU is idle (e.g. WAIT state in Normal mode). As shown in Fig. 2.3.b, the transceiver is off also during ACQ state.

As regards the accelerometer, since its wake-up time is not negligible, it can be shut-off to save power only under certain conditions. If these conditions are verified, the state diagram is as depicted in Fig. 2.3.c. Here, also the accelerometer can be powered off when idle. In particular, in the SLEEP state both transceiver and the sensing device are powered off. The accelerometer is off also during TX/RX state.

Conditions to apply power control on the accelerometer are related to the acquisition rate and for this reason they depend on the user application. In fact, the model we used is characterized by a wake-up time (T_{wu}) of 50ms. As a consequence, in order to shut-off the device without impacting acquisition performance, the minimum time interval between two acquisitions must be greater than $min(T_{wu} + T_{ACQ}, T_{BET})$, where T_{ACQ} is the acquisition time and

T_{BET} is the *break-even time*. T_{BET} is defined as the time that the device should remain in low-power state to compensate the additional power spent during wake-up. In our case, T_{BET} is zero, since, as shown in Section 2.7, wake-up power is lower than active power for the accelerometer. As a consequence, we just need to compare T_{wu} with the acquisition period, which is decided by the user.

2.5 WBASN Architecture

In this section we describe the organization of the wireless body area sensor network and the Medium Access Control protocols we implemented to coordinate access to the medium among sensing nodes.

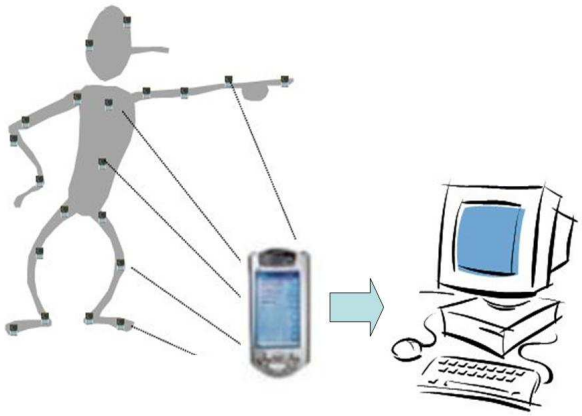


Figure 2.4: Example of body network

2.5.1 Topology and Organization

The WBASN is organized with a star topology. It can be applied to a human body as schematically shown in Fig. 2.4. WiMoCa nodes are end-points of the star and they are responsible for sensing and acquiring data from the environment and send them to a gateway node. The gateway performs bridging functions and has not sensing device. It is used to form the backbone of the system by connecting the WBASN to a host machine or to connect an external network. Coordination among transmissions of various nodes to the gateway is performed at the MAC-level.

2.5.2 MAC Layer

WiMoCa software implements a collision free real-time MAC protocol [39]. This protocol is suitable to handle real-time traffic, since it avoids the overhead imposed by collision detection packets by allocating a time frame to each message. In fact, the typical usage of the body area sensor network imposes a periodic sampling of acceleration values from a subset of nodes. For example, our system can recognize if the user is walking or sitting down through sensors placed on arms and legs. The CFRT (Collision Free Real-Time) protocol basically divides time into frames in which only one node is allowed to transmit. The scheduling order is derived by applying the EDF (Earliest Deadline First) algorithm to a message table stored in each node and identical for all the nodes, so that each of them knows when it has the right to transmit. The table contains an entry for each node allowed to transmit or receive in a frame. Fields in the entry specify source, destination nodes, message length and message period. To improve energy efficiency of WiMoCa system, we implemented a low-power version of this protocol where transceivers can sleep during time frames in which they are idle. This is possible since on-off transitions of transceivers are negligible, as detailed in the experimental results section. Compared to the standard protocol version [39] we discarded the frames dedicated to inter-cell messages since in the body network all the nodes belong to the same cell.

2.5.3 Message Format

Messages are exchanged from sensing nodes to gateway and from gateway to user application. Messages belonging to the first type are called *body messages* those belonging to the second type are called *backbone messages*.

Body Messages. Data coming from each sensor unit are called *body messages*, reflecting their collocation in the network topology. Each unit organizes payload between two marking stamps indicating the beginning (in the header) and end (in the tail) of one acquisition step. Each acquisition is identified by a sequence number, which is written in the tail. Thus, the packet sent by a single node appears as in Fig. 2.5. It is composed by a start character, two bytes representing node id, the payload, a byte for the actual frame and finally the end character. We use two bytes to identify the node. This is done for future integration in the network of different kind of sensors. In fact, the first one identifies the node, the second one may be used to distinguish among different sensing devices on the same node. Node identifier is used by applications to associate sensing data to positions along the body in which nodes are placed.

Backbone Messages. Data flowing from gateway to the processing unit (PU) where the application resides are called *Backbone messages*. The gateway

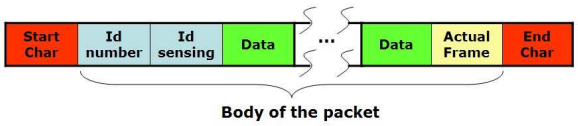


Figure 2.5: Packet sent by a single node

receives data from singles nodes (*body messages*) or from subnet of nodes. A *backbone message* may contain data from many nodes, as shown in Fig. 2.6. It is composed by a start character, a byte containing the number of remaining bytes (excluding end character), payload, end character. Payload is represented by the content of the low level frames, without start and stop marks (refer to "body of the packet" in Fig. 2.5). The link between the gateway and the PC is implemented through a wireless link (Bluetooth). As a consequence, these messages are encapsulated by Bluetooth protocol headers and footers.

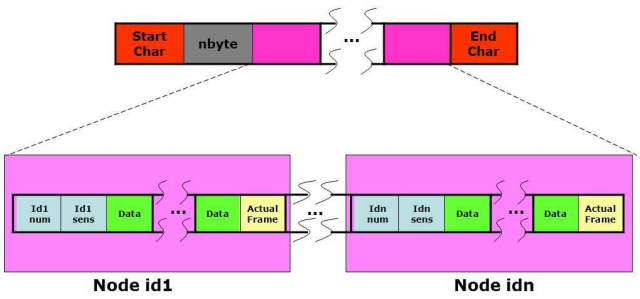


Figure 2.6: Packet sent from gateway to processing unit

2.6 Application Examples

In this section we give some example of utilization of a WiMoCa sensor node and its body area sensor network. We first describe an application that uses a single node in Fig. 2.7. The sequence of inclinations of a hand, provided by WiMoCa, can be interpreted by an application as a gesture that controls a 3D game. Mapping of inclinations to gestures is performed in this example by a software daemon running in the PU. However, depending on the type of elaboration, all or part of the processing can be made on the gateway node.

When multiple acceleration values are available, these can be used to recognize combined postures and movements. As an example, we can detect a person walking by correlating acceleration information coming from arms and legs as in [40], where a wired network of sensors was mounted on the body.

To assess the capability of the wireless body area sensor network, a test

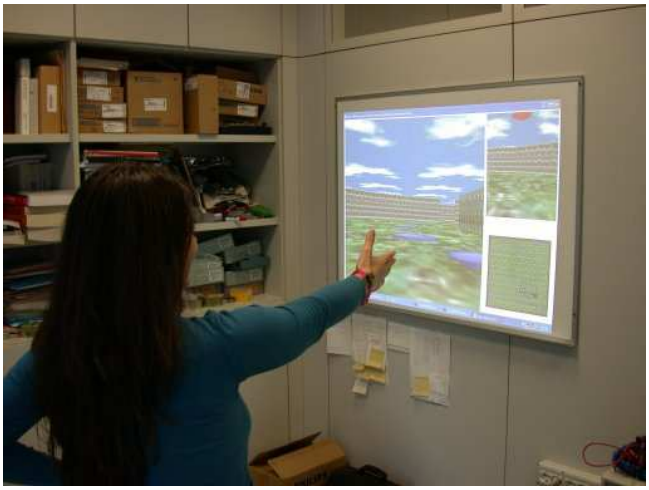


Figure 2.7: Usage of WiMoCa sensor node

application was implemented equipping a user with 3 sensing modules placed along the body, precisely on the trunk, on the shinbone and the thigh-bone as shown in Fig. 2.8. The axis relative to each module are shown in Fig. 2.9. In the leftmost part of this figure the user is represented equipped with sensor A on the trunk, sensor B on the thigh-bone and sensor C on the shinbone. If we refer to the plane defined by the direction of gravity and the ideal line between the shoulders, the projection onto this plane of the axis relative to each sensor module is shown in the right part of Fig. 2.9.

In Fig. 2.10 we show the inclinations data coming from the three accelerometers corresponding to the following sequence of movement: 1) seated; 2) standing; 3) seated; 4) seated with legs extended (as laying on a table). The plot reports computed angle degrees versus time (expressed as number of samples). Acceleration values are collected from the 3 modules shown in Fig. 2.8 and elaborated in order to obtain the angle of each axis with respect to the gravity. Thus it is possible to detect the orientation of each module and as a consequence the inclination of the three part of the body. This can be applied in a similar way to other body parts (arms, head, feet, etc.).

Figure 2.8: User equipped with 3 sensing units along the body

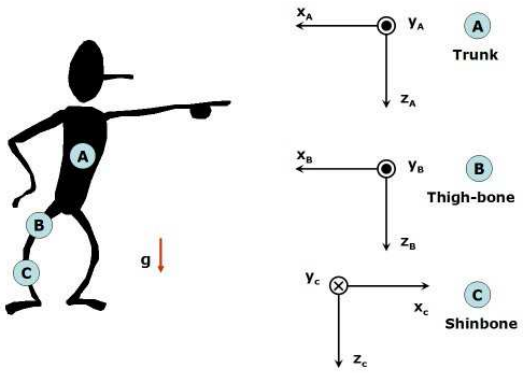


Figure 2.9: Schema of the user and the module setup and orientation along the body

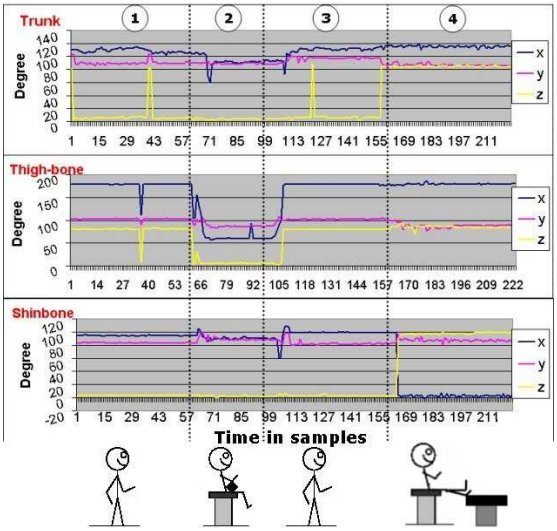


Figure 2.10: Sequence of movements

2.7 Characterization Experiments

In this section we report experiments that have been carried out to characterize performance and power consumption of WiMoCa nodes. Characterization has been performed at a speed of 9600 bps for wireless communication among nodes.

Performance. A first set of experiments have been made to characterize WiMoCa nodes both in static and dynamic conditions. Fig. 2.11 shows experimental setup for static measurements, where the device was mounted at the end of a pipe with two degrees of freedom in the plane orthogonal to the ground. Cartesian reference system is schematized in Fig. 2.12. During static

measurements, shown in Fig. 2.13.a,b and c, α was changed from 0 to 180 degree with 5 degree step. The acceleration measured (e.g. along X', Y' or Z') is the projection of the gravity acceleration along the axis. Plotted values were obtained as an average of 200 measurements. The axis of the accelerometer sensor which is parallel to the ground is Z, Y and X, for Fig. 2.13 a, b and c respectively.

Measured data are compared with expected data (xxxxx in Figures) that is equal to zero with the accelerometer parallel to the ground and has a sinusoidal behavior in other cases. It can be observed that the tri-axial accelerometer we use shows an excellent linearity for each axis and each plane.

Figure 2.11: Experimental setup

Dynamic tests were also performed to verify the response time of WiMoCa modules. Fig. 2.14 shows a 360 dynamic rotation at a rate of 120 degree/sec realized with a DC stepper motor. As for static measurements, the acceleration curve has a sinusoidal shape for the components non parallel to the ground. Signal noise was produced by the vibration of the stepper motor. This kind of noise is similar in amplitude to the one produced by rotation of a human hand and it is low enough to clearly detect the target movement.

Power Consumption. Measurement of power consumption was performed in all the states of the system (Table 2.3). To compute power consumption of each node, let us consider a sensing/transmission cycle as shown in Fig. 2.3. We consider a cycle to transmit one data packet of type shown in Fig. 2.6. If T is the time period to complete a cycle, the total energy consumption is:

$$E_T = T_{Acq} \cdot P_{Acq} + T_{TX} \cdot P_{TX} + T_{sleep} \cdot P_{sleep}$$

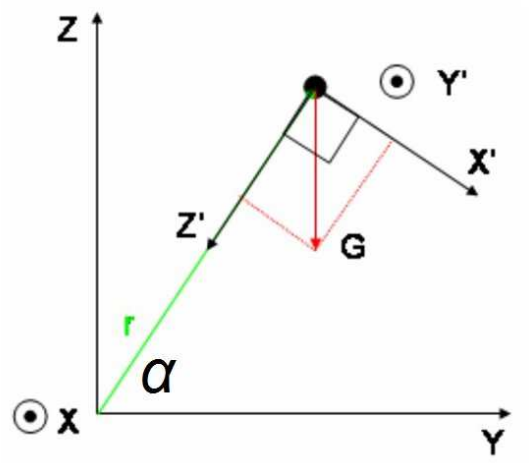
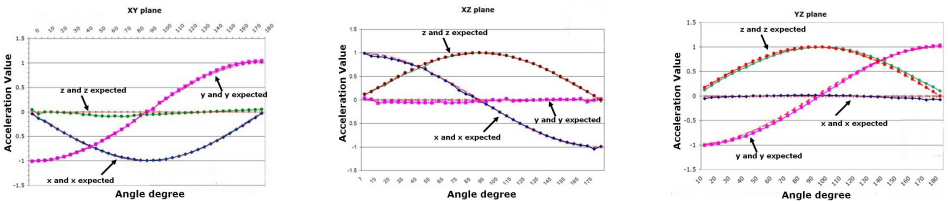


Figure 2.12: Cartesian representation



a) XY plane. b) XZ plane. c) YZ plane.

Figure 2.13: Static plot.

Where T_{Acq} and T_{TX} are constant while T_{sleep} depends on the acquisition rate. In the worst case for energy consumption (e.g. max sampling frequency) T_{sleep} is null and the energy spent in one hour is equal to 99.58mW. Considering the battery model used in the prototype (20 mAhour), WiMoCa upper bound on lifetime is 40 minutes. However, considering a more practical case, as the acquisition rate used in the recognition application described in Section 2.6 which was 30 Hz, the lifetime becomes 3 hours.

Status	Power Consumption	Time
RX state	48,10mW	2,2msec
TX state	59,40mW	21,2msec
ACQ	30,02mW	0.5msec
SLEEP	25,14mW	-

Table 2.3: Power consumption and Time requested per operation in the best case

Referring to Table 2.3, sleep time that depends on the programmed data rate, can be computed as:

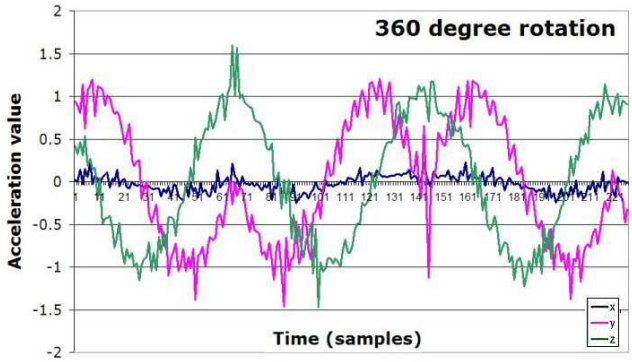


Figure 2.14: Static plot - YZ plane

$$T_{SLEEP} = T_{ACQ} - T_{TX}$$

.

In Fig. 2.15 the current is plotted during an acquisition/transmission cycle. The maximum current absorption is reached during TX mode. Transceiver and accelerometer wake-up and shut-off transitions are also shown. It can be noted that the accelerometer has a negligible wake-up current (marked with a circle in the plot) and shut-off time.

The plot in Fig. 2.16 shows the power as a function of the throughput (bit per second sent by a single node). As the throughput increases the sleep time decreases. The plot can be divided in 3 regions. A leftmost region below 2587 bps. In this region the node works as explained in Fig. 2.3.c, where the sleep time is large enough to allow both the accelerometer and the transceiver to be shut-off ($T_{sleep} < T_{wu}$=50ms). In the central region, between 2587 and 7913 bps, the node works as shown in Fig. 2.3.b. Here $T_{sleep} < T_{wu}$. It is worth to observe the step in the power consumption corresponding to transition between these regions, that arises when T_{sleep} is 50ms. The step is a consequence of the power management policy for the accelerometer. Finally, the rightmost region is a single point where T_{sleep} is null. This can be thought as a worst case when no power optimization is possible on the accelerometer because the maximum throughput is reached and data are sent continuously.

2.8 Conclusion

In this chapter we presented the design and implementation of WiMoCa, a wireless sensor node based on tri-axial integrated accelerometers, aimed at detecting human gesture and postures to implement a human computer interface

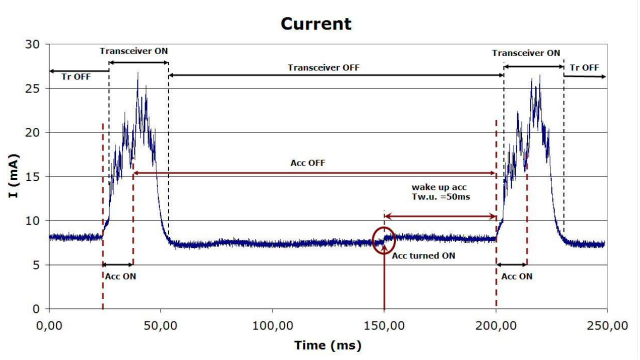


Figure 2.15: Current as a function of the node states

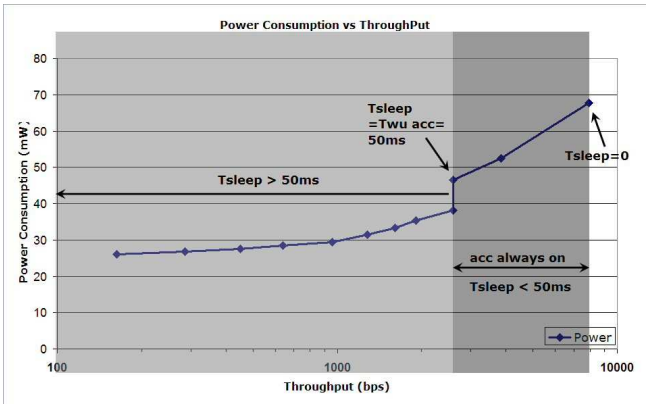


Figure 2.16: Power Consumption vs Throughput for a single node

system. We also introduce the concept of wireless body area sensor network and we present its implementation exploiting WiMoCa nodes. The sensor network enables the implementation of a distributed recognition system to detect combined body postures and movements, making a step forward traditional gesture recognition systems. We described all the software layers needed to support the acquisition, to implement the coordination and synchronization among nodes, to transfer gesture data up to the user application.

Chapter 3

Wearable system for Biomedical applications

In this chapter we describe Bio-WWS, a bio-feedback system for rehabilitation based on a dedicated wireless sensor network architecture (see Chapter 2). The sensor network is designed to be distributed on the body of the patient and to monitor its position. The hardware and software architecture (communication protocols, power management policies and application-level control) have been tuned for the application cost, battery autonomy and real-time performance requirements. Bio-WWS is an example of complete vertical integration (from hardware to application): the sensor network is fully integrated with processing and auditory feedback generation.

3.1 Introduction

Technology advances in wireless communications, light and miniaturized sensors and low power consumption have allowed the design of integrated Wireless Body Area Network (WBAN) and have suggested the use of WBAN for managing biomedical data. These developments enable applications of ubiquitous computing and embedded systems for diagnostic, monitoring, rehabilitation, and training purposes.

The potentialities of wearable technology for biomedical application are attracting a large interest in the scientific community [41], though still need to be widely exploited. In fact, the fast-paced development in sensor technology makes possible to remotely acquire and monitor a wide variety of physiological parameters, by simply equipping patients and general users with unobtrusive sensors. Measurements may include heart rate [42], respiratory rate and blood pressure, blood oxygen saturation, pulse waveform, physical activity (see [41]

for a detailed review of these systems), and motion tracking [43] in neurological and orthopedic diseases [44]. Applications vary from high-risk patients monitoring after surgery, to geriatric care [45]; and may include athletes performance monitoring, for example in scuba diving, mountaineering, and hiking [46]. Many of these applications aims at the early detection of abnormal states and the prevention of severe consequences.

Applications designed for impaired or elderly people often not only need monitoring, but also support through biofeedback to partially compensate the disability (prosthetic aim) or to restore the normal physical condition (rehabilitative aim).
An interesting area where provisioning of biofeedback coupled with monitoring may offer a great contribution is the maintenance of human balance control. In fact, postural unbalances are a crucial aspect in many musculoskeletal and neurological diseases, in chronic impairments of the motor system due to aging, but also in the medical research or in several sport disciplines [47]. Balance, under normal conditions, is the result of the contribution of three main sensory systems (visual, vestibular, somato-sensory), of sensory integration provided by the central nervous system, and of the action ensured by the musculoskeletal system. Thus, any impairment or limitation in the aforementioned physiological systems may result in the alteration of balance and posture. Furthermore, in several sport disciplines enhancement or restoration of balance and posture control after injuries is an important goal for athletes training and rehabilitation. Experimental studies [48] indicate balance training to be effective for gaining in muscular strength and, also, for the equalization of muscular imbalances. Finally, the biofeedback-assisted manipulation of balance control may be a relevant issue in studies about the physiological adaptations and conditioning that take place in weightlessness, which may heavily affect astronauts' health after returning from space flights [49].

Inertial sensors, especially accelerometers, are very suitable for posture detection applications because of their small dimensions and lightweight portability and the useful kinematic information they supply. In fact, the use of accelerometers in motor control or activity recognition is not a novelty. In [50] biaxial accelerometers are used to distinguish between static and dynamic activities and detect basic postures of sitting, standing and lying.

Usually, biofeedback systems in the field of postural control are cumbersome and for ambulatory use. They typically need expert operators, thus limiting their use to the place in which the infrastructure is mounted and to the availability of medical assistance.
In this scenario, the present work proposes a Biofeedback Wireless Wearable System (Bio-WWS), i.e. a specific biomedical application based on WBAN for

human motion tracking and on audio biofeedback restitution for optimization of posture and movement. Its application fields range from movement rehabilitation after a damage of the motor system, to aid in the case of a sensory system deficit or application in sports training. In the present work we will particularly emphasize the potential usefulness of Bio-WWS in the monitoring and control of human posture in quiet standing, using wireless and wearable accelerometer nodes and providing a biofeedback through an audio signal to improve or correct a user's balance. The use of the audio biofeedback was proven to be effective for balance improvement in patients with bilateral vestibular loss, and to significantly change structural properties of the postural sway in quiet standing [51]. In the design of the system's architecture, particular attention was paid to modularity, to allow large flexibility in the type and number of input sensors and, consequently, in the category and number of biomedical applications. In addition, we designed the system with off–the–shelf components, to benefit of low-cost design optimization, performance and ergonomics tuning typical of consumer market products, at the same time preserving the possibility of a large diffusion. Our system also avoids limitations due to wires among sensors and/or actuators and between single nodes and processing unit or the WBAN and the data collection server. Thus, Bio-WWS is a WBAN [52] integrating both sensors and actuators for long term monitoring and biofeedback during normal activity or intensive training and not only for assisted rehabilitation. The integration of a palmtop computer as a special, yet wearable and wireless, general purpose node of the WBAN enables the implementation of special algorithms for biofeedback, thus eliminating the need to use the system only in ambulatory condition. The audio biofeedback chosen for Bio-WWS is provided through lightweight wireless headphones.

In this chapter, after the description of each component of the system, the specific implementation of the general-purpose Bio-WWS architecture will be reported, and its biomedical potentialities discussed.

3.2 System Architecture

Bio-WWS is a mobile and WBAN for providing audio feedback as aid for elderly people or for people needing posture rehabilitation, sportive training and vestibular loss compensation. The most general architecture for such systems is the one described in Fig. 3.1. In the left side within the square a schematic idea of Bio-WWS architecture is provided. At the right side possible remote addressees for data coming through the network (via GPRS or other wireless interfaces) from Bio-WWS are in evidence, such as domotic servers, hospitals and call centers.

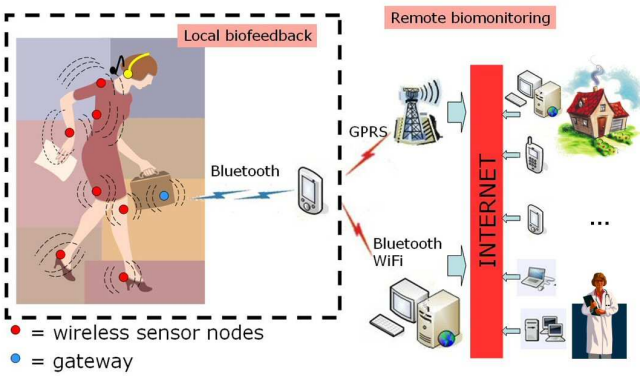


Figure 3.1: Potential context of the system integration

Bio-WWS is thought to be used not only in medical assisted environments, but also and particularly in daily activity at home, work and out-hospital environments. As a consequence a main requirement for the system is the ability to provide to user all the needed bio-feedback to be self-sufficient, independent from other people intervention. Thus, even if the general architecture is open to be connected with a remote server for medical assistance provisioning, the primary use of Bio-WWS is helping the user to self-correcting her/his posture both in static and dynamic conditions.

This chapter focuses on description of Bio-WWS used stand-alone, thus the target is not sending data to remote servers but providing actuation consistent with sensory data collection for a certain purpose.
Other requirements of the system are:

- Wearability and unobtrusiveness, thus needing light and small-sized components;
- Low-cost (affordable for consumer market);
- Low-power consumption (supporting daily continuous use);
- Flexibility and easy integration of other sensors/actuators;
- Easy maintenance and components update;
- Mobility for everywhere use;

Bio-WWS is composed of four kinds of node, all wireless and wearable: one or more sensor nodes, the audio actuator node, the PDA, which acts as a general purpose node, and the gateway node. At present the system has been tested with three sensor nodes, one located on the trunk for sway monitoring and the other two on a leg (thigh and calf) for distinguishing if the user is

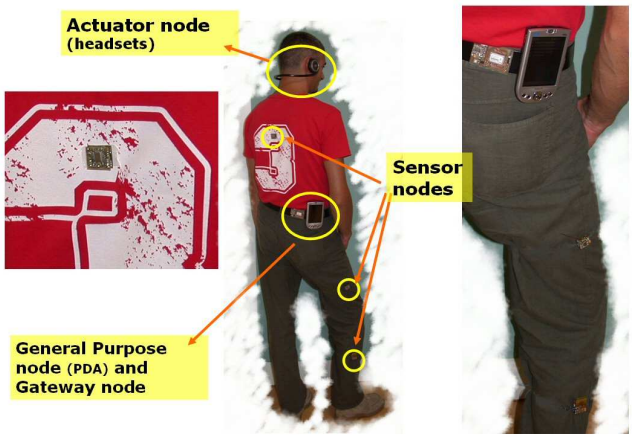


Figure 3.2: Bio-WWS set up

walking or standing. As actuator we tested commercial Bluetooth headsets (Fig. 3.2).

Each node collects data from the sensing unit, i.e. a digital tri-axial accelerometer that is plugged on. For a simple WBAN, as the one presented here, a star topology between nodes and gateway is sufficient. The gateway performs a signal pre-processing, data normalization and organizes data retrieved from the sensor nodes in packets to forward them to the handheld device over a Bluetooth link. At last the PDA can process the body data in order to resolve the human's posture condition and to perform a proportionate feedback to the user. In this system we use a sound that is dynamically modulated by.

The following sections will describe each part of the system both from a hardware and software perspective.

3.3 Hardware

The system is entirely composed with devices and materials available on the market. Particular attention was made to find the best trade-off between performance and cost of the system, since our target is a low-cost system that can be used by a number of people as large as possible.

3.3.1 Sensor nodes

The node adopts a modular architecture (Fig. 2.1) and it is the one presented in Chapter 2. It is organized in layers corresponding to functional blocks (Fig. 3.2) for reduction in node size and flexible reuse, upgrade and maintenance. For example, the existing transceiver working at 868 MHz can be easily replaced

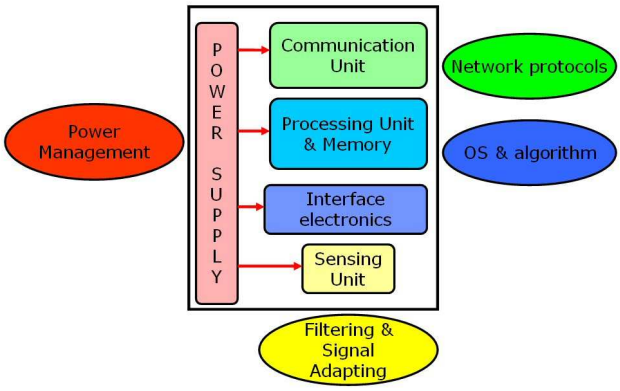


Figure 3.3: Architecture of the sensor node

with a different one (e.g. working at 2.4 GHz) only substituting the RF layer, not affecting the whole module re-design.

The main layer is equipped with the microcontroller. It is an ATMEL ATmega8, a low-cost, low-size 8-bit microcontroller based on the AVR RISC architecture, providing a good trade-off between performance (1 MIPS per MHz) and power consumption. It integrates an internal 8-channel A/D converter for analog sensors, 8KByte of FLASH for the firmware, 512 bytes of EEPROM and 1Kbyte of SRAM that are enough capable to store not only temporary sensor information, but also parameters and constants for a MAC protocol used by wireless communication. In order to reduce the area occupation, the microcontroller unit (MCU) can operate with an internal clock that is set to 4MHz. In the same layer is present also the sensing unit. It is a tri-axial digital output linear accelerometer manufactured by ST-microelectronics. It includes three MEMS sensing elements (one for each axis that are able to measures a range of [-2g, +2g]), internal A/D converters and a SPI serial interface for data exchange to MCU.

The RF section is placed on top of the node. It is currently composed by a RFM TR1001 transceiver and a small antenna that works in the 868 MHz European free bandwidth. Only few components are needed around the RF module, determined by the specification of the system. The RF communication can use both OOK and ASK modulation getting a maximum bit-rate of 115Kbps.

The power consumption of this device is very low. We have measured a 19 mW in receive state (RX) and 26mW in transmission (TX) using an ASK modulation, confirming that using current Bluetooth modules available on the market is not convenient because they are more energy hungry. The main disadvantage of TR1001 is the absence of protocol firmware for the access to the

medium (MAC). Thus we supported the node with a proprietary MAC implementation for setting up a wireless sensor network.
The node is supplied by a small series of NiMH rechargeable batteries. In order to obtain a stabilized voltage at 3.3V for the whole node, the power layer is also equipped with a charge-pump integrated device.

3.3.2 Gateway

The gateway (Fig. 3.4) is a special node, without sensing element, acting as a bridge between the Sensor Nodes applied on the Body (BSN) and the mobile devices. Conceptually each sensor node can act as gateway, if equipped with a Bluetooth transceiver or another interface to connect to a general purpose device. Not needing sensing capabilities, in the present implementation the gateway is very small, non-intrusive, and the user can place it wherever he wants (e.g. in a trousers pocket).
We explored at first the idea to design a dedicated board for the handheld device (e.g. exploiting the PCMCIA connector). Unfortunately, this kind of solution means the development of a proper device driver for any kind of platform and operating system that could be used. Instead, providing the BSN of a Bluetooth access port enhances portability and deployment, because it is easier to install a Bluetooth service on other systems.
The module has a stackable design as the sensor node, hence uses both the RF and the MCU layer designed for the sensor nodes. The Bluetooth layer is equipped with a module manufactured by SmartM, called Bluetopaz, to enable communication with the handheld device. It is a Bluetooth certified device with HCI and Serial Port Profile implemented into the firmware. It is a class 2 device, and it consumes 0,9mW when it is in sleep mode and requires maximum 150mW during the transmission. The interface to MCU is provided by two digital signals (TX and RX) implementing an UART serial protocol, consequently it is easy for ATmega8 MCU to retrieve and forward data.

The use of the gateway as a middle stage was necessary to achieve requirements of low power consumptions and real-time performances. In fact, current handheld devices interface with other system through cables (USB, RS232) or wireless protocols (Bluetooth, IRDA, IEEE802.11). Bluetooth connection results the best choice because it does not require line of sight between system and device, and makes the bio-feedback system more wearable and comfortable avoiding cables. Although it consumes less power than other wireless connections provided on a PDA, unfortunately it is not adequate to support BSN power requirements. Since a BSN usually manages several nodes, use of Bluetooth on each node should have been more energy expensive. Recent

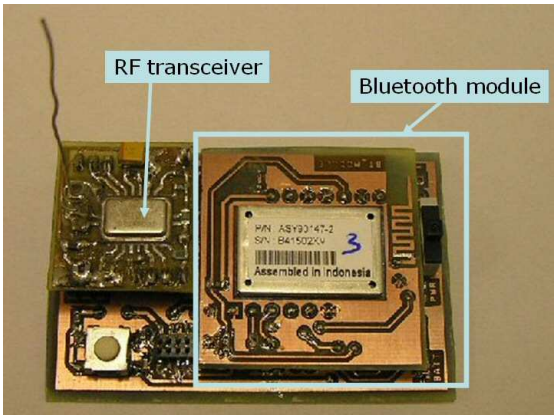


Figure 3.4: Gateway node

works [53] proved that a large wireless sensor network based only on Bluetooth protocol is not convenient, due to the consequent high energy consumption. Moreover Bluetooth enables up to eight devices to form a piconet. Deploying larger network (*scatternet*) is however possible, but it means multi-hop message delivering, with the consequence of decreasing significantly performance in bandwidth and time. The cost of the system is another requirement that justifies the use of another wireless protocol for the BSN (Bluetooth module cost is around 20$, while the price for low power transceiver such as RFM TR1001 and Chipcon CC2400 falls to few dollars).

3.3.3 Handheld Device and bio-feedback actuator

A general purpose node in our WBAN executes the bio-feedback algorithm based on sway detection around the user's center of mass. We chose the HP iPAQ 5550 with Bluetooth and WiFi capabilities. It is equipped with an Intel XScale 400 MHz, 32+32 KB data and instruction caches, and 128 MB SDRAM. We use Microsoft Pocket PC 2003 as operating system whereas the Bluetooth protocol stack is provided by Widcomm.
To reach a fully wearable system, we adopted also a wireless link through PDA and the headphone using a pair of Bluetooth enabled headsets. The Hewlett Packard Bluetooth Stereo Headphone acts as actuator node to provide the audio bio-feedback. Bluetooth uses synchronous and reliable connection (SCO) to perform an audio link, and all the procedure is managed by the operating system and the Bluetooth stack present on the platform.

Figure 3.5: Software tasks

3.4 Software

The software architecture is organized in four main sections, which are responsible for data acquisition from the BSN, processing and management of the bio-feedback to the user. At last, a section of software is developed to have an efficient network protocol among the sensor nodes and the gateway, because, as we have already explained, the low-power transceivers we have used lacks of a MAC protocol.
The part of software that runs on the handhelds device is developed entirely in C/C++ using the API of the SDK for Pocket PC 2003. The handheld device is adequate to run the algorithms of posture detection and sound generation, which are computationally demanding tasks. Furthermore, a graphical user interface (GUI) helps in set up and adjustment of main parameters (e.g. Bluetooth port parameters, sound biofeedback frequency, volume range).
The firmware that is downloaded into each node and into the gateway is developed using the AVR assembler for ATMega platforms. Data acquisition is the main task of a node, whereas the gateway module is committed for preprocessing. Fig. 3.5 summarizes tasks distribution along the whole system.

3.4.1 Data acquisition

As first release of the system, we have adopted only a tri-axial sensor accelerometer as sensing unit for all the nodes, because an accurate positioning of the sensor is coarsely enough to determine an alert on incorrect posture. In future we plan to integrate this information with other data, such as the pressure at the plants of the feet or a dynamic movement monitoring.
As we have described formerly, the output data of the accelerometer is already converted in digital form and represented in two complement in a range between 0 and 216. Since in steady state conditions, the main contribution is the gravitational acceleration, a geometrical approach for measuring tilt is used [54]. Currently, data are sampled at 16 KHz, but the system can provide a higher data rate and the firmware is arranged to share MCU resources between other sensors.

3.4.2 Communication

We have developed a simple MAC protocol for the communication among wireless sensor nodes and gateway. A typical non obtrusive WBAN does not exceed a maximum of a dozen of nodes placed on the user body, each node working at data rate ranging from a few Hz to 30/60 Hz. Therefore the data rate of the transmitted information is not critical. As a consequence, a real-time collision free protocol is adequate for the purpose. The MAC protocol chosen can be enhanced with power management features in order to save the energy consumption and to extend the life of the WBAN.
Assigning a time-slot for each sensor node, through a schedule based on identical tables which reside on each node, the gateway collects data from the sensor network. Therefore the first technique of power-saving is to shut-down the transmitter module when the current time-slot is not dedicated to the node. In the meantime the node can continue to sample and storage in local memory the acquired data from the sensing unit. Obviously a similar schedule can be performed only when the network has small size and the nodes are not far from each other, because the greater effort is to synchronize all the nodes to the same timing. We get through this problem, adopting a phase of synchronization between all nodes. Since this application has an intrinsic star topology, where the gateway module is the centre of the star, we have assigned a time slot to the gateway for the network synchronization as well. In this special slot, all sensor nodes switch in receiving mode and wait for the synchronization data. It is a useful solution also because the gateway can exploit this phase to send control command to the network (e.g. modifying the scheduling priority, change the type of desired data).

3.4.3 Processing

The system performs a processing of all the data coming from the body and determines user sway with respect to user center of mass. A first step of processing is performed by the gateway. Its task is to collect all the data retrieved by the BSN. Since there is a time constraint due to the human feedback, all the data are grouped by a timestamp in order to have a screenshot of all the sensor placed in the human body at a given time. Moreover the gateway provides to normalize all the information in a single range of 16-bit data. This operation is not a requirement, but we have noticed that it is very useful if the nodes are equipped with different type of sensing unit, in fact we can adopt an unique data gathering interface on the handheld devices and there is no need of further conversions. The grouped data are then sent over Bluetooth link to the handhelds device. We have chosen a transceiver with a serial port profile

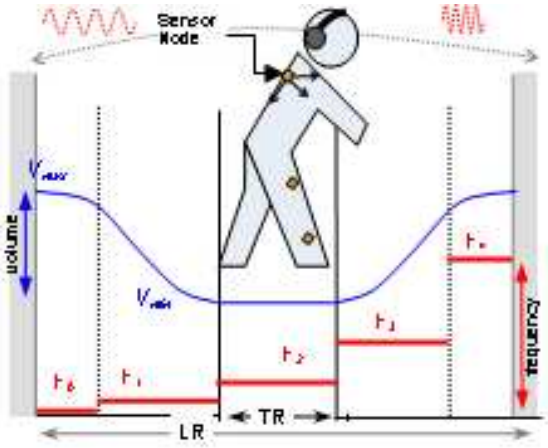


Figure 3.6: Audio biofeedback generation

(SPP) thus the connection to the gateway is mapped as a simple serial connection. At application side, it is easy for a developer receiving data using the serial API provided by the Pocket PC 2003 SDK.

3.4.4 Audio biofeedback generation

The audio generation is based primarily on information provided by the sensor node positioned on the trunk. The movements of the trunk along the forward-backward and lateral directions are coded through the modulation of sound that provides the user with the proper spatial information and with cues for how to correct, when necessary, the body position. The algorithm that generates the audio biofeedback moves from previous studies [55] that verified the basic criteria for audio biofeedback restitution, in order to achieve an high comfort level for the user, and high efficacy in the restoration of the appropriate postural position.
The user's movements in the forward-backward direction are coded by frequency and amplitude modulation of the sound (Figure 7); the frequency informs about the absolute position, the amplitude about its "dangerousness", i.e. about the distance from the target region (TR) within which the user should remain and where the volume and frequency remain unchanged.

Every 50 ms the position sensed by the trunk node is updated, and consequently also the biofeedback, following the specific rules represented in Figure 7. Frequency modulation is obtained by means of a discrete position mapping with five frequency values: F_k=F^k, with k=0:4, F_0=100 Hz (Figure 7), resulting in a TR frequency of 1000 Hz. These values were chosen so that users may have a comfortable sound perception and good sensitivity to sound changes [55] . Volume modulation follows a sigmodial law, and it is symmetrical to for-

ward and backward directions. Since balance is usually more critical in the backward-forward than to lateral direction we coded lateral movements by different volume balance to the right and left channel of the stereo headphones, as if the user would approach a (virtual) source of sound.
To date, the sensors on the leg are included in the biofeedback code as quiet standing on/off sensors. Imminent developments of the system will include also the leg sensor nodes, to provide an audio biofeedback during movements (in particular during walking). To this aim further sensor nodes will be added (on the other leg) and 3D accelerations will be considered. Indeed vertical acceleration from the trunk was not included in the up-to-date biofeedback generation, since it is negligible acceleration in quiet standing (but not in walking).
The dynamic of the sound generation is determined by the TR and by the Limit Region (LR) indicating the user stability limit. TR is the region within the user need to remain, and it reflects ballistic-like movements typical of the postural control system; it is subject-specific and is set in the first few seconds (5 s) of functioning of the system (calibration time); LR may be set consequently. Previous results and experimental sessions allowed us to set TR as 1.25 the standard deviation of the acceleration in the calibration time, and the LR to 10 times the standard deviation of the same value. The regions were computed both for the backward-forward direction and for the lateral direction, considering the correspondent accelerations.
The TR (and consequently the LR) may be changed according to the specific use of Bio-WWS. These limits may be smaller for applications that aim at enhancing balance control (in sport discipline such as dance, yoga, martial arts, gymnastic), or larger in case the standard limits would be inappropriate in severe damage of the postural system (as Parkinson's disease).

3.5 Discussion

3.5.1 Mobility, Usability and Costs

The small form factor both of the sensor nodes (size 20x20x18 mm) and the gateway (4x6x1.5cm, battery included) is suitable for easily wearing the system without experiencing obtrusiveness. Moreover, for the range of applications here presented for WBAN a small number of nodes is needed. Palmtop computers are decreasing their size while increasing computing capabilities (e.g. HP IPAQ example size is 7,5 x 1,9 x 11,9 cm) and they can easily be worn in pocket or carried in bags as it happens for mobile phones.

From a cost perspective, the proposed solution is extremely interesting, considering that the prototype node has a cost of 35€, the gateway of 50€ not

	ON	IDLE	SLEEP
Gateway	200 mW	25 mW	1,5 mW
Node	45 mW	20 mW	10 mW

Table 3.1: Power consumption comparison

including batteries. These costs can decrease significantly, since they are computed for small volume prototyping. Bluetooth headsets cost around 80€, but they can be easily replaced with cheapest wired one. For the application tested the WBAN, using low-cost headsets, has a cost ranging around 160€ (excluding the palmtop computer). Our system architecture is suitable for consumer market and widespread diffusion, e.g. as a potential partial substitute of expensive optical motion tracking systems (initial cost can easily reach 50,000€), which do not allow out-door acquisition and require high-cost maintenance during their lifetime.

3.5.2 Power consumption

Measurement of power consumption was performed in all the states of the system (Table 3.1), setting sampling frequency at 60Hz (to obtain a useful bandwidth of 30Hz, adequate for capturing human movements). Communication speed was set at 32kbps from node to gateway on OOK transmission at 868MHz, and at 230kbps from gateway to PDA, transmitting using Bluetooth at 2,4GHz. Power consumption is shown both for the gateway and the node, in three different states. The worst case indicated with "On" in table, corresponds to the case where all components are active and transceiver is continuously sending data. In this condition, gateway battery (500mAh) and single node battery (100mAh) have a lifetime of 8 hours. "Idle" corresponds to gateway having all components in idleness and not transmission happening, for the node only reception mode is active. Sleep mode corresponds to having all devices in the minor consumption state. For a 20% duty cycle the lifetime reaches 40 hours for the gateway and 38 hours for the node. This is a possible real situation, because typically it is not needed or it is not possible (e.g. the channel must be shared with other nodes) to send data continuously. The major limitation can be the palmtop computer lifetime. We tested the IPAQ transmitting a continuous data stream to a desktop PC through Bluetooth link. The evaluated duration of the battery in this worst case is of 12 hours.
Thus we can conclude that BAN lifetime is adequate for the applications we have in mind.

3.6 conclusion

In this chapter we present a low-cost and low power system based on a WBAN for provisioning of postural control audio biofeedback in daily life. Bio-WWS effectiveness is based on the ability to support the user at any time in any place without the need of expert assistance or ambulatory infrastructure. At the same time, by the integration with a general purpose computer such as a palmtop pc, provided with more than one wireless interface for networking, the system is open to be connected for remote supervision.
The user's comfort during the Bio-WWS utilization, mainly due to the wireless and miniaturized technology, is a promising characteristic for potential large diffusion and use. In fact, the flexibility of the Bio-WWS design offers the chance for application in more than one context, through the integration of many different sensors.

Chapter 4

Introduction to Energy Scavenging

The operating lifetime of wireless sensor networks, as for many other battery-operated embedded systems, is a crucial design parameter. Electronic systems continue to shrink and less energy is storable on-board. Although research continues to develop higher energy-density batteries and supercapacitors, and a few very low power wireless sensor platforms have recently entered the marketplace, the amount of available energy still severely limits the system's lifespan. One of the main advantages of sensor networks connected via wireless technology is their ability to operate in the absence of a pre-established infrastructure. This independence, however, severely depends on the autonomy of the single nodes. In order for wireless sensor networks to become a ubiquitous part of our environment, alternative power sources must be employed and devices that extract energy from their surroundings in some way (so called energy scavenging devices) have attracted attention from many researchers: if nodes are equipped with energy transducers the generated energy may increase the autonomy of the nodes significantly. The main purpose of this chapter is to overview the state of the art in energy harvesting techniques. In the following sections, each harvesting technology currently in use or just proposed as theoretical study will be investigated. We will describe briefly also some academic prototypes or product already available on the market.

4.1 Potential Sources

A possible classification of the methods of providing power from non conventional storage devices could be summarize as follows:

Source	Power Density	
Solar	1 100	$\mu W/cm^2$
Vibration Capacitive	100	$\mu W/cm^3$
Vibration Inductive	10 - 15	$\mu W/cm^3$
Vibration Piezoelectric	300 - 500	$\mu W/cm^3$
Thermoelectric	6 - 15	$\mu W/cm^3$
Radioisotope	100 - 250	$\mu W/cm^3$
High frequency vibration	100 - 150	$\mu W/cm^3$
Ambient radio frequency	< 1	$\mu W/cm^2$
Vibrational microgenerators	800 (machineskHz)	$\mu W/cm^3$
Ambient airflow	1	mW/cm^2
Biofuel cell	3,9	$\mu W/cm^2$

Table 4.1: Power sources comparison

- Harvesting available ambient power to the node (i.e. a solar cell), which implies a physical conversion from a type of energy to electrical energy;

- Distributing power to systems, using methods alternative to wires that not necessarily involve a conversion of ambient energy.

A direct comparison of vastly different types of power source technologies is difficult (for example, comparing the efficiency of a solar cell to that of a windmill is not very useful). For this reason the following table (Table 4.1) will consider only the power density, a metric that emphasizes the two main features of an energy harvester for embedded device: the power delivered and the minimum size required for it.

4.1.1 Thermoelectric conversion

Temperature differences can be use to provide electricity (Fig. 4.1). Thermoelectric conversion has been mainly proposed for wearable electronics systems, even if the power available from this source would vary as a person walked from an indoor, heated environment to a cold winter day outdoors.

Thermal gradients in the environment are directly converted to electrical energy through the Seebeck effect. Temperature differentials between opposite segments of a conducting material result in heat flow and, consequently, charge flow, since mobile, high-energy carriers diffuse from high- to low-concentration regions. The generated voltage and power are proportional to the temperature gradient between the hot and cold junctions and the Seebeck coefficient of the thermoelectric materials. Large thermal gradients are essential to produce practical voltage and power levels because of the Carnot cycle that fixes the fundamental limit to the energy obtained from a temperature difference. For

example, a a wearable thermopile placed on human skin yields only 5.5%, if the gradient goes from body temperature (37 degrees Celsius) to a cool room (20 degrees Celsius).

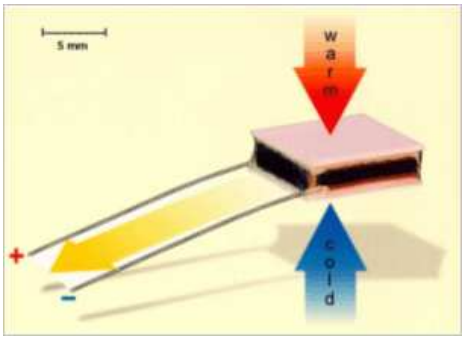


Figure 4.1: schematic of a thermopile

Commercial products, that exploit thermoelectric conversion, have been introduced. For example, the Seiko Thermic Wristwatch (Fig. 4.2) absorbs body heat through the back of the watch and uses 10 thermoelectric modules to generate sufficient power to run the clock engine. Thermo Life ® (Fig. 4.2), a thermopile from Applied Digital Solutions (ADS), is a thermoelectric generator measuring 0.5 cm2 in area by 1.6 millimeter thickness. It can generate 10 μA at 3 V (6 V open circuit) with only 5 degrees Celsius of temperature difference [56]. The energy generated is stored in a NanoEnergy battery, a thin-film storage device developed by Front Edge Technology. ADS claims that the possible applications are varied: attachable medical devices, electronic wrist watches, self powered heat sensors, and mobile electronics. Stark et al. [57], before designing Thermo Life, proposed also a Low Power Thermoelectric Generator for the D.T.S. company. It is a small compact thermoelectric generator whose output is compatible to the requirements of micro electronic and micro matched system loads. Some authors have been working recently in thermoelectric technology to generate electrical energy from the waste heat of microprocessors to drive a cooling fan. This approach allows to separate thermal solution of electronic equipment from battery power which is an attractive idea for portable applications [58].

4.1.2 Photovoltaic

Photovoltaic cells have been used to power electronic systems, converting solar energy into electricity, for decades. Each cell consists of a reverse-biased pn^+ junction, where light interfaces with the heavily doped and narrow n^+ region. Photons are absorbed within the depletion region, generating electron-hole pairs. The built-in electric field of the junction immediately separates each

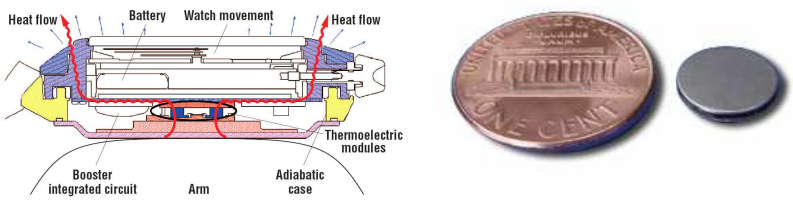


Figure 4.2: Seiko Thermic Wristwatch and Thermo Life® module

pair, accumulating electrons and holes in the n+ and p-regions, respectively, and establishing in the process an open-circuit voltage. With a load connected, accumulated electrons travel through the load and recombine with holes at the p-side, generating a photocurrent that is directly proportional to light intensity and independent of cell voltage.

Photovoltaic systems spans from the megawatt to the milliwatt range producing electricity for a wide range of applications: from low-power consumer electronics such as wristwatch and pocket calculator to grid-connected PV systems. The application of photovoltaics in portable products could be a valid option under the appropriate circumstances. Light intensity varies dramatically due to the 24 hour solar cycle and cloud cover for an outdoor sensor network. Solar radiation varies over the earth's surface due to the weather conditions and the location(longitude and latitude). For each location exists an optimum inclination angle and orientation of the PV solar cells in order to obtain the maximum radiation over the surface of the solar cell. However the gap between indoor irradiance and outdoor lighting still remains large, ranging from a few mW/cm^2 for indoor cell to hundreds of mW/cm^2 for outdoor solutions.

One problem that must be solved in harvesting efficiency is maximum power point tracking (*MPPT*), or impedance matching between the supply and the source at runtime. The impedance of a solar panel is primarily a function of the sunlight intensity and the current, and to a lesser extent of temperature and other factors. The maximum power point (*MPP*) is the point on the I-V curve that maximizes the power output at the given level of light intensity (Fig. 4.3). MPPT entails sensing the relevant supply condition and setting the current limit accordingly. If for wide scale photovoltaic systems the problem of an efficient *MPPT* is easily solved because the power required by the controller is available, for small solar harvester it is still a problem, because the trade-off between the energy gained using an optimal *MPP* tracker, and the energy used by the controller.

Due to the experience accumulated in domestic and civil system, there are

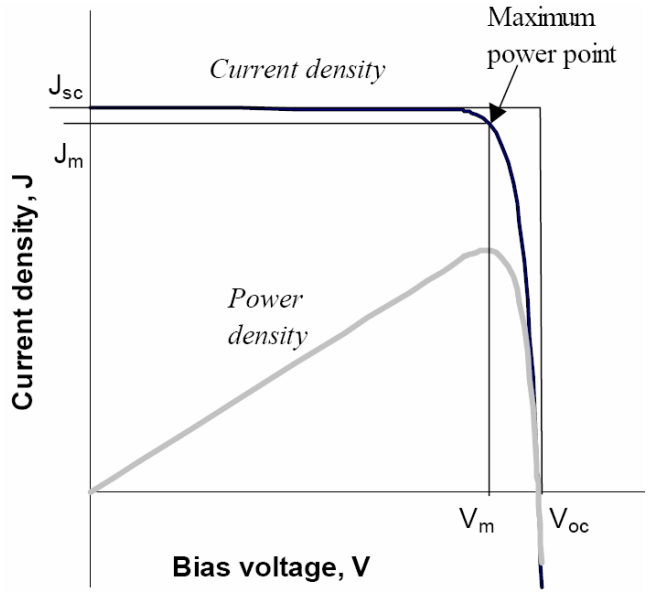


Figure 4.3: Maximum power point of a photovoltaic module

several academic prototype that use photovoltaic cells for embedded system and recently some solution has been proposed also for wireless sensor node: Prometheus, Ambimax (Fig. 4.4) and Everlast are the most recent prototype that supply a sensor node with solar cell of few cm^2 [59, 60, 61]. Veefking [62] presented an interesting work about industrial design and photovoltaic power that takes into account an energy balance of different consumer products to ensure that solar energy is a valid source. He presented Solar Tergo (Fig. 4.4), a charger for small portable products such as mobile telephones and MP3 players for use in combination with a backpack.

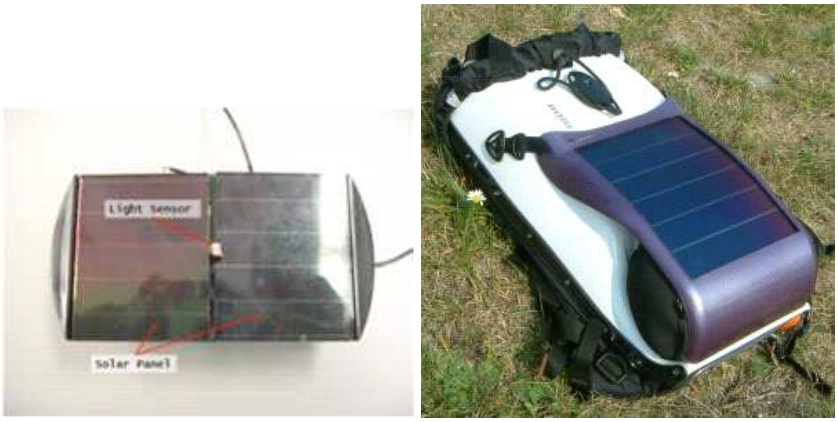


Figure 4.4: Ambimax and Solar Tergo, solar powered embedded systems

New technology of solar cells provides cheap and flexible modules. Gen-

eral Electric, for example, is producing strips of flexible plastic that are converting the light into electricity. Advances materials science, including nanomaterials, is the base of printable solar cells. Nanosolar is developing the idea of spraying nano solar cells onto almost any surface. This technology could enable Nanosolar to spray-paint photovoltaics onto building tiles, vehicles,... and wire them up to electrodes. Nanosolar predicts that by the end of 2005 will have prototypes that capture 10% of incoming solar energy [63].

4.1.3 RF induction

Since the number of radio transmitters scattered throughout today's urban landscape is increasing, exploiting background radio signals as a mobile-device power reservoir could be an attractive idea. Electronic systems that harvest energy from ambient-radiation sources, however, have extremely limited power and generally require either a large collection area or close proximity to the radiating source. The approach, heralding Nicola Tesla a century ago, involves broadcasting RF energy to power remote devices. This practice is now commonplace in passive radio-frequency identification systems RFID (Fig. 4.5) which derive their energy inductively, capacitively, or radiatively from the tag reader using a rectenna. A rectenna is a rectifying antenna that is employed in order to directly convert microwave energy into electrical energy. The simplest rectenna can be constructed from a schottky diode located between antenna dipoles. The diode rectifies the current induced in the antenna by the microwaves. Employing the RF radiation, as way of distributing power to embedded electronics, works well if there is a source very near to the electronic device. Moreover law regulations limit the power radiated to a values that allows to receive a strength around 50 μW from a transmitter 5 meters distant.

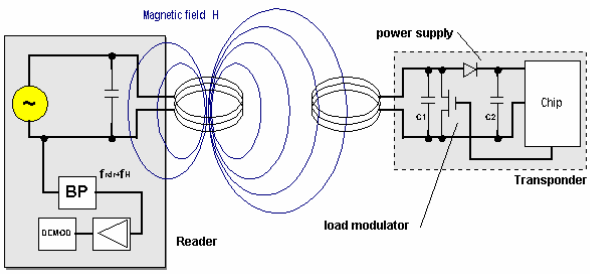


Figure 4.5: schematic of a RFID system (Reader + Tag)

The motivation because RF induction is mainly used by RFID systems resides on such devices can exploit RF very efficiently, using Radio backscatter techniques (Fig. 4.6) which allow to consume very small power on the RFID Tag during radio communication. Radio backscattering is often selected over

more conventional wireless transceiver designs because the sensor electronics are inherently less complex. In a radio backscatter sensor system, the system interrogator or reader provides the radio carrier that enables wireless communications and the power supply on the RFID Tag sensor. The sensor merely modulates its antenna reflecting an AM sideband with a digital message that is encoded with a unique identification code and data payload. In other words, the backscatter sensors do not require radio electronics circuitry (such as oscillators, mixers and amplifiers) and do not transmit or receive radio signals, therefore they can be made smaller and provide longer life than more complicated designs.

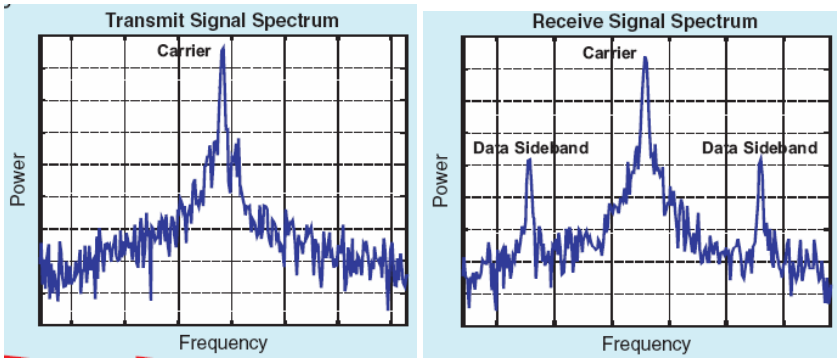


Figure 4.6: Spectrum of a Radio Backscattered system. The reader transmit the carrier, the RFID Tag sensor perturbs it adding the information

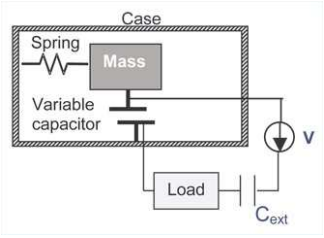
4.1.4 Exploiting Mechanical Vibrations

Mechanical vibrations are present in active machinery such as engine and turbines and building infrastructure (heating ducts, bridges, lamppost,...). The available output energy can vary dramatically with vibration amplitude or even at the vibration frequency, depending on the power electronics used to convert the harvester output voltage. Several recent papers describe the theoretical and measured output power from vibration-based energy scavenging devices. Vibrations with frequencies between 60 Hz and 1 kHz and amplitudes between 2.5 μm and 5 μm can produce power on the order of 100 $\mu W/cm^3$. There are three main techniques for scavenging energy through vibrations:

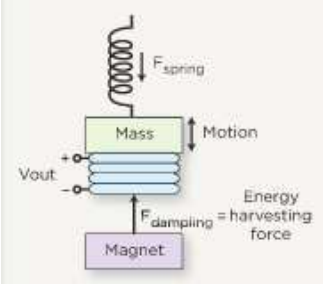
- Electromagnetic-energy harvesting uses a magnetic field for the convertion of mechanical energy to electrical. A coil attached to the oscillating mass crosses a magnetic field that is established by a stationary magnet. The coil travels through a varying amount of magnetic flux, inducing a voltage according to Faraday's law. The induced voltage is inherently small and must therefore be increased to viably source energy;

- Electrostatic converters place charge onto a variable capacitor and allow the vibrations to move the capacitor plates apart, reducing the capacitance. Capacitance and charge are related as $Q = CV$, so if Q is fixed and C decreases, V must increase. The most attractive feature of this method is its IC-compatible nature, since MEMS variable capacitors are fabricated with relatively mature silicon micromachining techniques. This scheme produces higher and more practical output-voltage levels than the electromagnetic method, with moderate power density;

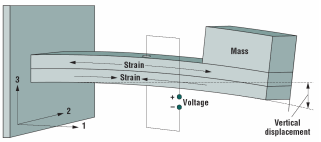
- Piezoelectric generators use a piezo beam mounted as a cantilever with a mass on the unsupported end. Electrodes are plated on surfaces of the beam and sometimes between layers of the piezo material. As the beam flexes, a potential difference is created between the electrodes. Strain, or deformation, in a piezoelectric material causes charge separation across the device, producing an electric field and, consequently, a voltage drop proportional to the stress applied. The oscillating system is typically a cantilever-beam structure with a mass at the unattached end of the lever, since it provides higher strain for a given input force. The voltage produced varies with time and strain, effectively producing an irregular ac signal. Piezoelectric energy conversion produces relatively higher voltage and power density levels than the electromagnetic system. Piezoelectric converters combine most of the advantages of both inductive and electrostatic generators. However, piezoelectric converters are difficult to implement on MEMS processes.



- More easily implemented in standard micromachining processes;
- Requires a separate voltage source (such as a battery) to begin the conversion cycle.



- Typically output AC voltages is below 1 volt in magnitude;
- Not easy to implement with MEMS technologies.



- The output voltage is irregular and depends on the constructions and thus an over voltage protection circuits may be required.

Several research groups have been worked on harvesting form vibration, in the following we will show only the most important and recent solutions. Roundy et al. [64] have analyzed common vibration sources that occur in large commercial buildings, cars, trains... These sources produce low level vibrations that can generate 300 $\mu W/cm^3$. Meninger et. al. [65] of MIT present an electrostatic generator that employs a variable micromachined capacitor. If the charge on the capacitor is maintained constant while the capacitance decreases (reducing the overlap area of the plates or increasing the distance between them), the voltage will increase. If the voltage on the capacitor is maintained constant while the capacitance decreases, the charge will decrease. They turned out that the mechanical energy converted into electrical energy is greater if the voltage across the capacitor is constrained than if the charge across the capacitor is constrained.

Electronic systems that harvest energy from vibration or movements require to be located close to the source. Human activity can be considered as a forge of kinetic and vibration energy. Walking is one of the usual human activities that have associated more energy [66]. Piezoelectric materials, electrostatic and electromagnetic transducers have been employed in order to harvest energy from human walking activity by the MIT Media Laboratory. In particular it has turned out the most important results employing piezoelectric materials [67]. The low-frequency piezoelectric shoe signals are converted into a continuous electrical energy source. Paradiso et al. [68] presented a piezoelectric pushbutton to wirelessly transmit a digital identification code using the mechanical energy given by pushing a button without the need of batteries.

Enocean [69], a spin-off of Siemens, has a commercial wireless remote control without batteries, PTM 100 and PTM 200. The force employed to push down the switch is used bending the cantilever piezoelectric ceramic at its extreme but without resonance. Another company that is looking into using piezoelectric sources to power networks of wireless sensors is MicroStrain Inc. [70]. This company setup an experiment where piezoelectric transducers were attached to the support beams in a structure. As the structure was constantly under strain, the voltage created by the piezoelectrics was stored up in a capacitor. Once the capacitor voltage reached a certain level, the power was than transferred to a transmitter which sent a wireless signal to some receiver. It was reported that the cycle time was about 20 to 80 seconds to store up a

charge of 9.5 V on the capacitor given the size of the piezoelectric was 17 cm2. Ferro Solutions makes a product called the Energy Harvester [71]. This little device about the size of two AA batteries contains an electromagnetic generator inside. There are two magnets and in between them is a coil of wire. When vibrations cause the coil of wire to move around in the magnetic field, current is generated in the wire. This small power source, about 1 to 10 mW, could be used in place of batteries or as a means to recharge batteries.

4.1.5 Air flow

Wind power has been used on a large scale as a power source for centuries, but it seems, to be less attractive for embedded system supply, probably for the size of mechanical components. The potential power from moving air is quite easily calculated as shown in the following equation.

$$P = \frac{1}{2}\rho A\nu^3 \tag{4.1}$$

where P is the power, ñ is the density of air, A is the cross sectional area, and v is the air velocity.

Zhand et al. [72] developed a piezoelectric windmill (Fig. 4.7) that provides electric power-generation from wind energy exploiting the piezoelectric effect. Piezo actuators are arranged along the circumference of the mill in the cantilever form. Using the camshaft gear mechanism an oscillating torque is generated through the flowing wind and applied on the actuators. The resulting power is around tens of mW.

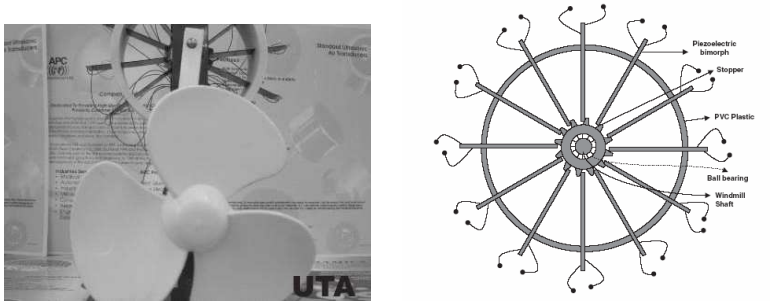


Figure 4.7: Prototype of the piezoelectric windmill and its schematic

4.1.6 Power Lines parasites

There are application scenarios in which embedded devices are very close to power lines but employing them is not allowed or it is just uncomfortable. The typical application is a wide monitoring system of a power grid. In this case

the most common solution to scavenge energy is the use of coils that transmit energy by the phenomenon of electromagnetic induction. Mark Major and his staff [90] at the Surveillance and Geolocation Department in SwRI's have developed a large sensor network that can widely monitor a power grid (Fig. 4.8). They supply the sensor nodes from power harvester coils used also for current measurements.

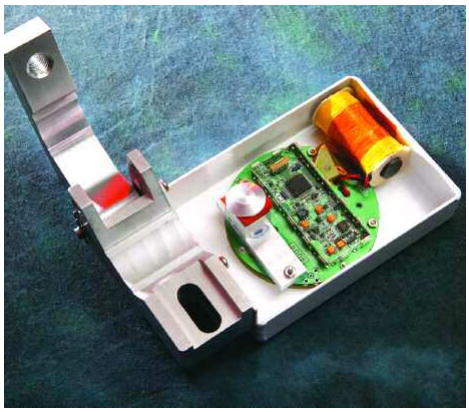


Figure 4.8: Power grid harvester, sensor node of SwRI's

Several types of non-contacting current sensors using a variety of magnetic materials are available, including current transformers, Hall sensors, Rogowski coils, magnetoresistive current sensors, and fluxgate sensors. Unfortunately not all these sensors can be exploited as power harvester. Hall effect sensors, for example, usually are not passive sensor and they are built with extra circuits for signal amplifying. Therefore, sensor device should operates passively, requiring no supply power in order to generate signal. Rogowsky coil (Fig. 4.9), instead, could be a suitable solution. It is light-weight and compact compared with most other devices, particularly where high currents are involved. This has obvious advantages with transport and ease of installation but there are other, less obvious advantages. Thin, flexible coils can be installed in uncomfortable places and are far less likely than any other transducers to need any modifications to be made to the plant.

In domestic power plant, Leland et al. [91] have realized a device that couple to magnetic fields surrounding wires carrying AC current, with view to developing both energy scavenging power sources for wireless sensor nodes and passive, proximity based current sensors (Fig. 4.10). Devices were constructed from cantilever mount piezoelectric material and permanent magnets. In energy scavenging operation, devices generated 208 μW.

Splashpower Ltd. [89] provides a product for wireless power transfer solutions to the mobile and portable electronics industry, enabling a wide range

Figure 4.9: Examples of Rogowski coils

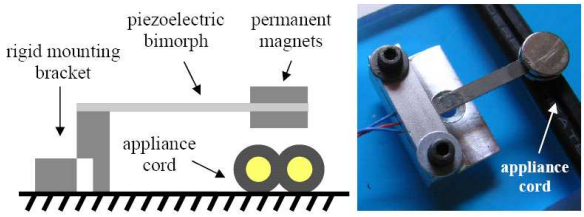


Figure 4.10: Power line scavenging using a piezo beam, schematic and photo

of portable electronic devices to be powered wirelessly from a single platform. It is proposed as universal wireless charging platform which delivers power to mobile devices. The SplashPad is a portable flat surface less than 6mm thick powered from any electric outlet. All mobile devices fitted with a suitable power module are ready to receive wireless power from the pad. The millimeter thin receiver module is customized to the shape, size and power requirements of the device and can be easily integrated into the host device or add-on accessories.

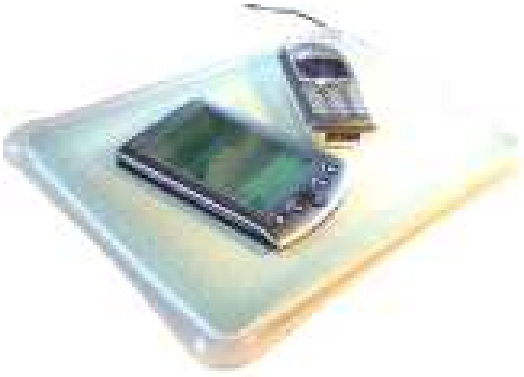


Figure 4.11: Splashpower, an electromagnetic pad that charges mobile devices

4.1.7 Biofuel Cells

Research on microbial fuel cells ('biofuel cells') has recently received increased attention as a means to produce 'green' electricity from natural substrates, such as carbohydrates In a microbial fuel cell, the bacteria do not directly transfer their produced electrons to their characteristic electron acceptor, but route them to an external path. Therefore the transport process is subsequently conducted over an anode, a resistance or power user, and a cathode.

There are different type of biofuel cell depending on the kind of bacteria (or enzyme) used, and the type of redox reaction performed. The enabling chemical components of the cell are the electrocatalyic "wired" enzyme films of its electrodes, comprising immobilized redox enzymes and redox hydrogels "wiring" Typically, bacterial energy is directly converted to electrical energy [73]. Research has obtained that a cell of 0.5mm^2 can operate continuously for one week at 37 $^\circ C$, providing a power of 1.9 μW at 0.52V.

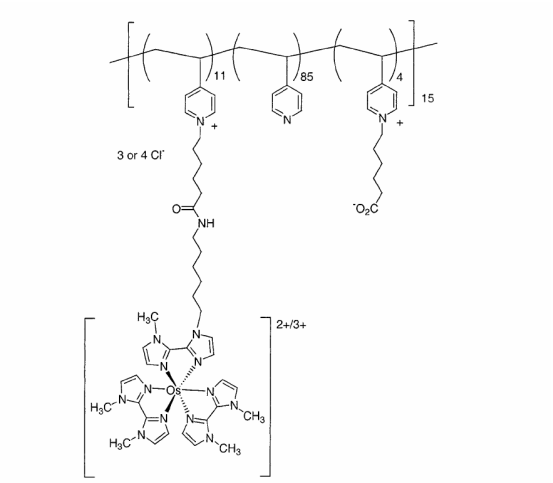


Figure 4.12: Structure of a polymer for a redox reaction.

4.2 Commercial products

There are several companies who are focusing on finding efficient energy scavengers for embedded devices. In the following, a little list of the most important ones will be described.

Enocean, GmbH

Enocean is a german company. The slogan for this company is "no batteries and no wires." They create products that use piezoelectric transducers to power RF transmitters. ECO 100, for example, is an energy har-

vester for linear motion. This device can be employed as a light switch that requires no wiring at all. When the light switch is flipped, this motion is used by the transducer to power the RF transmitter, which signals the receiver on the actual light to turn on.Enocean employs also photovoltaic technology to supply transmitters and transceivers, that are the most power consuming part of a sensor node. STM 100 is an extremely power saving RF transmitter module, that enables the realization of wireless and maintenance free sensors. Power supply is provided by a small solar cell. An integrated energy store allows unrestricted functionality for several days in total darkness.

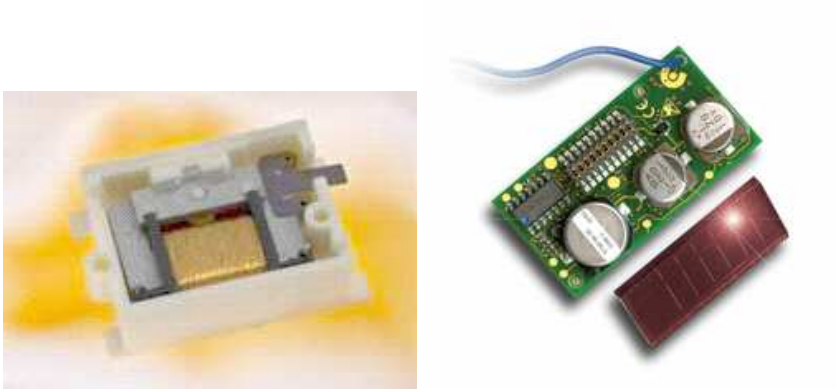


Figure 4.13: enOcean products

Ferro Solution, Inc.

Ferro Solutions has developed novel energy harvesting devices based on unique combinations of magnetic materials and patent-pending designs to Convert low-level vibrations into usable electricity. In many cases, energy harvesters can take the place of batteries and eliminate the need for costly maintenance to replace batteries. Our energy harvesters create usable electricity from low-level vibrations, power to run sensors, RF transmitters, micro-motors, and other electrical devices.

MicroStrain, Inc.

MicroStrain develops and produces innovative, smart, wireless, microminiature displacement, orientation, and force sensors. Although not directly involved in energy scavenger development, this company setup an experiment where piezoelectric transducers were attached to the support beams in a structure. As the structure was constantly under strain, the voltage created by the piezoelectric materials was stored up in a capacitor.

Thermo Life, Inc.

Thermo Life® Energy Corp. introduced its 3-volt thin film thermoelectric device in 2001. It is a small and compact Low Power Thermoelectric Generator (LPTG) and viable energy source for low-powered devices such as micro sensor systems, ZigBee chipsets, wearable electronics, implantable medical devices, active RFID tags and numerous other applications. Thermo Life® converts heat energy to electrical energy through its thermopile couples using the thermopile principle (Seebeck effect). When both heat couple plates are thermally connected with a heat source and a heat sink, heat flows through thermopiles and is converted directly into electrical energy. Even small temperature differences of less than 5 Kelvin can provide a source of thermal energy.

Splashpower, Ltd.

The Splashpower solution is based around technology that uses inductive coupling as a means of power transfer. It consists in has two parts: the SplashModule^{TM}, a sub-millimeter thin receiver module that can be customized to just about any size, shape or curve of a device and makes no visible impact to product appearance; and the SplashPad^{TM} that is a thin (less than 6MM) universal wireless charging platform that plugs into any electric outlet and can be built into any surface with no possibility of shock or discharge.

Perpetuum, Ltd.

Perpetuum is a vibration-energy harvesting microgenerator based on a magnet and coil arrangement that successfully transforms the kinetic energy of vibration into a low power electrical signal. It is designed to resonate at mains frequency (50 or 60Hz) with a bandwidth of 0.2Hz giving excellent performance on any AC synchronous motor powered equipment. Output is from 0.1mW to several mW power depending on the level of vibration (eg up to 5mW at 100mg or 0.4mW at 25mg).

Figure 4.14: Perpetuum microgenerator

Powercast, Llc.

Powercast uses the energy from a transmitted RF signal to power small embedded devices or wireless sensor node. The transmitter can be placed everywhere, for example, in anything that plugs into the wall (lamps, alarm clocks, and so on) and can send a low, continuous signal to small gadgets that contain an embedded receiver. Anything within its range and equipped with a Powercast receiver, if provided with either batteries or supercapacitor, will be continuously replenished. It isn't just a replacement for a universal charger, but it's developed to either continuously charge a battery or replace the need for them altogether.

Greene et al [98], have successfully tested and deployed radio frequency (RF) power harvesting in a real environment monitoring situation, using the Pittsburgh Zoo & PPG Aquarium as scene.

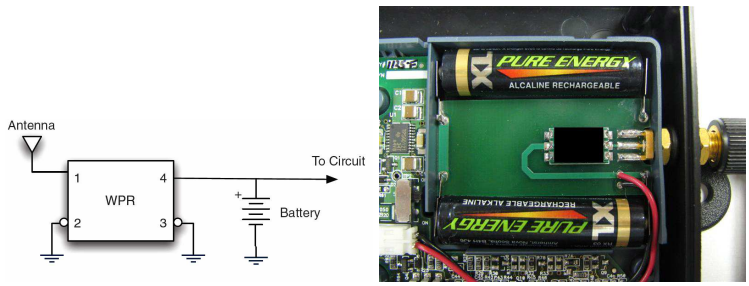


Figure 4.15: Powercast RF scavenger

Chapter 5

Efficient solar energy scavenger for wireless sensor nodes

Solar harvesting circuits have been recently proposed to increase the autonomy of embedded systems. One key design challenge is how to optimize the efficiency of solar energy collection under non stationary light conditions. This chapter proposes a scavenger that exploits miniaturized photovoltaic modules to perform automatic maximum power point tracking at a minimum energy cost. The system adjusts dynamically to the light intensity variations and its measured power consumption is less than 1mW. Experimental results show increments of global efficiency up to 80%, diverging from ideal situation by less than 10%, and demonstrate the flexibility and the robustness of our approach.

5.1 Introduction

Although research continues to develop higher energy-density batteries and very low power embedded platforms have recently entered the marketplace, the amount of available energy on board still severely limits the lifespan of distributed battery operated embedded systems. The ultimate goal is to achieve a perpetually powered system without a necessary periodical maintenance for battery replacement or recharging. Wireless sensor networks for outdoor environmental monitoring are a class of systems where exploiting alternative power sources could increase the autonomy of the nodes considerably. Energy harvesting techniques can solve the problem by supplying and converting energy from the surrounding environment and refilling an energy buffer formed

by a battery stack or by supercapacitors.

Energy scavengers using small photovoltaic *PV* modules have been recently proposed to enable perpetual operation of WSNs. Unfortunately the low energy budget available does not help to perform an efficient replenishment of the storage devices. Under varying temperature or irradiance condition the output characteristics of a *PV* module changes non-linearly. Therefore the problem is to automatically find the voltage (and the current) at which it should operate to obtain the maximum power output. An efficient photovoltaic energy harvesting system should track this particular operating point called *Maximum Power Point* (*MPP*). To this purpose many designers have actively investigated techniques for *MPP* tracking (*MPPT*) [74, 75]. So far *MPPT* methods have been roughly classified into two groups: large-scale *PV* power systems, generally based on digital signal processor (DSP) or microcontrollers [76], and small-scale *PV* power systems usually without DSP or any other digital controller. The distinction is important because using DSPs may reduce cost-efficiency in *PV* applications below 50W [77]. However, with the increased interest in harvesting technology for wireless sensor networks a new class of *MPPT* methods, focused on micro-scale *PV* power systems, has recently emerged. This approaches tackle the development of *MPPT* methods with power consumption of a few mW. In fact, *MPPT* methods using small-size *PV* modules are only practicable if the power consumed by the tracker is substantially lower than the amount of output power that it gains. The harvester circuit proposed in this thesis consumes less that 1mW and approaches the ideal situation beyond 90%. Since it does not use any microcontroller or DSP for *MPPT*, the embedded system can be shut down when unused to save energy. Moreover it does not require a pre-charged storage device (such as rechargeable batteries), but it simply starts to work even if the energy buffers are empty. Because of all these features, the proposed solution is suitable for low-power systems and in particular for wireless sensor networks.

The remainder of the chapter is organized as follows. Related works are reviewed in the next section followed by the list of the contributions and the innovations that our work proposes. A background on problems and characteristics of *MPPT* systems is discussed in section 5.4. Section 5.5 describes the actual implementation of our *PV* harvester. Experimental results and the performance achieved are the focus of section 5.6, finally section 5.7 concludes the chapter.

5.2 Related Work

Although the experience in exploiting *PV* modules is consolidated, research on solar power scavenging for sensor network is quite recent. Several solutions have been presented in the last years: *Prometheus* [59] and *Heliomote* [78] were probably the first proposals to supply a sensor node with the help of a small *PV* module. Both systems do not perform any *MPPT* and the replenishment of energy buffers (composed by supercapacitors, rechargeable batteries or a combination of both) is performed by a simple link between the *PV* panel and the storage device. A direct connection between the *PV* module and the buffer device forces the operating point to the capacitor voltage V_{CAP}. Since it is far from optimal, it drastically lowers the output power of the *PV* module. The adoption of a diode to protect the solar module does not help because the *PV* cell only works when the buffer voltage is lower than the *PV* panel voltage, and the diode is forward biased. As a consequence, the amount of power drawn from the *PV* module also depends on the energy buffer level (V_{BAT} or V_{CAP}). This also causes problems in developing power management software for sensor networks. Adding *Energy Harvesting aware features* might require the knowledge of the current available energy and the estimation of the future harvested power [79, 80] to tune the behavior of the system. Clearly, if there is a relation between conversion efficiency and energy buffer level, the energy prediction will require more computing effort and may lose in accuracy.

Chou et al. proposed *Everlast* [60], a *MPPT* system that uses only a supercapacitor as energy reservoir. The microcontroller on the node runs a tracking algorithm and drives a pulse width modulator (*PWM*) circuit to control the power converter. The microcontroller is essential for the *MPPT* system which leads to the main drawback of this implementation, that is the harvesting circuit is too deeply integrated to the rest of the sensor node. The *MPPT* circuit cannot work without microcontroller and exploiting the scavenger on another sensor node requires a revision of the firmware. *Ambimax* [61] is another solution from the same authors of [60]. It is a more refined version and tries to eliminate the overhead caused by an always-running algorithm on the microcontroller. The control unit is independent from the target embedded system and does not require any programmable device. Although it uses only low power logic components, the *MPPT* system requires the presence of a rechargeable battery as secondary buffer to work when the primary buffer is empty. The sensor used to estimate the position of the peak power point is a photodiode. Unfortunately this component has a not negligible power consumption and it requires also a conditioning circuit for the output signal. However the innovative idea to exploit a low power comparator to generate a *PWM* signal has

been adopted also in our work as starting point. We remove some inefficiency and add new features as described in the next section.

5.3 Contributions

This chapter points to the following new results:

- We present a low-energy photovoltaic harvesting circuit for wireless sensor nodes, that is able to track variations of the *MPP* during the day. Differently from [61], our method uses a very small *PV* module as pilot cell. To the best of our knowledge nobody has experimented and implemented this solution in a *MPPT* system, especially in a small-size energy scavenger for sensor networks. The advantage is that the pilot cell does not require any additional energy, whereas exploiting a photodiode consumes power;

- Differently from Chou et al. [61], who use a battery as secondary buffer, in our work *MPPT* does not require batteries. Our supply unit chooses automatically the source to power the *MPPT* system between the *PV* module or the DC/DC converter used by the sensor node. As a result, the *MPP* can be tracked immediately even with empty energy buffer. This behaviour is desirable since (a) it matches common models used in power management software and (b) it guarantees faster recovery;

- The control circuit of the proposed scavenger is completely independent from the sensor node and its logic. Therefore it saves more power because it is possible to shut down not only the microcontroller, sensors and other devices on-board, but also the step-up converter that is no more required for the scavenger;

- Even if the average power consumed by the sensor node is higher than the power drawn from the environment leading to the emptying of the energy reservoir (e.g. it happens in cloudy days, or in the sunset), the scavenger detaches the sensor node increasing the scavenging efficiency, and collects an amount of energy sufficient to perform some basic recovering operations before supplying the node again.

5.4 Background and Problem Statements

5.4.1 *MPPT* techniques

The *I-V* characteristic of a *PV* module, when neglecting the internal shunt resistance, is given by the following equation:

$$I_o = I_g - I_{sat}\left\{e^{\frac{q}{AKT}(V_o + IoR_s)} - 1\right\} \tag{5.1}$$

where I_g is the generated current, I_{sat} is the reverse saturation current, q is the electronic charge, A is a dimensional factor, K is the Boltzmann constant, T the temperature in degree Kelvin, R_s the series resistance of the cell. The plot of the *PV* module adopted in our solar harvester is shown in Fig. 5.1. It shows the behavior under two different light conditions.

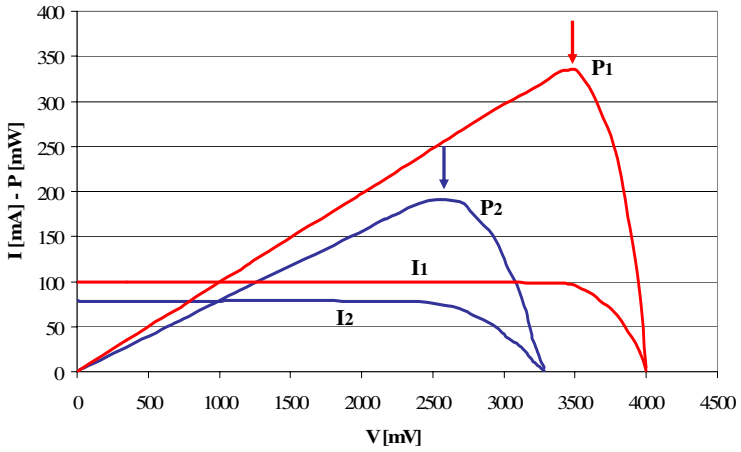


Figure 5.1: Characteristic I-V and P-V plots of the used photovoltaic module

The problem considered by *MPPT* is to automatically find the operating point (V_{PV}, I_{PV}) at which a *PV* module should operate to obtain the maximum power output under a given temperature and irradiance, and to follow it if light intensity changes (indicated with arrows in Fig. 5.1).

Fig. 5.2 shows the *I-V* characteristics of the solar array and the load, together with constant power curves ($P = VI = const$). It is evident that the delivered output power, which is represented by the operating point 1, is significantly smaller than the maximum output power, which is represented by point 2.

There are several methods and algorithms to analyze and find the *MPP*, certainly the most used are *Hill-Climbing/Perturb and Observe (P&O)* [81, 82, 60] and *Fractional Open-Circuit Voltage (FOC)* [77, 75]. *Hill-Climbing and P&O* methods are widely used because of their simplicity and the small number of parameters which have to be measured. The tracker operates by periodically per-

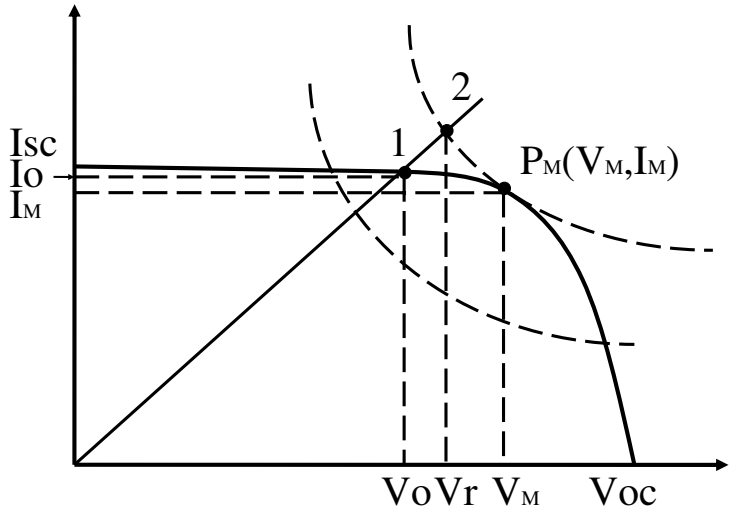


Figure 5.2: The operation of the *MPPT*

turbing the system (changing the duty ratio of the converter in Hill-Climbing or incrementing or decrementing the solar array voltage in the case of P&O). If a given perturbation leads to an increase (decrease) in output power, the subsequent perturbation is made in the same (opposite) direction. In this manner, the tracker continuously looks for the peak power point. Implementations of these methods are usually very accurate and exploit DSPs or microcontrollers, that consume non-negligible power. However, the method cannot readily track immediate and rapid changes in environmental conditions. On the other hand, among the simplest methods for *MPP* tracking, Fractional Open-Circuit Voltage is definitely the most used and cost-effective in small-scale *PV* systems. This method exploits the nearly linear proportional relationship between the operating voltage at *MPP* (V_{MPP}) of a *PV* module and the open circuit voltage (V_{OC}).

$$V_{MPP} \approx K_{FOC} \cdot V_{OC} \tag{5.2}$$

where the proportional constant K_{FOC} belongs to the interval $(0.71, 0.78)$, with a slow increase when the light intensity fades. Table. 5.1 shows the behavior of the constant (K_{FOC}) of the solar cell used, during different irradiance conditions.

The *MPP* can be approximated measuring periodically V_{OC} by a temporary disconnection of the *PV* module from the circuit. Of course it is a disadvantage because of the temporarily drop of power. To overcome this problem, we exploit an additional small *PV* module acting as pilot cell, carefully chosen to closely represent the characteristics of the principal *PV* array. We adopt the

$V_{OC}\,[V]$	$V_{MPP}\,[V]$	K_{FOC}
3,24	2,50	0,772
3,40	2,52	0,741
3,40	2,44	0,717
3,60	2,80	0,778
3,60	2,60	0,722
3,96	2,92	0,737
4,50	3,50	0,778
4,50	3,30	0,733

Table 5.1: Fractional Open-Circuit Voltage, Relation between V_{OC} and V_{MPP}

CPC1824 from Clare, Inc. [83]. It is a monolithic photovoltaic string of solar cells of only $0.1cm^2$, and it is used as light irradiance sensor to provide feedback information for the tracker. The current value of the pilot cell follows almost linearly the behavior of the main *PV* module during light variations. Fig. 5.3 compares the behavior of the two cells under the same solar intensity variation. The plot displays the operating voltage of the pilot cell ($V_{pilotcell}$) over the V_{OC} of the big *PV* module. For clarity we plot also the same V_{OC}, and the V_{MPP} to verify the affinity of the variations. As shown the behavior is near linear and it is possible using the signal from CPC1824 as a reference to track the position of the *MPP*. As a consequence, we can obtain a nearly linear relation between V_{MPP} and the reference from the pilot cell $V_{pilotcell}$(5.3).

$$V_{MPP} \approx K_{FOC} \cdot V_{OC} \approx K_{FOC} \cdot (K_{pilot} \cdot V_{pilotcell}) \tag{5.3}$$

Moreover, using a pilot cell has several additional advantages: 1) it is no more necessary to provide a power supply for an irradiance sensor in the tracking unit; 2) the circuit is small and does not require complex components such as other photo-detectors.

Another point to notice is that the linear relationship is only an approximation, and the *PV* cell technically never operates at the exact *MPP*. Even if this method is not a completely accurate *MPPT* technique, it does not require a DSP or microcontroller and it is very cheap and easy to implement.

5.4.2 Power converter

The power converter used in *MPPT* techniques is usually chosen between a buck, a boost or a buck/boost configuration. We opted for a buck power converter (Fig. 5.4a) since the solar cell's nominal voltage is higher than the values reachable by the supercapacitor. The operation of the buck converter is pretty simple, with an inductor and two switches (in our case a MOSFET transistor and a diode) that control the inductor. It alternates between energy stor-

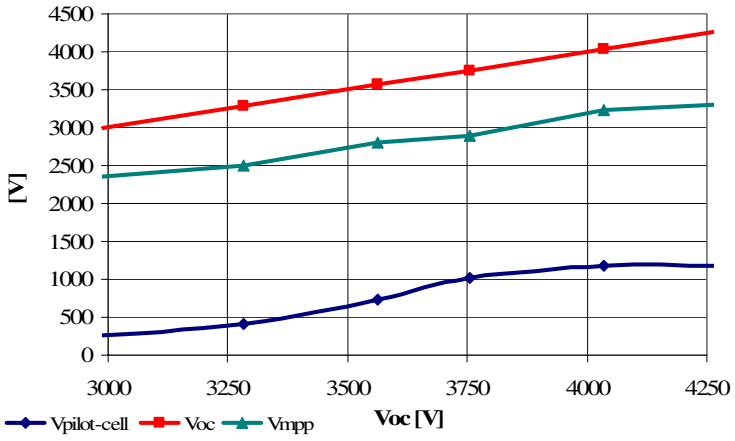


Figure 5.3: Near linear relation between V_{MPP} and the pilot cell

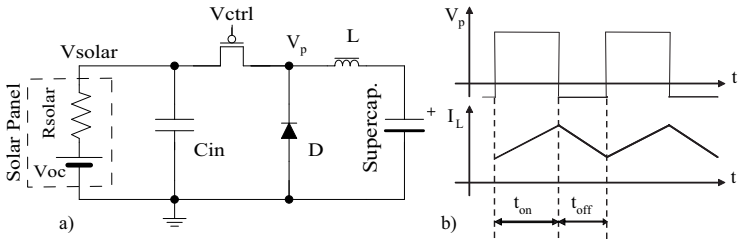


Figure 5.4: MPP regulator and waveforms

ing, connecting the inductor to voltage source, and discharging the inductor to replenish the load. Depending on the load and the circuit parameters, the operating modes can be either continuous or discontinuous. It operates in continuous mode if the current through the inductor (I_L) never falls to zero during the commutation cycle before the end of the switching period. We adopt this behavior and the associated waveforms are shown in Fig. 5.4b.

The relationship between input and output voltage can be described by the following equation:

$$\frac{V_{OUT}}{V_{IN}} = \frac{t_{on}}{T} = D \tag{5.4}$$

where D is the duty ratio of the switching cycle. When the buck converter is used in *PV* applications, the input power (V_{PV}, I_{PV}) varies continuously with the atmospheric conditions, thus the duty cycle (D) changes in order to track the *MPP*. The switching frequency, and the inductor value is a trade-off between converter efficiency, cost and power consumption. For example, the higher the switching frequency, the lower the inductor size, but also tracker power consumption and losses are higher. In our case the buck converter acts

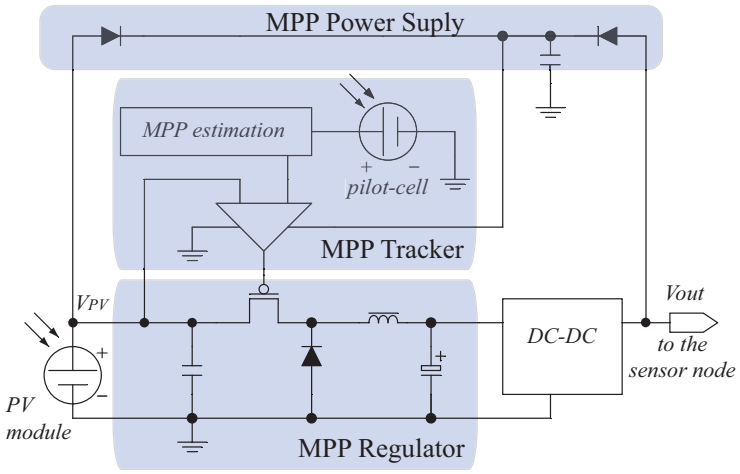


Figure 5.5: Schematic diagram of the harvester platform

as *MPP* regulator and is driven by narrow pulses with a duty-ratio D generated by the tracking unit.

5.5 System Design

The hardware architecture of the solar scavenger is displayed in Fig. 5.5. It consists of three units: the *MPP Regulator*, the *MPP Tracker* and the *MPP Power supply*. The *MPP Regulator* unit is formed by a buck power converter depicted in section 5.4.2.

The control signal V_{ctrl} is generated by the *MPP Tracker* that compares the V_{REF} and the current operating voltage V_{PV} of the *PV* module. As pointed out in sec. 5.4.1, the *Fractional Open-circuit* method is exploited by matching V_{REF} to the estimated V_{MPP} using approximation (5.3) and the pilot cell as input signal. The ultra-low power comparator adopted consumes less than $100\mu W$, and it is also possible to adjust the hysteresis window (see Fig. 5.6a) around V_{REF}, tuning the size and the position. In this way the actual operating point is not a fixed value, but oscillates around the *MPP*. Narrowing the windows around estimated *MPP* means to operate at higher switching frequencies, with higher conversion efficiency, on the other hand it requires a faster comparator, which is generally more power consuming. Moreover, tightening the hysteresis band makes the tracking operation critical and requires more accuracy.

The importance of a tracking method in this implementation is clearly explained in Fig. 5.6b. It displays the variation of the *P-V* characteristic of a photovoltaic module under three different irradiance intensities, keeping the window position fixed. A system equipped with a *MPP* regulator without a tracking unit can obtain the maximum efficiency only if the *MPP* is situated

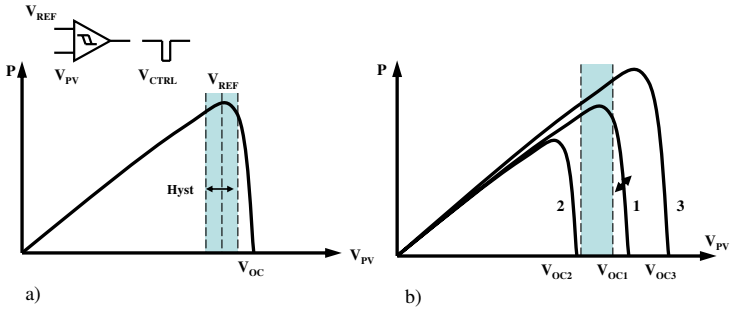


Figure 5.6: Window around *MPP* and tracking significance

inside the band (1). A higher light intensity shifts the curve to the right (3) and causes the *MPP* regulator to operate at a non-optimal power point. The worst and fatal case (2) occurs when the light fades. No point of the curve crosses the control window, therefore no PWM signal V_{ctrl} is generated to the power converter, causing a real disconnection between *PV* source and energy reservoir. (The performance in this case is obviously worse than not using *MPPT* at all and connecting the *PV* cell directly to a storage device).

The *MPP power supply* unit powers the MPP block even if the DC-to-DC does not work because the voltage of the supercapacitor is not enough to start it up or because it is shut down with the whole sensor node. The *MPP power supply* unit switches dynamically between the DC/DC output and *PV* module choosing the higher source. This permits to have the highest control signal V_{ctrl} to switch-off the MOS transistor, suppressing any drain current that could decrease the performance of the power converter. The presence of a couple of fast diodes allows supplying any *MPP* blocks with the highest available power source.

5.6 Experimental Results

We investigate the performance of the proposed energy scavenger by measuring the power consumed by the *MPPT* system, measuring the charging efficiency of the energy buffer and estimating the efficiency during the tracking of the peak power point. Finally we test the sustainability of the scavenger powering a real wireless sensor node. All the measurements have been performed using a *PV* module of $112cm^2$ in size and a nominal output power of $400mW$.

5.6.1 Power consumption

When the switch in the *MPP* regulator is open, the overall circuit consumes less than half a mW. ($P_{switch-off} \approx 300\mu W$). Peaks of about $1mW$ ($P_{switch-on}$)

have been measured when the comparator switches the MOS transistor on. The average power can be easily computed using the following equation.

$$P = D \cdot P_{switch-off} + (1 - D) \cdot P_{switch-on} \tag{5.5}$$

Considering that the duty cycle of the control signal V_{ctrl} is typically higher than 80%, the average power consumed by the whole circuit is below $1mW$.

5.6.2 Scavenging efficiency

Conversion efficiency is usually defined as follow:

$$\eta = \frac{P_{transferred}}{P_{MPP}} \tag{5.6}$$

$$P_{MPP} = V_{MPP} \cdot I_{MPP} \tag{5.7}$$

$$P_{transferred} = \frac{1}{2}\frac{C}{T}\left[V^2(t) - V^2(t-T)\right] \tag{5.8}$$

Where P_{MPP} is the power at the *MPP* and $P_{transferred}$ is the average power transferred to the energy buffer. When supercapacitors are used, $P_{transferred}$ is usually computed as the value necessary to increase the energy level from $E\left(t_1 - T\right)$ to $E\left(t_1\right)$ during a given time interval T . In our tests, η has been evaluated also considering the DC-to-DC converter which affects the result with its own intrinsic losses. Fig. 5.7 shows how efficient the proposed method is in replenishing the supercapacitor over elapsed time. The continuous curve represents the linear charging behavior in case of a direct connection between the *PV* panel and the storage device. The dashed curve shows the ideal trend: the supercapacitor is constantly refilled with the maximum available power P_{MPP}. The charging behavior using the proposed energy scavenger is plotted as the curve with dots. As shown it approaches closely the ideal curve, with a maximum error less than 10%. The curve with triangles has been obtained excluding from the scavenging platform the *MPP power supply* unit and using the DC/DC as only source. The discontinuous shape is due to the fact that for low voltages the *MPP* unit does not work properly, since the step-up converter is still switched-off.

Fig. 5.8 shows the same situation, but plots the efficiency, defined as in (5.6), over the voltage level of the supercapacitor. Looking at the curve with triangles, it is even more evident that only when the energy level of the buffer is high enough to turn the DC/DC on, the scavenger can work properly and increase the efficiency.

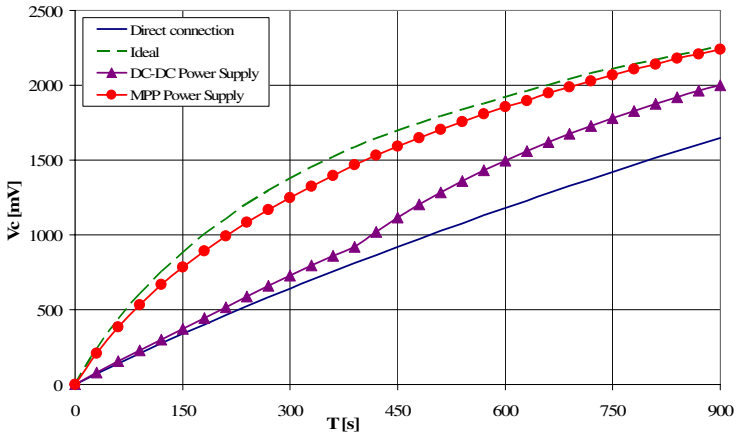


Figure 5.7: Comparison of different charging curves

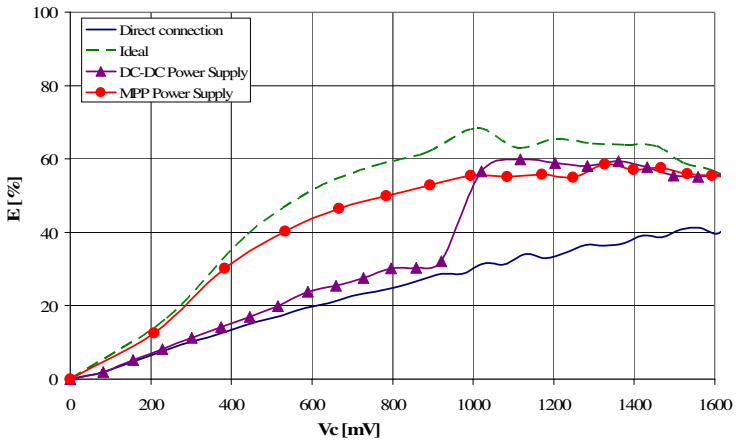


Figure 5.8: Efficiency of the power conversion

5.6.3 Tracking efficiency

To evaluate the performance of the peak power tracker, we tested how the scavenger efficiency varies if the *MPP Tracker* unit is switched off. Fig. 5.9 shows the efficiency defined in sec. 5.6.2 using a irradiance different from the previous plots. Also here the lowest continuous curve depicts the direct connection between *PV* module and supercapacitor, while the dashed curve is the ideal case using a constant P_{MPP} as source. The curve with dots is obtained with the *MPPT* proposed, and can achieve levels up to 80%. If *MPP Tracker* unit is excluded the efficiency curve varies widely with the irradiance condition, ranging from the dashed curve to below the continuous one, representing the direct connection. The figure displays a case in which η decreases to around 50% (curve with squares).

In Fig. 5.10, we also analyze the operating point V_{PV} with and without the

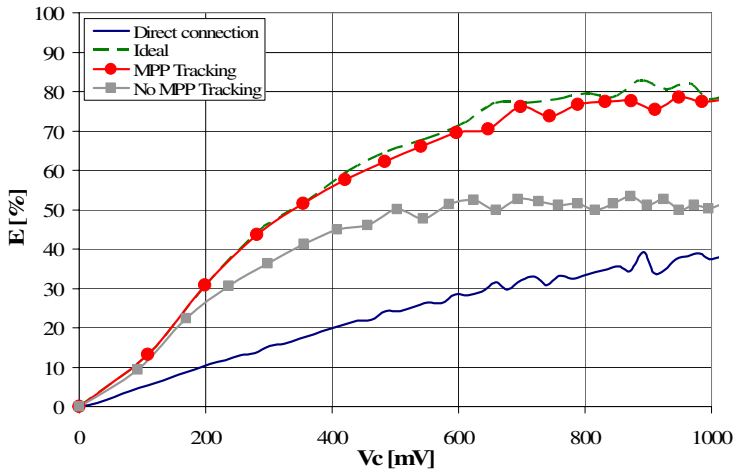


Figure 5.9: Efficiency of the tracking system

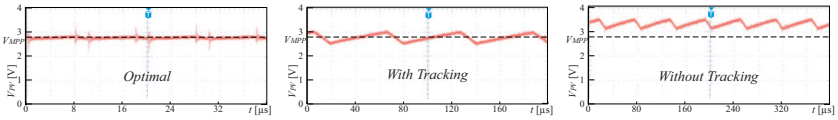


Figure 5.10: Comparison of the operating point with and without tracking circuit

tracker over the elapsed time. The ideal situation, depicted in the picture on the left, is obtained when the *MPP Regulator* is driven by a signal generator tuned at 100kHz and duty-ratio $D = 78\%$. D is computed in order to obtain an optimal behaviour, in fact the operating point continuously matches V_{MPP} at the given irradiance. Under the same condition, the proposed scavenger tightens around V_{MPP} (as shown in the middle picture), because it automatically generates a control signal with $D = 80\%$. The size of the window is due to a lower switching frequency (16KHz). Finally, the operating point of the circuit without the tracking system is plotted in the right picture. Since the duty cycle of the generated signal is very different from the ideal one (in this case $D = 91\%$), the *PV* module will never operate at *MPP*.

5.6.4 Powering a sensor node

To verify the sustainability of our solar energy harvester, we power a *Tmote sky* sensor node [7] with a simple example application. The application does not perform any duty ratio activity, neither any low power techniques. It just sequentially turns on the on-board LEDs, and transmits a counter value using the radio.

The average power consumed by the application is $90mW$. Using a fading light intensity for which the incoming average power is not enough to sustain

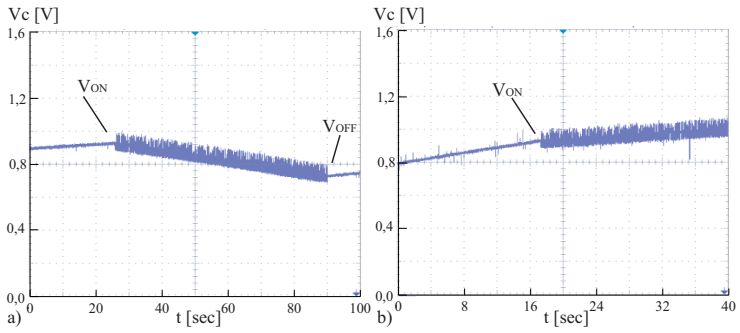


Figure 5.11: Sustainability of the scavenger under different light conditions

the sensor node, the behavior of the node is shown in Fig. 5.11a. The plot displays the voltage level of the supercapacitor used as energy reservoir, when the trace becomes larger the sensor node is active. It is easy to verify that the energy, required by the step-up converter to boot, is higher than the energy left when the DC-to-DC turns off again. Adopting a $50F$ supercapacitor, this amount of energy is computed as $\Delta E = \frac{1}{2}C\left[V_{ON}^2 - V_{OFF}^2\right]$ and it is around $6,93J$ allowing the sensor node to operate continuously for about 64 sec, before shutting down again.

Of course when the incident light increases, the power generated by the *PV* module becomes enough for continuous sensor node operations and for energy buffer replenishment (see Fig. 5.11b)

5.7 Conclusion

A highly efficient solar energy harvester for wireless sensor nodes and environmental embedded systems has been proposed. The adoption of a *MPPT* circuit leads to several benefits such as: the possibility to shrink the size of *PV* modules, to reduce the capacity of the energy reservoir, or to allow higher power consumption of a sensor node. The presented circuit performs a high-efficiency conversion through an ultra-low power *MPPT* technique that requires less than 1mW. The estimation of the peak power point is done automatically, using a small *PV* pilot cell as reference, whereby sensing operation does not require additional power. The scavenger can be used with any kind of embedded sensor node, because it is completely independent of the node operation. Experimental results have shown that the global efficiency diverges from the ideal situation by less than 10%. Finally, a case study on a real sensor node demonstrates the complete sustainability of the system with solar regenerative energy.

Chapter 6

Tasks scheduling on sensor nodes

Energy harvesting has recently emerged as a feasible option to increase the operating time of sensor networks. If each node of the network, however, is powered by a fluctuating energy source, common power management solutions have to be reconceived. This holds in particular if real-time responsiveness of a given application has to be guaranteed. Task scheduling at the single nodes should account for the properties of the energy source, capacity of the energy storage as well as deadlines of the single tasks. We show that conventional scheduling algorithms (like e.g. EDF) are not suitable for this scenario. Based on this motivation, we have constructed optimal scheduling algorithms that jointly handle constraints from both energy and time domain. Further we present an admittance test that decides for arbitrary task sets, whether they can be scheduled without deadline violations. To this end, we introduce the concept of energy variability characterization curves (EVCC) which nicely captures the dynamics of various energy sources. Simulation results show that our algorithms allow significant reductions of the battery size compared to Earliest Deadline First scheduling.

6.1 Introduction

Wireless sensor networks – consisting of numerous tiny sensors that are unobtrusively embedded in their environment – have been the subject of intensive research. As for many other battery-operated embedded systems, a sensor's operating time is a crucial design parameter. As electronic systems continue to shrink, however, less energy is storable on-board. Research continues to develop higher energy-density batteries and supercapacitors, but the amount of

energy available still severely limits the system's lifespan. As a result, size as well as weight of most existing sensor nodes are largely dominated by their batteries.

On the other hand, one of the main advantages of wireless sensor networks is their independence of pre-established infrastructure. That is, in most common scenarios, recharging or replacing nodes' batteries is not practical due to (a) inaccessibility and/or (b) sheer number of the sensor nodes. In order for sensor networks to become a ubiquitous part of our environment, alternative power sources should be employed. Therefore, environmental energy harvesting is deemed a promising approach: If nodes are equipped with energy transducers like e.g. solar cells, the generated energy may increase the autonomy of the nodes significantly.

In [84], several technologies have been discussed how, e.g., solar, thermal, kinetic or vibrational energy may be extracted from a node's physical environment. Moreover, several prototypes (e.g. [85, 59]) have been presented which demonstrate both feasibility and usefulness of sensors nodes which are powered by solar or vibrational energy.

From a networking perspective, classical protocols cannot harness the full potential provided by the harvesting technology. Here, several publications exist which make routing or clustering decisions within the network *harvesting aware* [86, 87]. Based on the knowledge of the currently generated power at the single nodes, the network lifetime can be optimized by shifting the communication and computation load. In the example of a solar powered network, nodes which are directly exposed to sunlight have to disburden nodes who are receiving less radiation due to shadowing effects.

In contrast, we focus on the *temporal* variations of the energy source experienced by a single node instead of *spatial* variations between several nodes. The obtained results can e.g. be applied to networks, whose nodes are independently from each other transmitting data to a base station. But even if we are dealing with a multi-hop scenario, a single sensor node may need to perform its activities in a timely manner using a limited and uncertain energy source. For example, a node may need to communicate with others by means of a energy-saving wireless protocol, e.g. by switching on transmitters only at well-defined time instances. In addition, there are application scenarios for which it is indispensable to fulfill *real-time* requirements like it is the case in, e.g., fire or intruder detection systems. In general, one can classify real-time application scenarios for wireless sensor networks into safety critical systems, smart spaces as well as entertainment [88]. For all these scenarios, our research reveals fundamental problems and tradeoffs when real-time behaviour has to be guaranteed although a sensor's driving energy source is highly unstable.

The following example shows why greedy scheduling algorithms (like Earliest Deadline First EDF) are not suitable in the context of regenerative energy.

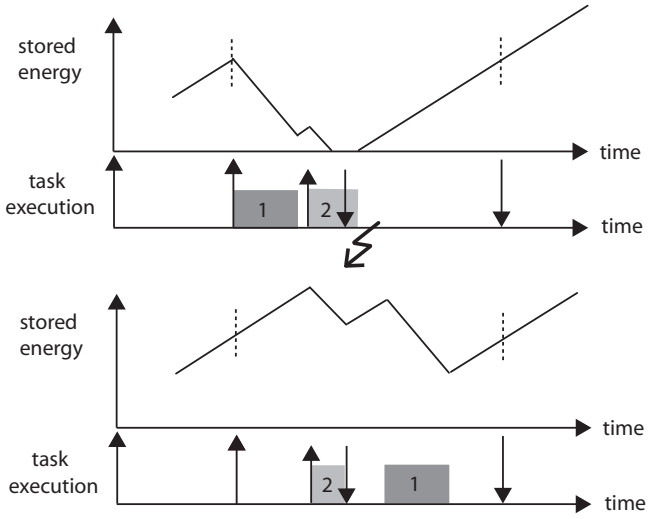


Figure 6.1: Greedy vs. lazy scheduling

Let us consider a node with an energy harvesting unit that replenishes a battery. For the sake of simplicity, assume that the harvesting unit provides a constant power output. Now, this node has to perform an arriving task ”1” that has to be finished until a certain deadline. Meanwhile, a second task ”2” arrives that has to respect a deadline which is earlier than the one of task ”1”. In Figure 6.1, the arrival times and deadlines of both tasks are indicated by up and down arrows respectively. As depicted in the top diagrams, a greedy scheduling strategy violates the deadline of task ”2” since it dispenses overhasty the stored energy by driving task ”1”. When the energy is required to execute the second task, the battery level is not sufficient to meet the deadline. In this example, however, a scheduling strategy that hesitates to spend energy on task ”1” meets both deadlines. The bottom plots illustrate how a *Lazy Scheduling Algorithm* described in this chapter outperforms a naive, greedy approach like EDF in this situation. Lazy scheduling algorithms can be categorized as non-work conserving scheduling disciplines. Unlike greedy algorithms, a lazy scheduler may be idle although waiting tasks are ready to be processed.

The research presented in this thesis is directed towards sensor nodes. But in general, our results apply for all kind of energy harvesting systems which must schedule processes under deadline constraints. For these systems, new scheduling disciplines must be tailored to the energy-driven nature of the problem. This insight originates from the fact, that energy – contrary to the computation resource ”time” – is storable. As a consequence, every time we withdraw energy from the battery to execute a task, we change the state of our scheduling

system. That is, after having scheduled a first task the next task will encounter a lower energy level in the system which in turn will affect its own execution. This is not the case in conventional real-time scheduling where time just elapses either used or unused.

The rest of the chapter is organized as follows: In the next section, we highlight the contributions of our work. Subsequently, Section 6.3 gives definitions that are essential for the understanding of the chapter. In Section 6.4, we present *Lazy Scheduling Algorithms* for optimal online scheduling and proof their optimality. Admittance tests for arbitrary task sets are the topic of Section 6.5. Simulation results are presented in Section 6.6 and Section 6.7 deals with practical issues. At the end, Section 6.8 summarizes related work and Section 6.9 concludes the chapter.

6.2 Contributions

The following chapter builds on [89] and [90], where Lazy Scheduling Algorithms have been presented for the first time. We combine both view points, extend the theoretical results by a formal comparison of the schedulability regions of EDF and LSA, include corresponding simulation results as well as a discussion of implementation aspects. Thereby, we outline how our algorithms could be implemented on real sensor nodes which illuminates the relevance of the proposed theory. The contributions described are as follows:

- We present an energy-driven scheduling scenario for a system whose energy storage is recharged by an environmental source.
- For this scenario, we state and prove optimal online algorithms that dynamically assign power to arriving tasks. These algorithms are energy-clairvoyant, i.e., scheduling decisions are driven by the knowledge of the future incoming energy.
- We present an admittance test that decides, whether a set of tasks can be scheduled with the energy produced by the harvesting unit, taking into account both energy and time constraints. For this purpose, we introduce the concept of energy variability characterization curves (EVCC). In addition, a formal comparison to EDF scheduling is provided.
- By means of simulation, we demonstrate significant capacity savings of our algorithms compared to the classical EDF algorithm. Finally, we provide approximations which make our theoretical results applicable to practical energy harvesting systems.

6.3 System Model

The chapter deals with a scheduling scenario depicted in Fig. 6.2. At some time t, an energy source harvests ambient energy and converts it into electrical power $P_S(t)$. This power can be stored in a device with capacity C. The stored energy is denoted as $E_C < C$. On the other hand, a computing device drains power $P_D(t)$ from the storage and uses it to process tasks with arrival time a_i, energy demand e_i and deadline d_i. We assume that only one task is executed at time t and preemptions are allowed.

The problem statement presented in this section comprises two major constraints which have to be satisfied: First, tasks can be processed exclusively with energy E_S generated by the energy source. And second, timing constraints in terms of tasks' deadlines d_i must be respected. For this purpose, two degrees of freedom can be exploited. The scheduler may decide which task J_i of all ready tasks to execute and what amount of power P_D to assign to this task. The following subsections define the relations between these quantities in more detail.

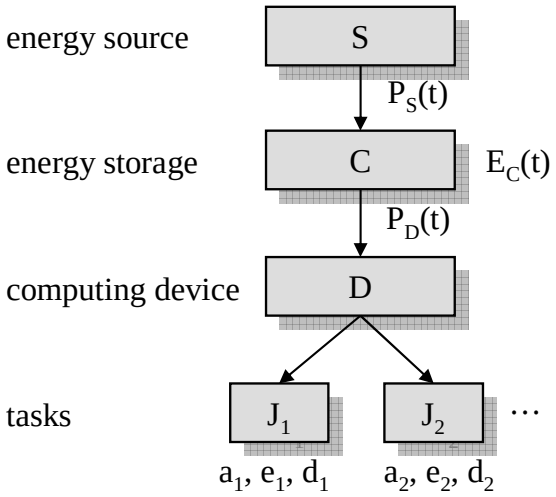


Figure 6.2: System model: Energy-driven scheduling scenario

6.3.1 Energy source

Many environmental power sources are highly variable with time. Hence, in many cases some charging circuitry is necessary to optimize the charging process and increase the lifetime of the storage device. In our model, the power P_S incorporates all losses caused by power conversion as well as charging process. In other words, we denote $P_S(t)$ the charging power that is actually fed into the

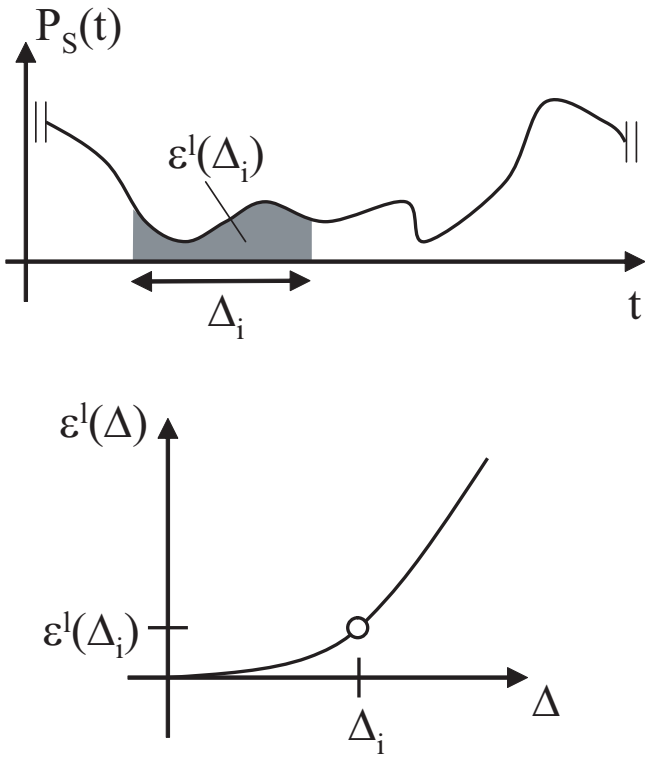


Figure 6.3: System model: Power trace and energy variability characterization curve EVCC

energy storage. The respective energy E_S in the time interval $[t_1, t_2]$ is given as

$$E_S(t_1, t_2) = \int_{t_1}^{t_2} P_S(t) \mathrm{d}t \quad .$$

In order to characterize the properties of an energy source, we define now energy variability characterization curves (EVCC) that bound the energy harvested in a certain interval Δ: The EVCCs $\epsilon^l(\Delta)$ and $\epsilon^u(\Delta)$with $\Delta \geq 0$ bound the range of possible energy values E_S as follows:

$$\epsilon^l(t_2 - t_1) \leq E_S(t_1, t_2) \leq \epsilon^u(t_2 - t_1) \quad \forall t_2 > t_1$$

Given an energy source, e.g., a solar cell mounted in a building or outside, the EVCCs provide guarantees on the produced energy. For example, the lower curve denotes that for any time interval of length Δ, the produced energy is at least $\epsilon^l(\Delta)$ (see Fig. 6.3). Three possible ways of deriving an EVCC for a given scenario are given below:

- A sliding window of length Δ is used to find the minimum/maximum energy produced by the energy source in any time interval $[t_1, t_2)$ with

$t_2 - t_1 = \Delta$. To this end, one may use a long power trace or a set of traces that have been measured. Since the resulting EVCC bounds only the underlying traces, these traces must be selected carefully and have to be representative for the assumed scenario.

- The energy source and its environment is formally modeled and the resulting EVCC is computed.
- Approximations to EVCCs can be determined on-line by using appropriate measurement and estimation methods, see Section 6.7.1.

In Section 6.5, the lower EVCC ϵ^l will be used in an admittance test which decides, whether a task set is schedulable given a certain energy source. Furthermore, both EVCCs will serve as energy predictors for the algorithms simulated in Section 6.6.

6.3.2 Energy storage

We assume an ideal energy storage that may be charged up to its capacity C. According to the scheduling policy used, power $P_D(t)$ and the respective energy $E_D(t_1, t_2)$ is drained from the storage to execute tasks. If no tasks are executed and the storage is consecutively replenished by the energy source, an energy overflow occurs. Consequently, we can derive the following constraints

$$0 \leq E_C(t) \leq C \qquad \forall t$$

$$E_C(t_2) \leq E_C(t_1) + E_S(t_1, t_2) - E_D(t_1, t_2) \qquad \forall t_2 > t_1$$

and therefore

$$E_D(t_1, t_2) \leq E_C(t_1) + E_S(t_1, t_2) \qquad \forall t_2 > t_1$$

6.3.3 Task scheduling

As illustrated in Fig. 6.2, we utilize the notion of a computing device that assigns energy E_C from the storage to dynamically arriving tasks. We assume that the power consumption $P_D(t)$ is limited by some maximum value P_{max}. In other words, the processing device determines at any point in time how much power it uses, that is

$$0 < P_D(t) < P_{max}\,.$$

We assume tasks to be independent from each other and preemptive. More precisely, the currently active task may be preempted at any time and have its execution resumed later, at no additional cost. If the node decides to assign power $P_i(t)$ to the execution of task J_i during the interval $[t_1, t_2]$, we denote the corresponding energy

$$E_i(t_1, t_2) = \int_{t_1}^{t_2} P_i(t)\mathrm{d}t \quad .$$

The effective starting time s_i and finishing time f_i of task i are dependent on the scheduling strategy used: A task starting at time s_i will finish as soon as the required amount of energy e_i has been consumed by it. We can write

$$f_i = \min\left\{t : E_i\left(s_i, t\right) = e_i\right\} .$$

The actual running time $(f_i - s_i)$ of a task i directly depends on the amount of power $P_i(t)$ which is driving the task during $s_i \leq t \leq f_i$. At this, the energy demand e_i of a task is independent from the power P_i used for its execution. Note that we are *not* using energy-saving techniques like Dynamic Voltage Scaling (DVS), where $e_i = \mathrm{f}(P_i)$. In our model, power P_i and execution time w_i behave inversely proportional: The higher the power P_i, the shorter the execution time w_i. In the best case, a task may finish after the execution time $w_i = \frac{e_i}{P_{max}}$ if it is processed without interrupts and with the maximum power P_{max}.

Current hardware technology does not support variable power consumption as described above. So clearly, the continuous task processing model presented in this section is idealized. However, a microprocessor for example may stepwise advance a task by switching power on ($P_i = P_{max}$) and off ($P_i = 0$). By tuning the so-called duty cycle accordingly, devices can approximate the average power $0 \leq \overline{P}_i \leq P_{max}$. For a more detailed discussion about practical task processing and the system model in general, see Section 6.7.

6.4 Lazy Scheduling Algorithms LSA

After having described our modeling assumptions, we will now state and prove optimal scheduling algorithms. In subsection 6.4.1, we will start with the analysis of a simplified scheduling scenario where tasks need only energy as computation resource but may execute in zero time. By disregarding the computation resource time, we focus on the energy-driven nature of the scheduling scenario presented in this chapter. In Section 6.4.2, we will consider finite execution times as well and construct a more general algorithm which manages to optimally trade off energy and time constraints. Theorems which prove op-

timality of both algorithms will follow in subsection 6.4.3.

6.4.1 Simplified Lazy Scheduling

We start with a node with infinite power $P_{max} = +\infty$. As a result, a task's execution time w_i collapses to 0 if the available energy E_C in the storage is equal to or greater than the task's energy demand e_i. This fact clearly simplifies the search for an adequate scheduling algorithm but at the same time contributes to the understanding of the problem.

As already indicated in the introduction, the naive approach of simply scheduling tasks with the EDF algorithm may result in unnecessary deadline violations, see Fig. 6.1. It may happen, that after the execution of task "1" another task "2" with an earlier deadline arrives. If now the required energy is not available before the deadline of task "2", EDF scheduling produces a deadline violation. If task "1" would hesitate instead of executing directly, this deadline violation might be avoidable. These considerations directly lead us to the principle of *Lazy Scheduling*: Gather environmental energy and process tasks only if it is necessary.

The *Lazy Scheduling Algorithm* LSA-I for $P_{max} = \infty$ shown below attempts to schedule a set of tasks J_i, $i \in Q$ such that deadlines are respected. Therefore, the processing device has to decide between three power modes. The node may process tasks with the maximal power $P_D(t) = P_{max}$ or not at all ($P_D(t) = 0$). In between, the node may choose to spend only the currently incoming power $P_S(t)$ from the harvesting unit on tasks. The algorithm is based on the three following rules:

- **Rule 1**: If the current time t equals the deadline d_j of some arrived but not yet finished task J_j, then finish its execution by draining energy $(e_j - E_j(a_j, t))$ from the energy storage instantaneously.
- **Rule 2**: We must not waste energy if we could spend it on task execution. Therefore, if we hit the capacity limit ($E_C(t) = C$) at some times t, we execute the task with the earliest deadline using $P_D(t) = P_S(t)$.
- **Rule 3**: Rule 1 overrules Rule 2.

Note that LSA-I degenerates to an *earliest deadline first* (EDF) policy, if $C = 0$. On the other hand, we find an *as late as possible* (ALAP) policy for the case of $C = +\infty$.

6.4.2 General Lazy Scheduling

The LSA-I algorithm instantaneously executes a task at its deadline. However, with limited power consumption P_D and finite, minimal computation times

Lazy Scheduling Algorithm LSA 1 ($P_{max} = \infty$)

Input: maintain a set of indices $i \in Q$ of all ready but not finished tasks J_i
$P_D(t) \Leftarrow 0$;
while (true) **do**
$d_j \Leftarrow \min\{d_i : i \in Q\}$;
process task J_j with power $P_D(t)$;
$t \Leftarrow$ current time;
if $t = a_k$ **then** add index k to Q;
if $t = f_j$ **then** remove index j from Q;
if $t = d_j$ **then** $E_C(t) \Leftarrow E_C(t) - e_j + E_j(a_j, t)$;
remove index j from Q;
$P_D(t) \Leftarrow 0$;
if $E_C(t) = C$ **then** $P_D(t) \Leftarrow P_S(t)$;

$w_i = \frac{e_i}{P_{max}}$ a general algorithm has to determine earlier starting times $s_i \leq d_i$ in order to respect deadlines. In the following, we sketch how to determine optimal starting times s_i that balance the *time* and *energy* constraints for each single task J_i. For a more detailed derivation of the starting times s_i the reader is referred to [89].

At first sight, starting a task as late as possible (ALAP) seems to be a promising approach. The upper plots in Fig. 6.4 display a straightforward ALAP-translation of the starting time for task "1": To fulfill its time condition, task "1" begins to execute at starting time $s_1 = d_1 - \frac{e_1}{P_{max}}$. As illustrated, it may happen that shortly after s_1 an unexpected task "2" arrives. Assume that this unexpected task "2" is nested in task "1", i.e., it also has an earlier deadline than "1". This scenario inevitably leads to a deadline violation, although plenty of energy is available. This kind of timing conflict can be solved by shifting s_1 to earlier times and thereby reserving time for the unpredictable task "2" (see lower plots Fig. 6.4). But starting earlier, we risk to "steal" energy that might be needed at later times (compare Fig. 6.1). According to the "lazy" principle we have to take care that we don't start *too* early.

From the above example, we learned that it may be disadvantageous to prearrange a starting time in such a way, that the stored energy E_C cannot be used before the deadline of a task. If the processing device starts running at time s_i with P_{max} and cannot consume all the available energy before the deadline d_i, time conflicts may occur. On the other hand, if we remember the introductory example in Fig. 6.1, energy conflicts are possible if the stored energy $E_C(t)$ is 0 at some time $t < d_i$. Hence we can conclude the following: The optimal starting time s_i must guarantee, that the processor *could* continuously use P_{max} in the interval $[s_i, d_i]$ and empty the energy storage $E_C(d_i) = 0$ exactly at time d_i. Before the optimal starting time s_i, the scheduler has to conserve energy and keep the storage level E_C as high as possible.

A necessary prerequisite for the calculation of the optimal starting time s_i is the knowledge of the incoming power flow $P_S(t)$ for all future times $t \leq d_i$.

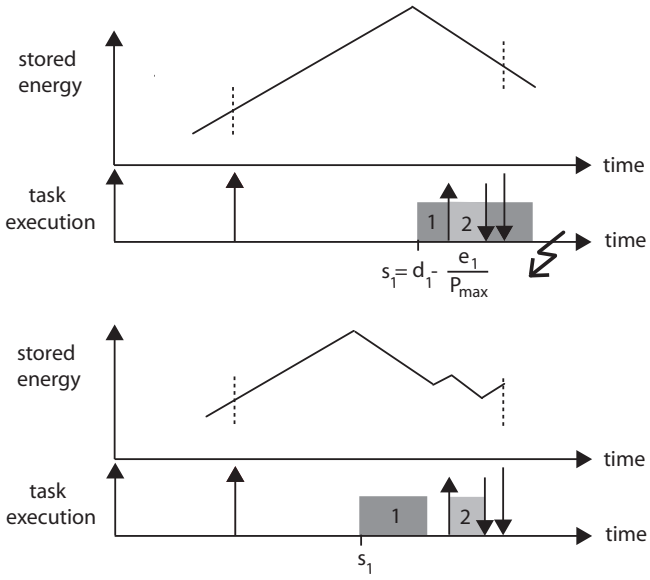


Figure 6.4: ALAP vs. lazy scheduling

Finding useful predictions for the power $P_S(t)$ can be done, e.g., by analyzing traces of the harvested power, as we will see in Section 6.6. In addition, we assume that $P_S(t) < P_{max}$, that is, the incoming power $P_S(t)$ from the harvesting unit never exceeds the power consumption P_{max} of a busy node. Besides from being a realistic model in many cases, this assumption ensures that no energy is wasted if the energy storage is full and the system is running with P_{max}.

To calculate the optimal starting time s_i, we determine the maximum amount of energy $E_C(a_i)+E_S(a_i, d_i)$ that may be processed by the node before d_i. Next, we compute the minimum time required to process this energy without interruption and shift this time interval of continuous processing just before the deadline d_i. In other words, we calculate the starting time s_i^* as

$$s_i^* = d_i - \frac{E_C(a_i) + E_S(a_i, d_i)}{P_{max}}.$$

If now the energy storage is filled before s_i^*, it is reasonable to advance task J_i with power P_S in order to avoid an energy overflow (compare Rule 2 of LSA-I). However, this also means that not all energy $E_C(a_i) + E_S(a_i, d_i)$ can be processed continuously in $[s_i^*, d_i]$ and we find $E_C(t) = 0$ at some time $t < d_i$. Thus a better starting time s_i' allows for the reduced amount of energy $C + E_S(s_i', d_i)$ which is processible in this situation:

$$s_i' = d_i - \frac{C + E_S(s_i', d_i)}{P_{max}}$$

By choosing the maximum of s_i^* and s_i' we find the optimal starting time

$$s_i = \max\left(s_i', s_i^*\right),$$

which precisely balances energy and time constraints for task J_i.

The pseudo-code of a general Lazy Scheduling Algorithm LSA-II is shown below. It is based on the following rules:

- **Rule 1**: EDF scheduling is used at time t for assigning the processor to all waiting tasks with $s_i \leq t$. The currently running task is powered with $P_D(t) = P_{max}$.
- **Rule 2**: If there is no waiting task i with $s_i \leq t$ and if $E_C(t) = C$, then all incoming power P_S is used to process the task with the smallest deadline ($P_D(t) = P_S(t)$).

Lazy Scheduling Algorithm LSA 2 ($P_{max} = \text{const.}$)

Input: maintain a set of indices $i \in Q$ of all ready but not finished tasks J_i
$P_D(t) \Leftarrow 0$;
while (true) **do**
  $d_j \Leftarrow \min\{d_i : i \in Q\}$;
  calculate s_j;
  process task J_j with power $P_D(t)$;
  $t \Leftarrow$ current time;
  if $t = a_k$ **then** add index k to Q;
  if $t = f_j$ **then** remove index j from Q;
  if $E_C(t) = C$ **then** $P_D(t) \Leftarrow P_S(t)$;
  if $t \geq s_j$ **then** $P_D(t) \Leftarrow P_{max}$;

Although it is based on the knowledge of the future incoming energy E_S, LSA-II remains an online algorithm. The calculation of s_i must be performed once the scheduler selects the task with the earliest deadline. If the scheduler is not energy-constraint, i.e., if the available energy is more than the device can consume with power P_{max} within $[a_i, d_i]$, the starting time s_i will be before the current time t. Then, the resulting scheduling policy is EDF, which is reasonable, because only time constraints have to be satisfied. If, however, the sum of stored energy E_C plus generated energy E_S is small, the scheduling policy changes towards an ALAP policy. In doing so, LSA avoids spending scarce energy on the "wrong" tasks too early.

In summary, LSA-II can be classified as an energy-clairvoyant adaptation of the Earliest Deadline First Algorithm. It changes its behaviour according to the amount of available energy, the capacity C as well as the maximum power consumption P_{max} of the device. For example, the lower the power P_{max} gets, the greedier LSA-II gets. On the other hand, high values of P_{max} force LSA-II to hesitate and postpone the starting time s_i. For $P_{max} = \infty$, all starting times collapse to the respective deadlines, and we identify LSA-I as a special case

of LSA-II. In the remainder of the chapter, we will solely consider the general LSA-II algorithm derived in this section. From now on, we will denote this algorithm LSA.

6.4.3 Optimality of Lazy Scheduling

In this section, we will show that Lazy Scheduling algorithms optimally assign power P_D to a set of tasks. For this purpose, we formulate Theorem 1 and Theorem 2 which show that LSA makes the best use of the available time and energy, respectively. With the help of Theorem 1 and 2, we proof optimality of LSA in Theorem 3.

The scheduling scenario presented in this chapter is inherently energy-driven. Hence, a scheduling algorithm yields a deadline violation if it fails to assign energy e_i to a task before its deadline d_i. We distinguish between two types of deadline violations:

- A deadline cannot be respected since the time is not sufficient to execute available energy with power P_{max}. At the deadline, unprocessed energy remains in the storage and we have $E_C(d) > 0$. We call this the **time limited** case.
- A deadline violation occurs because the required energy is simply not available at the deadline. At the deadline, the battery is exhausted (i.e., $E_C(d) = 0$). We denote the latter case **energy limited**.

For the following theorems to hold we suppose that at initialization of the system, we have a full capacity, i.e., $E_C(t_i) = C$. Furthermore, we call the computing device *idle* if no task i is running with $s_i \leq t$.

Let us first look at the time limited case.

Theorem 1 *Let us suppose that the LSA algorithm schedules a set of tasks. At time d the deadline of a task J with arrival time a is missed and $E_C(d) > 0$. Then there exists a time t_1 such that the sum of execution times $\sum_{(i)} w_i = \sum_{(i)} \frac{e_i}{P_{max}}$ of tasks with arrival and deadline within time interval $[t_1, d]$ exceeds $d - t_1$.*

Proof 1 *Let us suppose that t_0 is the maximal time $t_0 \leq d$ where the processor was idle. Clearly, such a time exists.*

We now show, that at t_0 there is no task i with deadline $d_i \leq d$ waiting. At first, note that the processor is constantly operating on tasks in time interval $(t_0, d]$. Suppose now that there are such tasks waiting and task i is actually the one with the earliest deadline d_i among those. Then, as $E_C(d) > 0$ and because of the construction of s_i, we would have $s_i < t_0$. Therefore, the processor would actually process task i at time t_0 which is a contradiction to the idleness.

Because of the same argument, all tasks i arriving after t_0 with $d_i \leq d$ will have $s_i \leq a_i$. Therefore, LSA will attempt to directly execute them using an EDF strategy.

Now let us determine time $t_1 \geq t_0$ which is the largest time $t_1 \leq d$ such that the processor continuously operates on tasks i with $d_i \leq d$. As we have $s_i \leq a_i$ for all of these tasks and as the processor operates on tasks with smaller deadlines first (EDF), it operates in $[t_1, d]$ only on tasks with $a_i \geq t_1$ and $d_i \leq d$. As there is a deadline violation at time d, we can conclude that $\sum_{(i)} w_i > d - t_1$ where the sum is taken over all tasks with arrival and deadline within time interval $[t_1, d]$.

It can be shown that a related result holds for the energy limited case, too.

Theorem 2 *Let us suppose that the LSA algorithm schedules a set of tasks. At time d the deadline of a task J with arrival time a is missed and $E_C(d) = 0$. Assume further, that deadline d is the first deadline of the task set that is violated by LSA. Then there exists a time t_1 such that the sum of task energies $\sum_{(i)} e_i$ of tasks with arrival and deadline within time interval $[t_1, d]$ exceeds $C + E_S(t_1, d)$.*

Proof 2 *Let time $t_1 \leq d$ be the largest time such that (a) $E_C(t_1) = C$ and (b) there is no task i waiting with $d_i \leq d$. Such a time exists as one could at least use the initialization time t_i with $E_C(t_i) = C$. As t_1 is the last time instance with the above properties, we can conclude that everywhere in time interval $[t_1, d]$ we either have (a) $E_C(t) = C$ and there is some task i waiting with $d_i \leq d$ or we have (b) and $E_C(t) < C$.*

It will now be shown that in both cases a) and b), energy is not used to advance any task j with $d_j > d$ in time interval $[t_1, d]$. Note also, that all arriving energy $E_S(t_1, d)$ is used to advance tasks.

In case a), all non-storable energy (i.e. all energy that arrives from the source) is used to advance a waiting task, i.e., the one with the earliest deadline $d_i \leq d$. In case b), the processor would operate on task J with $d_j > d$ if there is some time $t_2 \in [t_1, d]$ where there is no other task i with $d_i \leq d$ waiting and $s_j \leq t_2$. But s_j is calculated such that the processor could continuously work until d_j. As $d_j > d$ and $E_C(d) = 0$ this can not happen and $s_j > t_2$. Therefore, also in case b) energy is not used to advance any task j with $d_j > d$.

As there is a deadline violation at time d, we can conclude that $\sum_{(i)} e_i > C + E_C(t_1, d)$ where the sum is taken over all tasks with arrival and deadline within time interval $[t_1, d]$.

From the above two theorems we draw the following conclusions: First, in the **time limited** case, there exists a time interval before the violated deadline with a larger accumulated computing time request than available time. And second, in the **energy limited** case, there exists a time interval before the violated deadline with a larger accumulated energy request than what can be provided at best. These considerations lead us to one of the major results of the thesis:

Theorem 3 (Optimality of Lazy Scheduling) *Let us consider a system characterized by a capacity C and power P_{max} driven by the energy source E_S. If LSA cannot schedule a given task set, then no other scheduling algorithm is able to schedule it. This holds even if the other algorithm knows the complete task set in advance.*

Proof 3 *The proof follows immediately from Theorems 1 and 2. Assume a set of tasks is scheduled with LSA and at time d the deadline of task J is missed. Assume further, that deadline d is the first deadline of the task set that is violated by LSA. In the following, we distinguish between the case where the energy at the deadline is $E_C(d) > 0$ and $E_C(d) = 0$, respectively.*

In the first case, according to Theorem 1, there exists a time t_1 such that the sum of execution times $\sum_{(i)} w_i = \sum_{(i)} \frac{e_i}{P_{max}}$ of tasks with arrival and deadline within time interval $[t_1, d]$ exceeds $d - t_1$. Here, knowing arrival times, energy demands and deadlines in advance does not help, since every scheduling algorithm will at least violate one deadline in $[t_1, d]$.

In the energy limited case with $E_C(d) = 0$, according to Theorem 2, there exists a time t_1 such that the sum of task energies $\sum_{(i)} e_i$ of tasks with arrival and deadline within time interval $[t_1, d]$ exceeds $C + E_S(t_1, d)$. Hence, no algorithm can hold deadline d without violating an earlier deadline in $[t_1, d]$. This holds also for omniscient scheduling algorithms, since (a) at the beginning of the critical interval $[t_1, d]$, the energy level may be $E_C(t_1) = C$ at most and (b) the execution of the critical tasks can start at time t_1 at the earliest.

So every time LSA violates deadline d, we have either the time limited case ($E_C(d) > 0$) or the energy limited case ($E_C(d) = 0$). Since in both cases it is impossible for another algorithm to respect deadline d and all earlier deadlines simultaneously, we conclude that LSA is optimal.

If we can guarantee that there is no time interval with a larger accumulated computing time request than available time and no time interval with a larger accumulated energy request than what can be provided at best, then the task set is schedulable. This property will be used to determine the admittance test described in the next section.

On the other hand, given a task set and a certain energy source $E_S(t)$ we can also make a statement about the necessary hardware requirements of the sensor node: Due to its optimality, LSA requires the minimum processing power P_{max} and the minimum capacity C necessary to avoid deadline violations:

Theorem 4 (Optimal tuple $(C\,; P_{max})$) *Let us assume a given task set has to be scheduled using an energy source E_S. Among all algorithms, LSA requires the minimum capacity C and the minimum power P_{max} that are necessary to successfully schedule the task set.*

Proof 4 *We proceed by contradiction. Let us denote C and P_{max} the minimum capacity and the minimum processing power which are needed to schedule a given task set with LSA. Assume that an adversary algorithm succeeds to schedule the task set with some $C' < C$ or $P'_{max} < P_{max}$ given the same energy source. This means the adversary algorithm can schedule the respective task set with (C', P'_{max}) and LSA cannot. This however contradicts the optimality of LSA according to Theorem 3.*

The admittance test in the next section will allow us to explicitly determine the minimum values of C and P_{max} for LSA scheduling.

6.5 Admittance Test

6.5.1 Lazy Scheduling Algorithm

In this section, we will determine an offline schedulability test in case of periodic, sporadic or even bursty sets of tasks. In particular, given an energy source with lower EVCC $\epsilon^l(\Delta)$, the device parameters $(C\,; P_{max})$ and a set of periodic tasks J_i, $i \in I$ with period p_i, *relative* deadline d_i and energy demand e_i, we would like to determine whether all deadlines can be respected.

To this end, let us first define for each task its arrival curve $\alpha(\Delta)$ which denotes the maximal number of task arrivals in any time interval of length Δ. The concept of arrival curves to describe the arrival patterns of sets of tasks is well known (request bound functions) and has been used explicitly or implicitly in, e.g., [91] or [92]. To simplify the discussion, we limit ourselves to periodic tasks, but the whole formulation allows to deal with much more general classes (sporadic or bursty) as well.

In case of a periodic task set, we have for periodic task J_i, see also Fig. 6.5:

$$\alpha_i(\Delta) = \left\lceil \frac{\Delta}{p_i} \right\rceil \quad \forall \Delta \geq 0$$

In order to determine the maximal energy demand in any time interval of length Δ, we need to maximize the accumulated energy of all tasks having their arrival and deadline within an interval of length Δ. To this end, we need to shift the corresponding arrival curve by the relative deadline. We are doing this since the actual energy demand becomes due at the deadline of the respective task. In case of a periodic task J_i, this simply leads to:

$$\overline{\alpha}_i(\Delta) = \begin{cases} e_i \cdot \alpha_i(\Delta - d_i) & \Delta > d_i \\ 0 & 0 \leq \Delta \leq d_i \end{cases}$$

In case of several periodic tasks that arrive concurrently, the total demand curve $A(\Delta)$ (called demand-bound function in [91]) can be determined by just

adding the individual contributions of each periodic task, see Fig. 6.5:

$$A(\Delta) = \sum_{i \in I} \overline{\alpha}_i(\Delta)$$

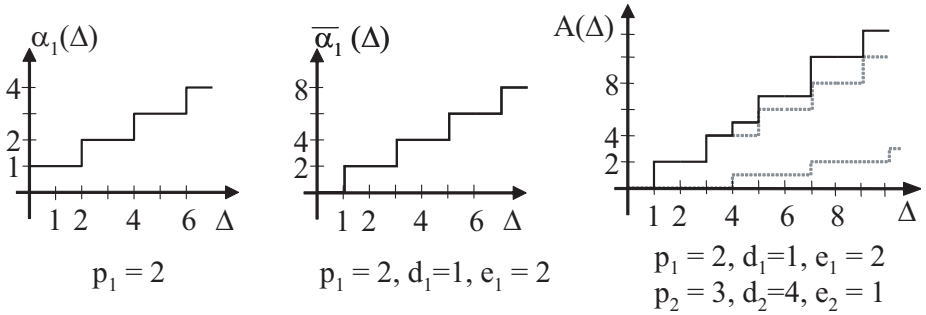


Figure 6.5: Examples of an arrival curve $\alpha_i(\Delta)$, a demand curve $\overline{\alpha}_i(\Delta)$ and a total demand curve $A(\Delta)$ in case of periodic tasks

Using the above defined quantities, we can formulate a schedulability test for the LSA algorithm that can be applied to a quite general class of tasks specifications.

Theorem 5 (LSA Schedulability Test) *A given set of tasks J_i, $i \in I$ with arrival curves $\alpha_i(\Delta)$, energy demand e_i and relative deadline d_i is schedulable under the energy-driven model with initially stored energy C, if and only if the following condition holds*

$$A(\Delta) \leq \min\left(\epsilon^l(\Delta) + C\,,\, P_{max} \cdot \Delta\right) \quad \forall \Delta > 0$$

Here, $A(\Delta) = \sum_{i \in I} e_i \cdot \alpha_i(\Delta - d_i)$ denotes the total energy demand of the task set in any time interval of length Δ, $\epsilon^l(\Delta)$ the energy variability characterization curve of the energy source, C the capacity of the energy storage and P_{max} the maximal processing power of the system. In case of periodic tasks we have $A(\Delta) = \sum_{i \in I} e_i \cdot \left\lceil \frac{\Delta - d_i}{p_i} \right\rceil$.

Proof 5 *The proof of the if direction is omitted, since it is a direct consequence of Theorems 1 and 2. We just prove the only-if direction.*

Remember that the total demand curve $A(\Delta)$ denotes the maximal energy demand of tasks in any interval $[t_1, t_2]$ of length Δ. It equals the maximal accumulated energy of tasks having their arrival **and** *deadline within $[t_1, t_2]$. Therefore, in order to satisfy all deadlines for these tasks, at least energy $A(t_2 - t_1)$ must be available.*

Let us suppose that the condition in Theorem 5 is violated for some Δ due to missing energy. Let us suppose also that the task arrival curve and the energy variability characterization curve are strict, i.e., there exists some time interval $[t_1, t_2]$ where the energy demand is $A(t_2 - t_1)$ and at the same time the energy $E_S(t_2 - t_1)$ is received. Then in time interval $[t_1, t_2]$ with $\Delta = t_2 - t_1$ the difference between the

energy demand and the received energy is larger than the maximal stored energy C as $A(\Delta) > \epsilon^l(\Delta) + C$. As a result, the task set is not schedulable.

On the other hand, whenever the demanded computation time $\frac{A(\Delta)}{P_{max}}$ of a task set in the interval Δ is larger than the interval itself, a task set is not schedulable. Therefore it is evident, that both the energy condition $A(\Delta) \leq \epsilon^l(\Delta) + C$ and the time condition $A(\Delta) \leq P_{max} \cdot \Delta$ must be fulfilled in order to avoid deadline violations.

Theorem 5 tells us that we can decouple energy and time constraints if we have to decide whether a task set is schedulable or not. On the one hand, only if

$$P_{max} \geq \max_{0 \leq \Delta} \left(\frac{A(\Delta)}{\Delta} \right) ,$$

the system is fast enough to process the given task set. This condition is independent of the energy provided by the environmental source (i.e. ϵ^l) and the capacity of the storage device. Even increasing the capacity does not help. If a task set however satisfies the time constraint, the role of the capacity C as a design parameter for the energy harvesting system becomes important.

Suppose now that the time constraint is fulfilled. For this case, Theorem 5 states that the capacity C needed to schedule a task set with $A(\Delta)$ using a source with $\epsilon^l(\Delta)$ is constrained by

$$C \geq \max_{0 \leq \Delta} \left(0, A(\Delta) - \epsilon^l(\Delta)\right) .$$

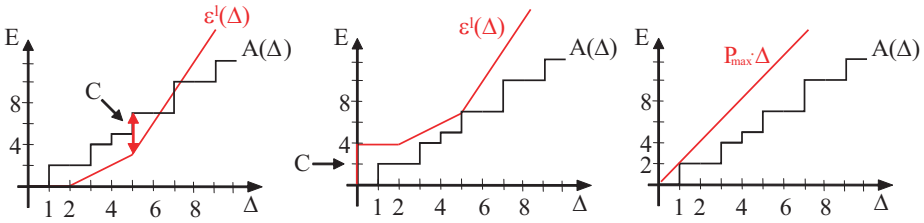


Figure 6.6: Determining the optimal tuple $(C; P_{max})$ according to theorem 5

Fig. 6.6 illustrates an example schedulability test. The left diagram displays the total demand curve $A(\Delta)$ for two periodic tasks with p_1=2, d_1=1, e_1=2 and p_2=3, d_2=4, e_2=1. Furthermore, the EVCC $\epsilon^l(\Delta)$ is given by a piecewise linear function using three pieces (0,0,0),(2,0,1),(5,3,3), where each piece i is defined by the triple of the form (initial Δ_i, initial $\epsilon^l(\Delta_i)$, slope of piece i). Now, the maximal difference between the total demand curve A and the EVCC ϵ^l can be computed. It has value 4 and is obtained at $\Delta = 5$. Therefore, one can conclude that the set of tasks can be scheduled (with LSA) using a minimal capacity $C = 4$. The respective schedulability test with $C = 4$ is shown in the middle diagram of Fig. 6.6. As displayed in the right diagram, power $P_{max} = 2$

is required to fulfill the time condition in theorem 5.

As a last point to mention, let us consider the middle diagram in Fig. 6.6 once again. Regarding the slopes of the curves, we can guess that A and ϵ^l won't intersect after the critical time interval of length 5. Formally, this is because the minimum average power $\lim_{\Delta\to\infty} \frac{\epsilon^l(\Delta)}{\Delta}$ is higher than the maximum average power demand $\lim_{\Delta\to\infty} \frac{A(\Delta)}{\Delta}$ of the task set. Simply stated, the average harvested power is higher than the average power demand of the application. It becomes evident that an optimal algorithm, like LSA, can only optimize the short-term behaviour of the system by suitable power management. LSA achieves the minimum capacity C needed to temporarily buffer energy for single tasks, but on the long run the average of power P_D is dictated by the task set.

6.5.2 Comparison to EDF

It is useful to formally compare LSA with the well know EDF algorithm in terms of the schedulability region and the required capacity C. In order to simplify the discussion, we will investigate an energy limited scenario only, see Section 6.4.1 and 6.4.3. Then we can state the following result:

Theorem 6 (EDF Schedulability Test) *A given set of tasks J_i, $i \in I$ with arrival curves $\alpha_i(\Delta)$, energy demand e_i and relative deadline d_i is schedulable with initially stored energy C, only if the following condition holds for any deadline d_k, $k \in I$:*

$$\sum_{i\in I} e_i \cdot \alpha_i(\Delta - d_k) \leq C + \epsilon^l(\Delta) \quad \forall \Delta > 0$$

In case of periodic tasks with period p_i we have $\alpha_i(\Delta) = \lceil \frac{\Delta}{p_i} \rceil$.

Proof 6 *Remember that the left hand side of the condition denotes the maximal energy used by all the tasks in any interval $[t_1, t_2]$ of length $t_2 - t_1 = \Delta - d_k$. There is an instance of task arrivals compliant with the arrival curves α_i such that task J_k arrives at time t_2, i.e. by correctly adjusting the phase of all instances of task J_k. In this case, the deadline of the task instance arriving at t_2 is $t_2 + d_k$. In order to be able to execute this instance within its deadline, the available energy in any interval $[t_1, t_2 + d_k]$ must be larger than $\sum_{i\in I} e_i \cdot \alpha_i(t_2 - t_1)$, i.e. the energy used by tasks arriving in $[t_1, t_2]$. The maximal energy available in $[t_1, t_2 + d_k]$ is in the worst case given by $C + \epsilon^l(t_2 + d_k - t_1)$. Replacing $t_2 - t_1$ by $\Delta - d_k$ yields the desired result.*

The strongest bound is obtained by using the task J_k with the smallest deadline $d^{min} = \min_{i\in J}\{d_i\}$. Comparing Theorems 5 and 6 in the energy-constraint case we obtain the two constraints $\sum_{i\in I} e_i \cdot \alpha_i(\Delta - d^{min}) \leq C + \epsilon^l(\Delta)$ for EDF and $\sum_{i\in I} e_i \cdot \alpha_i(\Delta - d_i) \leq C + \epsilon^l(\Delta)$ for LSA. Clearly, EDF has a smaller

schedulability region as $\sum_{i\in I} e_i \cdot \alpha_i(\Delta - d^{min}) \geq \sum_{i\in I} e_i \cdot \alpha_i(\Delta - d_i)$ for all $\Delta \geq 0$.

Finally, let us derive specialized results in the case of periodic tasks J_i with $p_i = d_i$ (period equals deadline) and a simple energy variability characterization curve $\epsilon^l(\Delta)$ shown in Fig. 6.7 with $\epsilon^l(\Delta) = \max\{\sigma \cdot (\Delta - \rho), 0\}$.

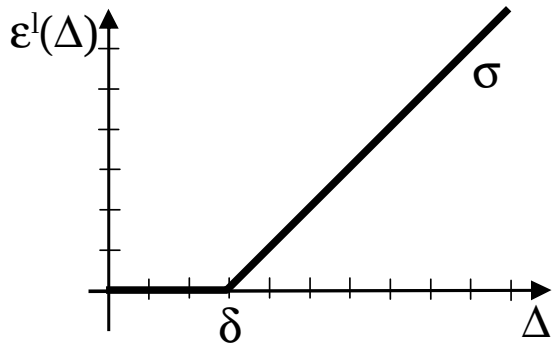


Figure 6.7: Simple EVCC for comparing EDF and LSA scheduling

We also suppose that the available average power σ from the energy source is sufficient to support the long term power demand $\bar{\sigma}$ of the task set

$$\sigma \geq \bar{\sigma} = \sum_{i\in I} \frac{e_i}{p_i}$$

as otherwise, deadline violations are unavoidable. In the following comparison, we suppose that the energy source has a minimal average power, i.e. $\sigma = \bar{\sigma}$, i.e. it is as weak as possible. Under these assumptions (periodic task with periods equal deadlines, energy-limited scenario) and using the results of Theorems 5 and 6, one can compute the minimal possible capacities of the energy storage for the two scheduling methods LSA and EDF as follows:

$$C^{EDF} = \sum_{i\in I} e_i + (\delta - p^{min}) \cdot \bar{\sigma} \qquad ; \qquad C^{LSA} = \delta \cdot \bar{\sigma}$$

Therefore, the relative gain in the necessary storage capacity between the two scheduling methods can be quantified and bounded by

$$0 \leq \frac{C^{EDF} - C^{LSA}}{C^{LSA}} = \frac{1}{\delta \cdot \bar{\sigma}}(\sum_{i\in I} e_i + p^{min} \cdot \bar{\sigma}) \leq \frac{p^{max} - p^{min}}{\delta}$$

For the bounds we use the fact that $p^{min} \leq (\sum e_i)(\sum(e_i/p_i)) \leq p^{max}$ where p^{min} and p^{max} denote the minimal and maximal period of tasks J_i, respectively. In other words, the maximal relative difference in storage capacity depends on the differences between the task periods. The larger the difference between the

largest and smallest period is, the large the potential gain in storage efficiency for the LSA algorithm.

6.6 Simulation Results

In the previous section, a method to compute the minimum capacity for a certain energy source characterization ϵ^l was presented. In the following, we will call this optimal value C_{min}. The value C_{min} obtained represents a lower bound since it is obtained for energy-clairvoyant LSA scheduling. In addition, it remains unclear, which capacities C^*_{min} are required if other scheduling disciplines are applied. For this reason, we performed a simulative study to evaluate the achievable capacity savings in a more realistic scenario. The EDF algorithm – which is optimal in traditional scheduling theory– serves as a benchmark for our studies.

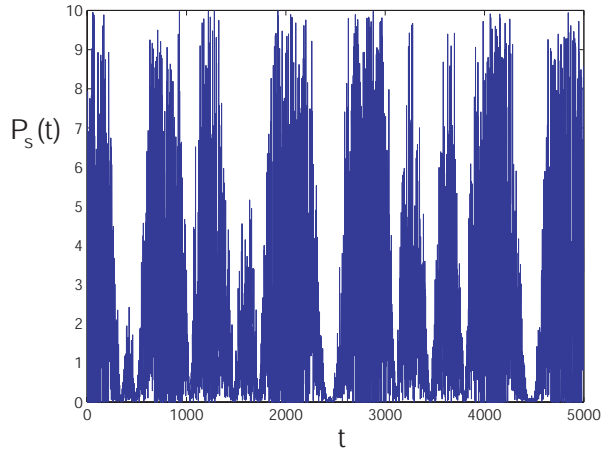
We investigated variants of LSA which utilize the measured EVCCs ϵ^l and ϵ^u to *predict* the future generated energy $E_S(t)$ for the LSA algorithm. Each time a starting time s_i has to be calculated for the task i with the earliest deadline d_i, the energy $\epsilon^l(d_i - a_i)$ (or $\epsilon^u(d_i - a_i)$) plus the stored energy $E_C(a_i)$ is assumed to be processible before the deadline.

The trace of the power source $P_S(t)$ is generated by a random number generator according to

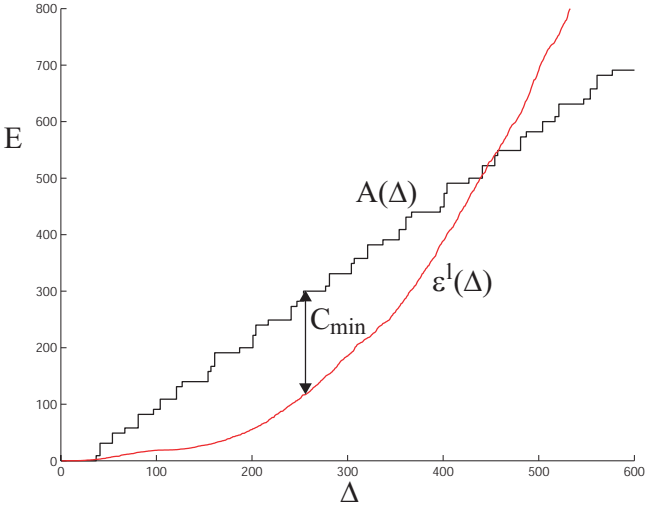
$$P_S(t) = \left| 10 \cdot N(t) \cdot \cos(\frac{t}{70\pi}) \cdot \cos(\frac{t}{100\pi}) \right| ,$$

where $N(t)$ denotes a normally distributed random variable with mean 0 and variance 1. The values of P_S have been cut off at the value $P_{S,max} = 10$. As illustrated in Fig. 6.8(a), the obtained power trace $P_S(t)$ exhibits both stochastic and deterministic, periodic behaviour. The latter is simulating day and night periods similar to those experienced by solar cells in an outdoor environment. From this trace we compute the average power $\overline{P_S}$ as well as upper and lower EVCCs ϵ^u and ϵ^l.

A task set consists of an arbitrary number of periodic tasks. Periods p are taken from a set {10, 20, 30, ... , 100}, each value having an equal probability of being selected. The initial phases φ are uniformly distributed between [0,100]. For simplicity, the relative deadline d is equal to the period p of the task. The energies e of the periodic tasks are generated according to a uniform distribution in $[0, e_{max}]$, with $e_{max} = \overline{P_S} \cdot p$.



(a) Power trace $P_S(t)$



(b) Determining C_{min} for a random task set

Figure 6.8: Calculation of C_{min} in two steps: (1) Extract $\epsilon^l(\Delta)$ from $P_S(t)$ and (2) Compute C_{min} for every task set with respective energy demand $A(\Delta)$

We define the utilization $U \in [0, 1]$ of a scheduler as

$$U = \sum_i \frac{\frac{e_i}{\overline{P_S}}}{p_i}.$$

One can interpret U as the percentage of processing time of the device if tasks are solely executed with the average incoming power $\overline{P_S}$. A system with, e.g., $U > 1$ is processing more energy than it scavenges on average and will deplete its energy reservoir.

In dependence of the generated power source $P_S(t)$, N task sets are gen-

erated which yield a certain processor utilization U. For that purpose, the number of periodic tasks in each task set is successively incremented until the intended utilization U is reached. Hence, the accuracy of the utilization U is varying $\pm 1\%$ with respect to its nominal value.

At the beginning of the simulation, the energy storage is full. We set $P_{max} = 10$. The simulation terminates after 10000 time units and is repeated for 5000 task sets. In order to show the average behaviour of all task sets in one plot, we normalized the capacities C with the respective C_{min} of the task set. Fig. 6.8(b) shows the calculation of C_{min} for a random task set.

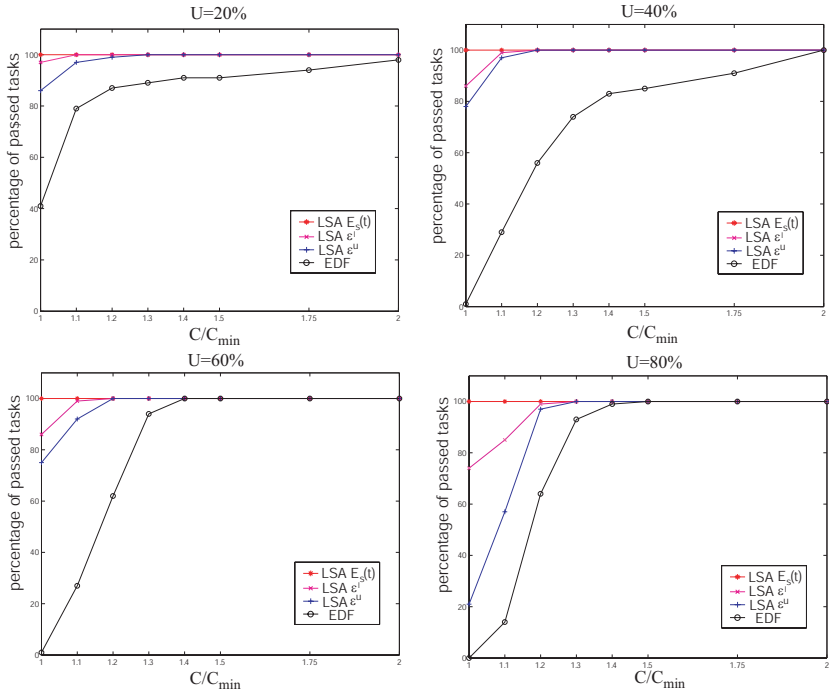


Figure 6.9: Comparison of pure LSA, LSA using ϵ^l, LSA using ϵ^u and EDF for different utilizations U

Fig. 6.9 illustrates the percentage of tasks that could be scheduled without deadline violations for different utilizations U. Clearly, no deadline violations occur for energy-clairvoyant LSA scheduling and values of $\frac{C}{C_{min}} \geq 1$. For all values of U, both approximations of LSA with ϵ^l and ϵ^u outperform the EDF algorithm, whereat the lower curve ϵ^l seems to be the better approximation. At $U = 40\%$ and $C = C_{min}$, e.g., almost no task set is schedulable with EDF. Here, LSA with ϵ^u is able to schedule $\approx 78\%$ of all task sets; LSA with ϵ^l even $\approx 85\%$.

Concerning the relative capacity savings achieved with our algorithms, we are especially interested in the smallest capacities C necessary to avoid any deadline violations. The highest savings are obtained at $U = 20\%$, where EDF needs more than $2.0 \cdot C_{min}$ to respect deadlines whereas LSA using ϵ^l shows

the same behaviour at $1.1 \cdot C_{min}$. This translates into capacity savings of $\approx 45\%$. At higher values of the utilization U these savings are decreasing, yet they are still significant: At utilizations of $U = 40\%, 60\%, 80\%$, the capacity savings of LSA with ϵ^l compared to EDF are still $\approx 40\%, 21\%, 20\%$, respectively.

Albeit randomized task sets are not necessary representative for all kind of applications, these simulation results demonstrate that significant capacity savings are possible. If the application involves bursty instead of periodic task processing, the benefits of Lazy Scheduling may be indeed even more striking: As showed in [89], a greedy algorithm like EDF may violate an arbitrary number of deadlines and may suffer from worst case scenarios. This holds in particular for sensor nodes, where the energy demands of different tasks are highly varying (e.g. communication, sensing and data processing tasks) and tasks have to satisfy various timing constraints (e.g. urgent and less urgent tasks which have to run in parallel).

6.7 Practical Considerations

The system model introduced in Section 6.3 and used throughout this chapter implies idealized modelling abstractions, which demand further explanations. Therefore, this section is dedicated to general implementation aspects and possible application scenarios

6.7.1 Energy Source Predictability

Clearly, the performance of LSA is strongly dependent on the accuracy of the predicted power $P_S(t)$ of the harvesting unit. The better the approximation, the better the algorithm performs in terms of optimality. As illustrated by the simulation results of the previous section, energy variability characterization curves (EVCC) are suitable for that purpose. Especially for small utilizations U of the sensor node, EVCCs appear to converge towards the optimal, energy-clairvoyant LSA. It should be mentioned, that the prediction of $E_S(t)$ by EVCCs may even be improved if the sensor node is learning the characteristics of the energy source adaptively: By observing energy values $E_S(\Delta')$ for past intervals Δ', the prediction for future intervals Δ can be optimized online. This extension, however, increases at the same time the computational demand of the scheduler, which is one of the advantages of using simple EVCCs.

Solar energy harvesting through photovoltaic conversion is deemed a number one candidate for the power source P_S described in our model. If we assume the sensor node to be placed in an outdoor environment, the impinging radiation is variable with time, but follows a diurnal as well as annual cycle.

Moreover, during short time intervals, the produced power P_S can be regarded as constant and sudden changes of the light intensity are improbable. Due to this specific nature of solar energy, a two-tiered prediction methodology is self-evident: On the one hand, long-run predictions must be made for less urgent tasks with rather late deadlines. Here, using exponential decaying factors to weight the history of recorded powers P_S is one possibility. An alternative is to combine daily and seasonal light conditions of the past with the knowledge about a sensor's environment and possible shadowing. One can think of a plurality of prediction mechanisms, which are clearly out of the scope of this thesis.

For urgent tasks with close deadlines within milliseconds or seconds, intelligent prediction algorithms may not be necessary. Here, tasks like, e.g., sending a few bits over the wireless channel may be planned assuming constant power $P_S(t) = P_{S,const}$ during $s_i \leq t \leq d_i$. For stationarity of the power inflow P_S the calculation of the starting time s_i for a task i simplifies to

$$s_i = d_i - \min\left(\frac{E_C(a_i) + (d_i - a_i)P_{S,const}}{P_{max}}, \frac{C}{P_{max} - P_{S,const}}\right).$$

In the worst case, a sensor node is powered by an energy source with pure stochastic behaviour. If nothing is known about this source, the currently stored energy E_S is the only indicator for making scheduling decisions. By iteratively updating the starting time $\hat{s}_i = d_i - \frac{E_C(t)}{P_{max}}$ (and thereby increasing the computational overhead) starting task i too early can be avoided. However, once the device is running with P_{max}, the incoming energy $E_S(\hat{s}_i, d_i)$ may not be processible in the remaining interval $[\hat{s}_i, d_i]$. Consequently, optimality cannot be guaranteed for this scenario since the starting time $\hat{s}_i$ is always earlier then the optimal starting time s_i.

6.7.2 Task Processing

The task processing model presented in this chapter exhibits two major assumptions:

1. We assume the power $P_D(t)$ driving a task to be *continuously* adjustable with respect to its value in $[0; P_{max}]$ as well as with respect to time t. That is, at any point in time a task can be advanced with an accurately defined amount of power.

2. We assume a *linear* relationship between the power P_D used for executing a task and the execution time w. We can say: the higher the power P_D, the shorter the execution time w.

The first modelling assumption is needed in situations when the energy storage is full ($E_C(t) = C$). In practice, there is no existing hardware that supports a continuous consumption of the scavenged power $P_S(t)$ as claimed by LSA. A microcontroller, e.g., drains roughly constant power from the battery when running a piece of program. The same holds for the radio interface. When transmitting a certain amount of data, most radios won't operate properly with unstable power supply. Therefore, we assume that the respective hardware attempts to approximate the power level of the power source by continuously switching power on ($P_D = P_{max}$) and off ($P_D = 0$). In Fig. 6.10, the achieved average power $\overline{P_D} \approx P_S$ is sketched.

It becomes evident that in an implementation, one will have to respect a certain granularity. The scheduler needs to determine when the energy storage is full and then, a task is executed for a given interval of time Δt which results in an energy consumption of $\Delta E = P_{max} \cdot \Delta t - E_S(\Delta t)$. For a microcontoller or a sensing unit, the time intervals Δt can be considered rather short while radio communication may require larger Δt due to packetized nature of the transmitted data. During the subsequent idle time, the stored energy is recovering again. In the worst case, this "duty cycling" results in a stored energy that is reduced by

$$\Delta E_{max} = P_{max} \cdot \max\{\Delta t\} - \epsilon^l(\max\{\Delta t\}))$$

in comparison to ideal LSA. With $\max\{\Delta t\}$ we denote the maximum period of continuous processing that can be observed for a given task. In terms of the admittance test in section 6.5, a task set is schedulable if it is schedulable under ideal LSA with reduced capacity $C - \Delta E_{max}$.

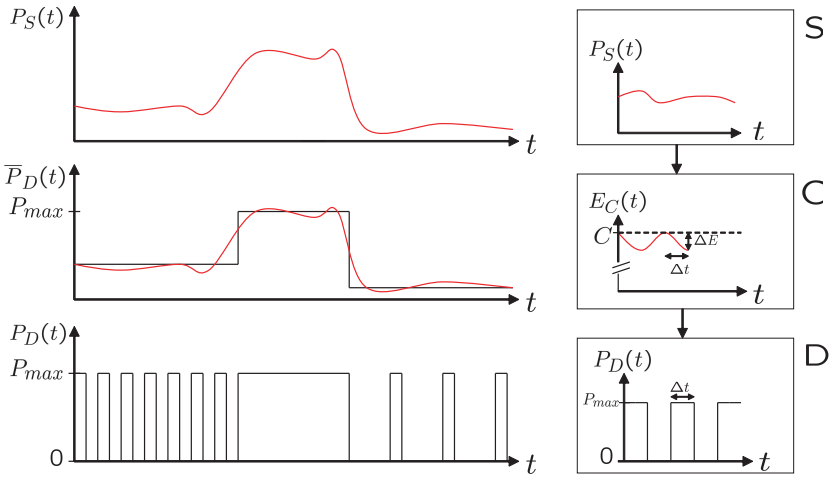


Figure 6.10: Approximated power consumption $\overline{P_D} \approx P_S$ by means of duty cycling and resulting non-ideal storage level $E_C(t) = C - \Delta E$

In the light of the considerations regarding assumption 1, also assumption 2 becomes plausible. Unlike common power management solution like Dynamic Voltage Scaling DVS, the energy e_i consumed by a task is the same regardless of the power P_D used during its execution. Although attractive due to its potential to *save* energy, DVS will come to its end if feature sizes of ICs get smaller. Hence, DVS or similar techniques were not considered in our work. In contrast, time and energy are assumed to be directly proportional – as indicated in Fig. 6.10 – with power $\overline{P_D}$ as constant of proportionality. The greater the number and size of time slots allocated to a given task, the higher the average power $\overline{P_D}$ and the faster the execution.

6.7.3 Energy Storage Model

An important step for the validation of the theory presented in this thesis is the discussion of the energy storage model. Looking at the various devices available on the market, there are two principal methods to store energy in a small volume or mass device: using an electro-chemical process or just performing physical separation of electrical charges across a dielectric medium. The first technique is used by rechargeable batteries and it is currently the most common and for long time it was the only method to achieve high capacities in a small size. Nevertheless, research in the last years has found new materials in order to increase the specific energy of capacitors, producing devices that are called supercapacitor or ultra-capacitor [93]. As shown in the so-called 'Ragone plot' in Fig. 6.11, supercapacitors offer a trade-off between power- as well as energy-density, filling the gap between batteries and capacitors. Beyond their ability to support higher power flows than batteries, supercapacitors overcome many other drawbacks of batteries: They have very long lifetimes and tolerate an almost unlimited number of charge/recharge cycles without performance degradation. Unlike batteries, no heat is released during charging/discharging due to parasitic, chemical reactions. In the following, we will focus on supercapacitors as possible candidates for an energy storage device.

Charge retention

Self-discharging is a natural phenomena that occurs in all kind of storage devices. It is caused by leakage currents that flow inside the device discharging it. Supercapacitors exhibit leakage currents that are typically in the order of magnitude of μA (see e.g. [94]). Moreover, it should be mentioned that the leakage is proportional to the energy level. In [59], the leakage behaviour of different supercapacitors have been tested. From Fig.2 in the latter work it becomes evident, that there exist a potential to minimize the leakage of fully

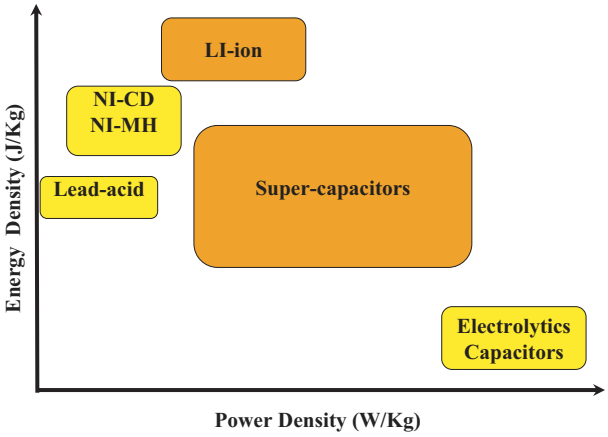


Figure 6.11: Power and energy characteristics of storage devices

charged supercapacitors by appropriate choice (manufacturer technology) and arrangement (serial, parallel) of devices.

Apart from the intrinsic energy leakage of supercapacitors, also the idle power consumption of the sensor node has to be considered. This "external leakage" can be reduced by switching to a low-power mode if no tasks are executed. In case of a low power wireless sensor node like Moteiv's Tmote Sky [95], its ultra low power Texas Instruments MSP430 F1611 microcontroller exhibits a maximum current of $3.0\mu A$ in low power mode (LPM3). The wakeup to active mode is finished after $6\mu s$.

Altogether, it can be assumed that the energy conservation laws described in Section 6.3.2 hold and introducing an additional term allowing for energy leakage is dispensable.

Monitoring the stored energy

An important feature of energy harvesting systems is the capability to estimate the remaining energy in the storage device. Unlike batteries, the energy of supercapacitors can be measured in a straightforward way: The equations describing the physical behaviour of supercapacitors are nearly the same as the ones for ordinary capacitors, and the energy stored is hence $E_C \approx \frac{1}{2}CV^2$.

Storage efficiency

The efficiency η of a supercapacitor can be regarded as the quantity that relates the power flows and energies displayed in Fig. 6.11. Since charging a supercapacitor with P_S and discharging it with P_D can be seen as symmetrical operations, let us consider the efficiency η when the supercapacitor is charged.

In this case, the increment of stored energy in time interval $\Delta = t_2 - t_1$ can be written as

$$E_C(t_2) - E_C(t_1) = \int_{t_1}^{t_2} P_S(t)\mathrm{d}t = \eta \int_{t_1}^{t_2} P_{S,raw}(t)\mathrm{d}t\,,$$

where $P_{S,raw}$ denotes the power fed into the supercapacitor. In general, supercapacitors may suffer from low efficiencies η due to their high equivalent series resistance [96]. This circumstance, however, does not jeopardize our modeling assumptions as such since, as stated in section 6.3.1, all losses are included in the definition of P_S. Actually, the more important property of the efficiency η is its independence of the energy level $E_C(t)$ at time t, which is approximately true for supercapacitors. Moreover, supercapacitors barely produce thermal heat which could reinforce non-linear charging/discharging behaviour.

In summary, we conclude that a linear charging/discharging behaviour with constant efficiency η is a reasonable abstraction for the example of a supercapacitor and hence our modeling assumptions hold. It should be mentioned, that possible variations of the efficiency η have been disregarded in related work like [59] and [97], too.

6.8 Related Work

In [97], the authors use a similar model of the power source as we do. But instead of executing concrete tasks in a real-time fashion, they propose tuning a node's duty cycle dependent on the parameters of the power source. Nodes switch between active and sleep mode and try to achieve sustainable operation. This approach only indirectly addresses real-time responsiveness: It determines the latency resulting from the sleep duration.

The approach in [98] is restricted to a very special offline scheduling problem: Periodic tasks with certain rewards are scheduled within their deadlines according to a given energy budget. The overall goal is to maximize the sum of rewards. Therefore, energy savings are achieved using Dynamic Voltage Scaling (DVS). The energy source is assumed to be solar and comprises two simple states: day and night. Hence the authors conclude that the capacity of the battery must be at least equal to the cumulated energy of those tasks, that have to be executed at night. In contrast, our work deals with a much more detailed model of the energy source. We focus on scheduling decisions for the online case when the scheduler is *indeed* energy-constraint. In doing so, we derive valuable bounds on the necessary battery size for arbitrary energy sources and task sets.

The research presented in [99] is dedicated to offline algorithms for schedul-

ing a set of periodic tasks with a common deadline. Within this so-called "frames", the order of task execution is *not* crucial for whether the task set is schedulable or not. The power scavenged by the energy source is assumed to be constant. Again – by using DVS – the energy consumption is minimized while still respecting deadlines. Contrary to this work, our systems (e.g. sensor nodes) are predominantly energy constrained and the energy demand of the tasks is fixed (no DVS). We propose algorithms that make best use of the available energy. Provided that the average harvested power is sufficient for continuous operation, our algorithms minimize the necessary battery capacity.

The primary commonality of [100] and our work is the term "lazy scheduling". In [100], lazy packet scheduling algorithms for transmitting packetized information in a wireless network are discussed. The approach is based on the observation that many channel coding schemes allow to reduce the energy per packet if it is transmitted slower, i.e. over a longer duration. Goal of this approach is to minimize the energy, whereas in our work the energy consumption is fixed (given by the task set). Furthermore, their scheduling algorithms are fully work conserving, which is not true for our algorithms.

6.9 Conclusions

We studied the case of an energy harvesting sensor node that has to schedule a set of tasks with real-time constraints. The arrival times, energy demands and deadlines of the tasks are not known to the node in advance and the problem consists of assigning the right amount of power in the right order to those tasks. For this purpose, we constructed optimal Lazy Scheduling Algorithms LSA which are energy-clairvoyant, i.e., the generated energy in the future is known. Contrary to greedy scheduling algorithms, LSA hesitates to power tasks until it is necessary to respect timing constraints. As a further result, we discuss an admittance test that decides, whether a set of energy-driven tasks can be scheduled on a sensor node without violating deadlines. This admittance test simultaneously shads light on the fundamental question of how to dimension the capacity of the energy storage: Provided that the average harvested power is sufficient for continuous operation, we are able to determine the minimum battery capacity necessary. Furthermore, achievable capacity savings between 20% and 45% are demonstrated in a simulative study, comparing the classical Earliest Deadline First algorithm with a variant of LSA, which uses energy variability characterization curves EVCC as energy predictor. Finally, practical considerations are provided that suggest practical applicability of the theoretical results.

By starting to study a single node, we believe that extensions towards mul-

tihop networks where end-to-end deadlines have to be respected are possible. If sensors jointly perform a common sensing task, distributed energy management solutions are needed. An example for the common task could be redundantly deployed sensors with overlapping coverage regions.

Chapter 7

Adaptive control for utility maximization

Recently, there has been a substantial interest in the design of systems that receive their energy from regenerative sources such as solar cells. In contrast to approaches that attempt to minimize the power consumption we are concerned with adapting parameters of the application such that a maximal utility is obtained while respecting the limited and time-varying amount of available energy. Instead of solving the optimization problem on-line which may be prohibitively complex in terms of running time and energy consumption, we propose a parameterized specification and the computation of a corresponding optimal on-line controller. The efficiency of the new approach is demonstrated by experimental results and measurements on a sensor node.

7.1 Introduction

The idea of harvesting energy to power electronic devices is not new, but it has gained a lot of attention recently. Especially in the domain of wireless sensor networks, techniques to transform environmental energy into usable electrical power have been in the focus of the research community. As a matter of fact, the benefits of solar or vibrational energy harvesters for sensor nodes are striking: One of the main advantages of sensor networks connected via wireless technology is their ability to operate in the absence of a pre-established infrastructure. This independence, however, severely depends on the autonomy of the single nodes. It has been shown in several feasibility studies [101, 102], that sensor nodes scavenging ambient energy may operate perpetually, without the need of manually replacing or recharging their batteries, see also the most recent results in [103].

In contrast to systems with a finite energy reservoir, systems powered by a regenerative energy source have to adapt to the stochastic nature of the energy available from the environment. The goal of such an adaptation is to maximize the utility of the application. For example, it has been shown that distributed applications such as clustering [87] and routing [86] in sensor networks can benefit if the underlying algorithms take advantage of spatial variations of the scavenged energy. By allocating different loads to the nodes of the network, the overall performance can be optimized.

Concerning the temporal variations of the environmental source at a single device, in [90] an optimal on-line algorithm to schedule a set of tasks within their deadlines is constructed. Taking into account available time as well as processable energy, an optimal task ordering is determined based on the prediction of the available energy in the future. In contrast to the work presented in this chapter, the average power consumption is not affected by the approach in [90] and parameters of the application are not changed.

Early work addressing the long-term performance of a device has been presented in [97]. Here, the average power consumption of a sensor node is adjusted using different duty cycles, i.e. by switching the sensor node on and off. But the method described is not adaptive since a fixed duty cycle is applied after having learned the characteristics of the energy source during a pre-operational phase.

In [98], a computing device may control its performance level by running different versions of a software application, each version having a different reward, time as well as energy demand. The energy source assumed is solar and comprises two simple states: day and night. The system considered assumes a minimum of harvested energy during daytime. In contrast to the work mentioned before, adaptive power management in [98] reduces to utilize surplus energy *after* having harvested it. To this end, several heuristics are presented.

The first work that can be classified as *adaptive* power management for energy harvesting systems has been published recently in [104, 105]. Here, the problem of tuning a sensor node's duty cycle has been formulated as a linear program (LP). This LP has to be solved periodically since the energy generated by a solar cell can only be estimated. Based on the assumption that the system is (a) capable of using the solar energy directly and (b) storing energy in a storage device with low efficiency, the objective in [104, 105] is to maximize the utilization of solar energy. The performance objective is to maximize the *average* duty cycle. For this objective, the authors propose a heuristic algorithm which attempts to decrease the duty cycle when the scavenged energy is low (e.g. at night) and increase the duty cycle when scavenged energy is high (e.g. during the day).

In contrast to the latter work, the class of linear programs presented in this chapter is capable of modeling a much larger variety of application scenarios, constraints and optimization objectives. For example, tradeoffs between the use of local memory and communication can be exploited. Finally, we are able to handle arbitrary, non-trivial objectives such as maximizing the *minimum* duty cycle. In addition, we propose to compute the optimal solution of a linear program by a method which exhibits low computational complexity and is thus well suited for resource constrained systems.

7.2 Contributions

This chapter points to the following new results:

- We present a specification model that is able to capture the performance, the parameters and the energy model of environmentally powered systems.
- As a main contribution, we suggest the use of simple, but optimal model predictive controllers to adapt parameters of an application. We are adapting results of the well-established framework of multiparametric programming to the emerging area of energy harvesting systems.
- We propose a practical technique for the efficient implementation of the controller and demonstrate the practical relevance of our approach by measurements of the controller running on a real sensor node.

7.3 System Concept

The system model is depicted in Fig. 7.1. The whole hardware/software system is powered by an energy harvesting device that delivers in a unit time interval starting at t the energy $E_S(t)$ which loads the energy storage. In the same time interval, the system uses energy $E_D(t)$. At time t, there is the stored energy $E_C(t)$ available.

Besides the application, there are two additional software tasks running on the target architecture. The estimator predicts future energy production of the harvesting device based on measurements of the past. The controller adapts properties of the application, e.g. task activation rates, based on the estimation of future available energy, the currently stored energy and additional information about the system state, e.g. the amount of available data memory. Parameters of the application are modified by the on-line controller. During execution, the system state (e.g. the amount of information stored in local memory and the stored energy) is changed.

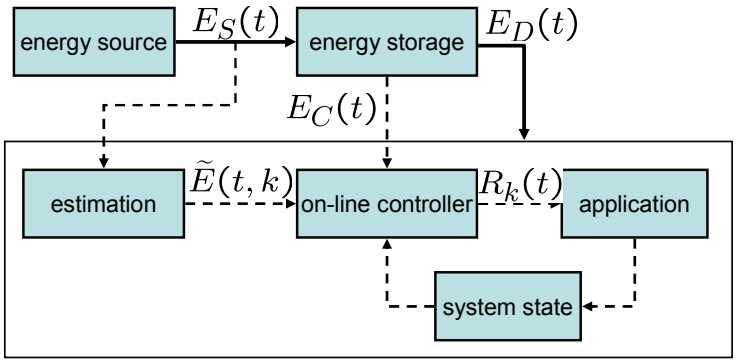


Figure 7.1: Illustration of the system concept.

It is the duty of the controller to adjust properties of the application such that overall objectives are optimized (for example maximizing the sampling rate of a sensor) while respecting system constraints (for example using not more than the available memory). The complexity of the controller design is caused by the fact that (a) the harvested energy changes in time, (b) the available future energy can only be estimated and (c) the overall objective is usually a long term goal, e.g. maximizing the minimal sampling rate of a sensor in the future. For example, a sensor node powered by an outdoor solar cell may save energy during the day in order to have enough energy available at night for transmitting sensor data. But it may also store the sensor data at night and send them during the day. As a result, complex controller strategies may be required.

Note that in contrast to other work in that area, energy $E_S(t)$ is always stored before used. For a direct usage of $E_S(t)$ with high efficieny bypassing the energy storage, our system concept (in particular the LP in 7.4.4) can be extended easily. In [104, 105], an example for such an extension is provided.

7.4 Basic Models and Methods

7.4.1 Power Model

The modeling is based on the notion of discrete time $t \in \mathbb{Z}_{\geq 0}$ where the difference in physical time between two discrete time instances is denoted as T. Energy related sensing and control may happen only at times t. In a practical setting, one may have a basic time interval T of a few minutes or even an hour.

The energy harvesting device is modeled as a power source which delivers energy $E_S(t)$ in the time interval $[t, t+1)$ of length T. Therefore, in time interval

$[t_1, t_2)$ with $t_1, t_2 \in \mathbb{Z}_{\geq 0}$ it delivers energy

$$E_S(t_1, t_2) = \sum_{t_1 \leq u < t_2} E_S(u)$$

The incoming power can be stored in a energy storage device, e.g. a rechargeable battery, a super-capacitor or a combination of both. The energy level at time t is denoted as $E_C(t)$.

The available energy can be used to execute tasks on various system components. Tasks may be as diverse as sensing, signal processing, A/D or D/A conversion, computing, or communicating. A task $\tau_i \in \mathbf{I}$ from the set of tasks $\mathbf{I}$ needs energy e_i for completing a single instance. The corresponding energy is drawn from the energy stroage. We suppose that a task is activated with a time-variant rate $s_i(t)$, i.e. during the basic time interval T starting at t, the task is executed $s_i(t)$ times. Therefore, a task needs energy

$$E_i(t_1, t_2) = \sum_{t_1 \leq u < t_2} e_i \cdot s_i(u)$$

in time interval $[t_1, t_2)$ for successful execution. The detailed application and task model will be described in Section 7.4.3.

7.4.2 Estimation

According to the system model described in Section 7.3, an important component of the adaptive power management is the predictor. It receives tuples $(t, E_S(t))$ for all times $t \geq 1$ and delivers N predictions on the energy production of the energy source. We assume that the prediction intervals are of equal size denoted as the number L (in units of the basic time interval T). Therefore, at time t, the predictor produces estimations $\widetilde{E}_S(t + k \cdot L, t + (k + 1) \cdot L)$ for all $0 \leq k < N$. We write $\widetilde{E}(t, k) = \widetilde{E}_S(t + k \cdot L, t + (k + 1) \cdot L)$ as a shorthand notation, i.e. the estimation of the incoming energy in the $(k + 1)$st prediction interval after t.

The prediction algorithm should depend on the type of the energy source and the system environment. Standard techniques known from automatic control and signal processing can be applied here. In this chapter, we provide a very simple and obvious technique only which is useful for solar cells that operate in an outdoor environment. It has been used for the experimental results presented in Section 7.5.

The estimation takes into account that the solar power can be considered to be periodic in $D = N \cdot L$, i.e. the length of a day (again in units of the basic time interval T), see also Fig. 7.2. Therefore, the prediction algorithm 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. We are using a simple exponential decay of old data with factor $0 < \alpha < 1$. In order to obtain predictions for the desired time intervals of length L we just add the predicted energy values for the corresponding intervals of length 1.

Finally, this long term prediction could be enhanced using a short term predictor that directly uses the information of the received power $P_S(t)$ for all t, again via an exponential decay with some factor β. This information could now used to improve the prediction for the next interval $\tilde{E}_S(t, t+L) = \tilde{E}(t, 0)$ by combining the long term and short term prediction using a weight γ. In the experiments, we did not make use of this possibility for simplicity reasons.

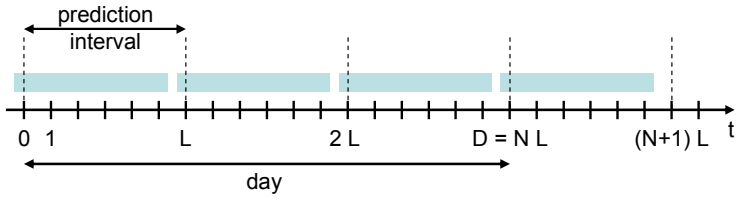


Figure 7.2: Illustration of prediction intervals.

7.4.3 Rate-Based Application Model

As described in Section 7.3, parameters of the application are changed during run-time in order to optimally use the available energy in the future. In this chapter, we restrict ourselves to a rate-based application model.

The application consists of tasks τ_i, $i \in \mathbf{I}$. A task is instantiated $s_i(t) \geq 0$ times in the interval of length T starting at time t and the execution of each instance needs energy e_i. The activation of tasks can be modeled by a rate graph whose nodes and edges represent tasks and activation relations, respectively. In particular, an edge from i to j denotes that task τ_i activates task τ_j. We may also say, that an edge (i, j) is activated r_{ij} times; such an activation is caused by the activation of task τ_i and leads to an activation of task τ_j.

A scaling factor σ_{ij} is associated to an edge (i, j) that represents how often task τ_j is activated for each activation of it. The default scaling is $\sigma_{ij} = 1$. If there are several edges leaving a node, then the sum of their rates equals the activation rate s of the source node, e.g. $s_1 = r_{12} + r_{13}$. This way, we can model a decision in the application, i.e. the execution of a task may either lead to the activation of one or another subsequent task. If there are several edges leaving from the same (graphical) location at some node then their activation rates are equal. This models the case that two subsequent tasks are activated with the same rate, i.e. $r_{12} = r_{13}$ if $(1, 2)$ and $(1, 3)$ have their tails at the same location.

The rate relations that are covered by a rate graph as defined above can be formulated as a set of linear (in)-equalities. Free variables $R_k(t)$ in this system are determined by the controller shown in Fig. 7.1, i.e. they are used to adapt the properties of the application in order to optimize its overall behavior while satisfying constraints, e.g. concerning energy and memory. In general, we can formulate the rate equations as

$$\mathbf{P} \cdot \mathbf{R}(t) + \mathbf{Q} = \mathbf{S}(t) \tag{7.1}$$

$$\mathbf{F} \cdot \mathbf{R}(t) + \mathbf{G} \geq \mathbf{0} \tag{7.2}$$

where $\mathbf{S}(t)$ is a vector containing all activation rates $s_i(t)$, $\mathbf{R}$ contains all controlled parameters $R_k(t)$ and $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{F}$ and $\mathbf{G}$ are vectors and matrices of appropriate dimensions.

The next Fig. 7.3 exemplifies the rate graph and gives an example for the associated rate equations.

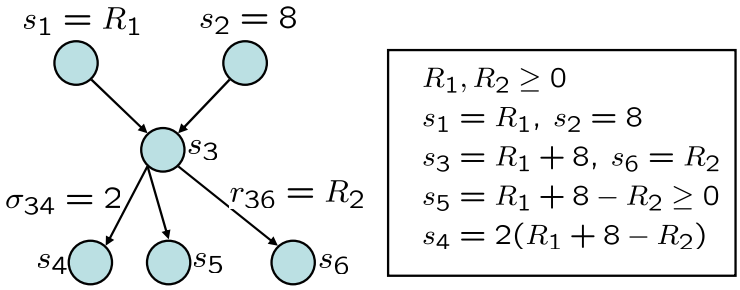


Figure 7.3: Illustration of rate graph and rate equations.

7.4.4 Linear Program Specification

The first step in constructing the on-line controller is the formulation of the optimization problem in form of a parameterized linear program (LP). The corresponding solution methods will be described in Section 7.4.5.

One of the essential states of the system is the stored energy. Using the power model described in Section 7.4.1 we obtain the following state equations:

$$\begin{aligned} E_C(t + k \cdot L) = E_C(t) - k \cdot \sigma + \\ + \sum_{j=0}^{k-1} (\gamma \cdot \widetilde{E}(t, j) - \epsilon \cdot L \cdot \mathbf{E} \cdot \mathbf{S}(t + j \cdot L)) \end{aligned} \tag{7.3}$$

They determine the expected contents of the energy storage at times $t + kL$ for $1 \leq k \leq N$. $\mathbf{S}(t)$ is a vector containing all activation rates $s_i(t)$ and $\mathbf{E}$ is a

(row)-vector that contains all energy requirements e_i for all $i \in \mathbf{I}$. Therefore, $\mathbf{E} \cdot \mathbf{S}(t) = \sum_{i \in \mathbf{I}} e_i \cdot s_i(t)$.

The factor σ accounts for energy leakage and γ, ϵ take into account a reduced energy efficiency. The equation also supposes that we have available a new estimation of the received energy only every L time instances and that the rate of the different tasks s_i is constant during each of these time intervals.

In a similar way, we can also model other system states, for example memory. A task could produce a certain amount of data that is stored and another removes it, e.g. by means of communication to another unit. In this case we would have for $1 \leq k \leq N$ the state equations:

$$M(t + k \cdot L) = M(t) + L \sum_{j=0}^{k-1} \mathbf{M} \cdot \mathbf{S}(t + j \cdot L) \tag{7.4}$$

$M(t)$ denotes the amount of stored data at time t and m_i is the amount of data produced or consumed by a task τ_i with rate s_i in a time interval of length T. $\mathbf{M}$ is a (row)-vector that contains all data amounts m_i for all $i \in \mathbf{I}$.

Of course, (7.3) and (7.4) provide only examples of possible system states and their associated changes. Moreover, there may be additional constraints on the set of feasible states.

One can now easily combine (7.1)-(7.4) and obtain a system of linear equalities and inequalities that contain as free variables $M(t + k \cdot L)$ (the state of the memory), $E_C(t + k \cdot L)$ (the state of the energy) for $1 \leq k \leq N$ and $\mathbf{R}(t + k \cdot L)$ (the rate control) for $0 \leq k < N$.

So far, no optimization goal has been formulated and therefore, any feasible rate control $\mathbf{R}(t + k \cdot L)$ could be a solution. Any linear objective function that makes use of the free variables given above is possible in this case. One may also define additional variables in order to model specific objectives. One possible (very simple) example would be the objective

$$\text{maximize } \lambda \text{ subject to } s_1(t + k \cdot L) \geq \lambda \quad \forall 0 \leq k < N \tag{7.5}$$

which would attempt to maximize the minimal rate with which the task τ_1 is operated in the finite horizon $0 \leq k < N$.

7.4.5 Controller Generation

Next, we will show how to design an on-line controller based on a state feedback law which avoids solving a linear program at each time step. Thereby, we are following the ideas in [106], where the regulation of discrete-time constrained linear systems is studied in the context of model predictive control (MPC).

As a first step, we define a state vector $\mathbf{X}$ consisting of the actual system state, the level of the energy storage as well as the estimation of the incoming energy over the finite prediction horizon (cp. Fig.7.1). Resuming the system dynamics formulated in (7.1)-(7.4), the state vector $\mathbf{X}$ can be written as

$$\mathbf{X}(t) = \left(E_C(t)\,,\, M(t)\,,\, \widetilde{E}(t,0)\,,\, \ldots,\, \widetilde{E}(t,N-1) \right)^T \tag{7.6}$$

Furthermore, let us denote the vector of planned control inputs to the system, i.e., the vector of future rates $\mathbf{R}$ as

$$\mathbf{U}(t) = \left(\mathbf{R}(t)\,,\, \mathbf{R}(t+L)\,,\, \ldots,\, \mathbf{R}\left(t+(N-1)\cdot L\right) \right)^T . \tag{7.7}$$

The state space of $\mathbf{X}$ (in our case $\mathbb{R}^{N+2}$ bounded by possible constraints on $E_C(t), M(t)$ and $\widetilde{E}(t,i)$) can now be subdivided into a number N_{CR} of polyhedrons. For each of these polyhedrons i (also called critical regions) the optimal solution $U_{opt,i}$ of the control problem can be made available explicitly as

$$\mathbf{U}_{opt,i}(t) = \mathbf{B}_i\mathbf{X}(t) + \mathbf{C}_i \quad \text{if } \mathbf{H}_i\mathbf{X}(t) \leq \mathbf{K}_i, i = 1, \ldots, N_{CR} \tag{7.8}$$

where $\mathbf{B}_i \in \mathbb{R}^{(N+2)\times N}, \mathbf{C}_i \in \mathbb{R}^N$ and $\mathbf{H}_i\mathbf{X}(t) \leq \mathbf{K}_i, i = 1 \ldots N_{CR}$ is a polyhedral partition of the state space of $\mathbf{X}(t)$. The computation of the vectors and matrices of control law (7.8) is done off-line using, e.g., the algorithm presented in [107] or other efficient solvers cited in the latter work.

In the on-line case, the controller has to identify to which region i the current state vector $\mathbf{X}(t)$ belongs. After this simple membership test, the optimal control moves $\mathbf{U}_{opt}(t)$ for the next N prediction intervals are computed by evaluating a linear function of $\mathbf{X}$. However, only the first entry of $\mathbf{U}_{opt}(t)$ is actually used to drive the application and we obtain the optimal rates

$$\mathbf{R}_{opt}(t) = (\,\mathbf{I}_N,\, 0,\, \ldots, 0\,) \cdot \mathbf{U}_{opt}(t) \tag{7.9}$$

where $\mathbf{I}_N$ denotes the $N \times N$ identity matrix.

In summary, the main idea of the proposed technique is to treat the linear program (LP) formulated in the last section as a multiparametric linear programm (mp-LP) with parameters $\mathbf{X}$ and optimization variables $\mathbf{U}$. The controller computes the optimal control inputs $\mathbf{U}_{opt}(t)$ as an explicit function of the current state $\mathbf{X}(t)$. The resulting control profile can be seen as a piecewise affine function of the state vector $\mathbf{X}(t)$.

It should be mentioned that evaluating the solution of a mp-LP results in the same control vectors $\mathbf{R}_{opt}(t)$ as would be obtained by iteratively solving the LP. However, the computational demand is greatly reduced compared to

solving a LP on-line. After having solved the mp-LP in advance, a set of N_{CR} polyhedra with associated control laws has to be stored and evaluated at each time step t. If the number of critical regions N_{CR} gets large, the computational effort still may be large as many tests of the form $\mathbf{HX}(t) \leq \mathbf{K}$ must be performed in order to determine the correct law $\mathbf{U}_i(t) = \mathbf{BX}(t) + \mathbf{C}$. Efficient methods to solve this potential bottleneck are described in Section 7.6.

7.5 Experimental Results

In this section, both feasibility and practical relevance of the proposed approach is demonstrated by means of simulation and measurements. For this purpose, we implemented the computation of on-line controllers for two exemplary case studies using the MATLAB toolbox in [108]. We simulated the behavior of the controlled system and measured the resulting controller overhead on a sensor node.

Measurements of solar light intensity $[\frac{W}{m^2}]$ during 28 consecutive days recorded at [109] serve as energy input $E_S(t)$. The time interval between two samples is 10 minutes, so we set the basic time interval $T = 10\,\text{min}$. Of course, one would have to scale the measured power profile in [109] with the size, number and efficiency of the actually used solar panels. However, we expect the influence of this scaling on our qualitative results to be negligible. In the prediction algorithm described in Section 7.4.2.

7.5.1 Adaptation of Sensing Rate

In this example, a sensor node is expected to measure some physical quantity like e.g. ambient temperature or mechanical vibrations and has to transmit the sampled data to a base station. We can model these requirements as a single sensing task τ_1 with rate $R_1(t)$, i.e. a task which is instantiated R_1-times in the interval $[t, t+T)$. For the sake of simplicity, the sensing task τ_1 drains at every instantiation 1 energy unit from the battery. Assume further that we want the maximum interval between two consecutive reports to be as small as possible, as in (7.5). For this setup, we can formulate the linear program LP in (13). Note that the last inequality in (13) is used to stabilize the receding horizon controller.

$$\text{maximize } \lambda \text{ subject to:} \qquad (13)$$

$$s_1(t + k \cdot L) \geq \lambda \qquad \forall 0 \leq k < N$$

$$E_C(t + k \cdot L) = E_C(t) +$$

$$+ \sum_{j=0}^{k-1} \left(\widetilde{E}(t, j) - L \cdot s_1(t + j \cdot L) \right) \geq 0 \quad \forall 1 \leq k \leq N$$

$$E_C(t + N \cdot L) \geq E_C(t) - 100$$

In general, the number of partitions N_{CR} of a multiparametric solution grows with the size of the state vector $\mathbf{X}$. Hence, it is of practical concern to keep the number of prediction intervals $\widetilde{E}(t, i)$ and therewith the dimension of $\mathbf{X}$ as small as possible. We chose $L = 24$ and $N = 6$ and obtain the states $\mathbf{X}(t) = \left(E_C(t), \widetilde{E}(t, 0), \ldots, \widetilde{E}(t, 5) \right)^T$. The resulting on-line controller consists of $N_{CR} = 7$ partitions.

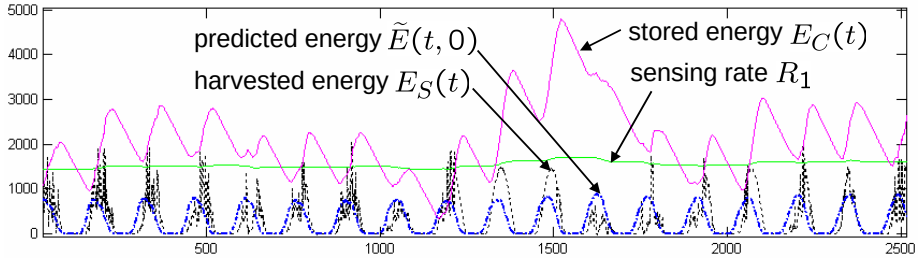


Figure 7.4: Adaptation of the sensing rate R_1.

Figure 7.4 shows the simulated sensing rate R_1 over a time period of 17 days for the generated optimizing controller. We started the simulation with an energy level $E_C(0) = 1300$ and found a nearly constant rate R_1 during the whole simulation time of 28 days. On the other hand, the stored energy $E_C(t)$ is highly varying, since the controller successfully compensates the unstable power supply $E_S(t)$. As a consequence, the stored energy $E_C(t)$ is increasing during the day and decreasing at night. Even the 8th, 12th and 13th displayed day with significant less sunshine are not jeopardizing the sensing rate R_1. Figure 7.4 clearly demonstrates that the simple on-line controller manages to meet the optimization goal.

7.5.2 Local Memory Optimization

Another simple scenario of practical importance may consist of two tasks running on a sensor node: A first task τ_1 is sampling some physical quantity, i.e., performing an A/D-conversion and storing the data in some local memory. A second task τ_2 is transmitting stored samples with a rate R_2 and thereby frees memory from the storage device. Clearly, the scaling factor $\sigma_{12} = R_1/R_2$ represents the ratio with which the amount of stored data M is increasing ($\sigma_{12} < 1$) or decreasing ($\sigma_{12} > 1$).

For this application, two reasonable optimization objectives would be (a) to minimize the unobserved intervals between any two consecutive samples and (b) to minimize the amount of stored data M. The purpose of the second objective is twofold: On one hand, sensor nodes are usually small, inexpensive low power devices with constrained hardware resources such as memory. On the other hand, the objective may to some extent enforce the freshness of data arriving at the base station.

$$\text{maximize } (\lambda - \mu) \text{ subject to:} \tag{14}$$

$$\begin{aligned}
& s_1(t + k \cdot L) \geq \lambda && \forall 0 \leq k < N \\
& M(t + N) \leq \mu \\
& E_C(t + k \cdot L) = E_C(t) + \textstyle\sum_{j=0}^{k-1} \widetilde{E}(t, j) - \\
& - \textstyle\sum_{j=0}^{k-1} \left(L \cdot [0.1 \;\; 0.9] \cdot \mathbf{S}(t + j \cdot L)\right) \geq 0 && \forall 1 \leq k \leq N \\
& M(t + k \cdot L) = M(t) + \\
& + L \textstyle\sum_{j=0}^{k-1} [1 \quad -1] \cdot \mathbf{S}(t + j \cdot L) \geq 0 && \forall 1 \leq k \leq N \\
& E_C(t + N \cdot L) \geq E_C(t) - 150
\end{aligned}$$

In general, radio communication is the main energy consumer on a sensor node. Hence we set the energies $e_1 = 0.1$ and $e_2 = 0.9$. The corresponding linear program LP is given by (14). To account for the additional system state $M(t)$ we reduced the number of prediction intervals and set $L = 36$ and $N = 4$. The computed control law divides the state space of $\mathbf{X}(t) = \left(E_C(t), M(t), \widetilde{E}(t, 0), \ldots, \widetilde{E}(t, 3)\right)^T$ in $N_{CR} = 39$ critical regions.

Fig. 7.5 displays the simulated curves of the state and control variables dur-

ing 6 days. The figure at the top represents a scenario with no use of local memory, i.e. the control problem in (7.10) whereas the bottom figures shows a comparable scenario using local memory according to (7.10). Until $t = 1700$, both tasks are adjusted to the same rate $R_1 = R_2$ and the memory $M(t)$ is empty. However, after two days with little harvested energy, the energy level $E_C(t)$ on the sensor node is falling and the controller starts to decrease the energy-costly communication task τ_2 by reducing the rate R_2. Consequently, the number of buffered samples M is increasing starting before $t = 1800$. In the following, the controller achieves to autonomously regulate the tradeoff between $E_C(t)$ and $M(t)$. While a lack of $E_C(t)$ would cause the non-initiation of task τ_1, an increasing $M(t)$ directly affects the second optimization objective. After $t = 1950$, the controller autonomously reduces the amount of occupied local memory by increasing the rate R_2.

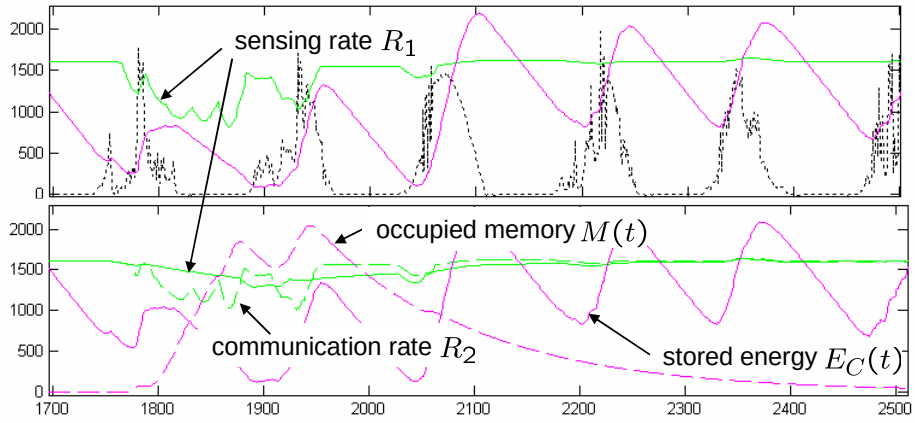


Figure 7.5: Top: scenario without local memory. Bottom: scenario with optimized local memory.

7.6 Efficient Implementation of the Control Law

In general, the identification of the active region i dominates the linear function evaluation of a control law (7.8) in terms of time and energy consumption. In the worst case, for all N_{CR} regions a matrix multiplication has to be performed in order to identify the active region i at time t. However, the identification of the active region i can be simplified due to the following facts: First, the matrices $\mathbf{H_i}$ are sparse which reduces the number of necessary multiplications and additions significantly. Second, usually only a subset of all N_{CR} is activated in practice and some of the regions in this subset are activated more frequently than others.

The second observation can be exploited by starting the search always with the region with the highest statistical occurrence, continue with the second highest, ..., and so on. To this end, an algorithm should maintain a list of regions i ordered by their frequencies which is updated every time step t.

For the memory optimization problem in Section 7.5.2, we found that on average $\approx 40\%$ of the entries of a matrix $\mathbf{H}$ are different from zero. Moreover, only 7 of 39 regions were used during the whole simulation period. Using probabilistic region testing as described above, the critical term of the form $\mathbf{H} \cdot \mathbf{X} \leq \mathbf{K}$ is only evaluated ≈ 1.45 times at time t, taking the average over all 4032 time steps.

Figure 7.6 displays the voltage measured at a 10Ω shunt resistor in series with a BTnode [110]. At first, the current energy level $E_C(t)$ of the battery and the scavenged energy $E_S(t)$ are determined via two A/D-conversions, which mark the two major peaks in plot 7.6. Focussing on the overhead of the proposed controller, we omit predicting the future energies $\widetilde{E}(t,i)$. Instead, an average situation is displayed where the region with the second highest frequency is the active one. Subsequently, the optimal control output for this region is calculated. It becomes evident, that the computations leading to the actual control actions take as long as the two A/D-conversions ($\approx 2ms$), having at the same time a significant lower power consumption. Hence, these measurements demonstrate how the simple, but efficient implementation of the proposed controller is applicable to sensor nodes, involving only marginal computation overhead.

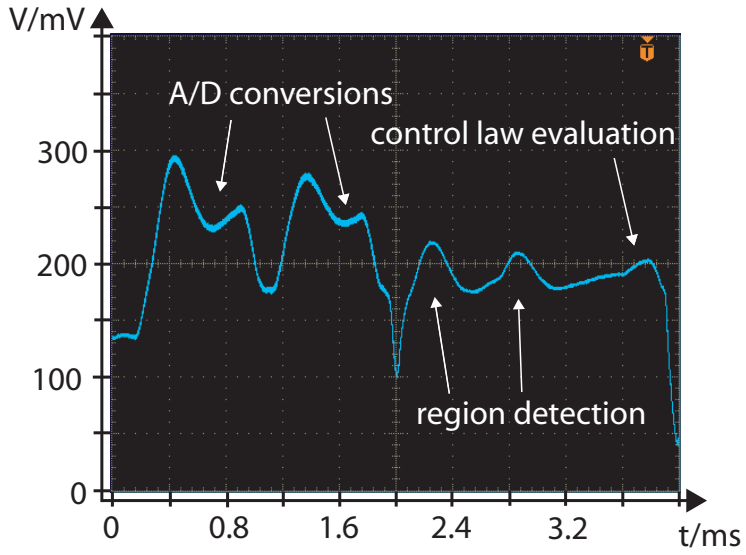


Figure 7.6: Computation of the optimal rates R_1, R_2 for the memory optimization problem on a BTnode.

7.7 Concluding Remarks

In this chapter we presented an approach to optimize the performance of energy harvesting systems subject to temporal variations of the energy source. For a rate-based application model, we propose to solve a linear program in a multiparametric fashion and shift most of the associated overhead to an offline computation. To this end, an energy prediction algorithm for outdoor solar cells is provided which delivers suitable estimations. The solution of the mp-LP is optimal with respect to the energy prediction but involves significantly lower overhead than solving the respective LP on-line. Measurements of the controller running on a real sensor node show how the method can be implemented efficiently.

Conclusions

Ambient Intelligence promotes pervasive and distributed technologies, that are not intrusive and always present. Wireless Sensor Network (WSN) is certainly the most important of these technologies and allows an environment (such as a room, a building, a park) to be user-interactive and to be aware of the intentions of the users.

While advances in IC fabrication technologies, circuit designs, and networking techniques have greatly reduced the cost, size, and power consumption of potential wireless sensor platforms, the development of suitable power sources for many applications lags and do not follow the Moore's Law. Thus, the amount of available on-board energy still limits heavily the operating lifetime of wireless sensor nodes. Alternative power sources must be employed and devices that extract energy from their surroundings in some way (the so called energy scavenging devices) have recently attracted attention from many researchers.

In this thesis we have discussed our results about the hardware/software design of systems that receive their energy from regenerative sources such as solar cells. We started with the design of a testbed node, suitable also for wearable computing applications and we have continued with the analysis on energy harvesting techniques to achieve a perpetual node supplying. We then focused on the problem of autonomy and energy management, with systems which extract energy form the surroundings exploiting miniaturized PV modules. Since devices are powered by a fluctuating energy source, common power management policies have to be reconceived. Based on this motivation, we finally developed algorithms that jointly handle constraints from both energy and time domain. Solutions have been proposed both for sensor nodes equipped with Operating System and for OS-free sensor nodes.

In conclusion, Ambient Intelligence will have a major impact on software and embedded systems design, it will introduce many new media applications and new user interface concepts, bringing innovations in several fields of human activity. In this thesis we have contributed tackling some of the numerous open research challenges in the sensor networks domain.

Bibliography

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 66–75, Sept. 1991.

[2] Mark Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999.

[3] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, New York, NY, USA, 2002, pp. 88–97, ACM Press.

[4] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," 2001.

[5] Ember coorporation web site, "http://www.ember.com," .

[6] Moteiv company web site, "http://www.moteiv.com," .

[7] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research.," in *4th International Conference on Information Processing in Sensor Networks (IPSN05)*, Piscataway, NJ, April 2005, pp. pp. 364–369.

[8] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for smart dust," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, New York, NY, USA, 1999, pp. 271–278, ACM Press.

[9] J.M. Rabaey, M.J. Ammer, Jr. da Silva, J.L., D. Patel, and S. Roundy, "Picoradio supports ad hoc ultra-low power wireless networking," *Computer*, vol. 33, no. 7, pp. 42–48, July 2000.

[10] The Official Bluetooth Membership Site, "http://www.bluetooth.org," .

[11] Zigbee Alliance site, "http://www.zigbee.org," .

[12] Proceedings of the IEEE, Ed., *Sensor networks: evolution, opportunities, and challenges*, vol. 91, August 2003.

[13] CrossBow solutions web site, "http://www.xbow.com," .

[14] The MillennialNet coorporation web site , "http://www.millennialnet.com," .

[15] The Dust Networks web site, "http://www.dustnetworks.com," .

[16] iBadge project website, "http://nesl.ee.ucla.edu/projects/ibadge/default.htm," .

[17] Sung Park, I. Locher, A. Savvides, M.B. Srivastava, A. Chen, R. Muntz, and S. Yuen, "Design of a wearable sensor badge for smart kindergarten," in *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on*, 7-10 Oct. 2002, pp. 231–238.

[18] Intel motes, "http://www.intel.com/research/exploratory/motes.htm," .

[19] R. Barbieri, E. Farella, L. Benini, B. Ricco, and A. Acquaviva, "A low-power motion capture system with integrated accelerometers [gesture recognition applications]," in *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE*, 5-8 Jan. 2004, pp. 418–423.

[20] T. Harada, H. Uchino, T. Mori, and T. Sato, "Portable orientation estimation device based on accelerometers, magnetometers and gyroscope sensors for sensor network," in *Multisensor Fusion and Integration for Intelligent Systems, MFI2003. Proceedings of IEEE International Conference on*, 30 July-1 Aug. 2003, pp. 191–196.

[21] E.R. Bachmann, Xiaoping Yun, D. McKinney, R.B. McGhee, and M.J. Zyda, "Design and implementation of marg sensors for 3-dof orientation measurement of rigid bodies," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 14-19 Sept. 2003, vol. 1, pp. 1171–1178vol.1.

[22] TinyOS, "http://www.tinyos.net/," .

[23] E. Foxlin and M. Harrington, "Weartrack: a self-referenced head and hand tracker for wearable computers and portable vr," in *Wearable Computers, 2000. The Fourth International Symposium on*, 16-17 Oct. 2000, pp. 155–162.

[24] J.K. Perng, B. Fisher, S. Hollar, and K.S.J. Pister, "Acceleration sensing glove (asg)," in *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, 18-19 Oct. 1999, pp. 178–180.

[25] A.D. Cheok, K. Ganesh Kumar, and S. Prince, "Micro-accelerometer based hardware interfaces for wearable computer mixed reality applications," in *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on*, 7-10 Oct. 2002, pp. 223–230.

[26] Nicky Kern and Bernt Schiele, "Multi-sensor activity context detection for wearable computing," November 2003.

[27] C. Randell and H. Muller, "Context awareness by analysing accelerometer data," in *Wearable Computers, 2000. The Fourth International Symposium on*, 16-17 Oct. 2000, pp. 175–176.

[28] Kamiar Aminian, Eduardo De Andres, Karen Rezakhanlou, Carlo Fritsch, Y. Schutz, Michèle Depairon, Pierre-François Leyvraz, and Philippe Robert, "Motion analysis in clinical practice using ambulatory accelerometry," in *CAPTECH '98: Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, London, UK, 1998, pp. 1–11, Springer-Verlag.

[29] Andrew R. Golding and Neal Lesh, "Indoor navigation using a diverse set of cheap, wearable sensors," in *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, Washington, DC, USA, 1999, p. 29, IEEE Computer Society.

[30] Rong Zhu and Zhaoying Zhou, "A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Rehabilitation Engineering]*, vol. 12, no. 2, pp. 295–302, June 2004.

[31] H. Junker, P. Lukowicz, and G. Troster, "Padnet: wearable physical activity detection network," in *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, 18-21 Oct. 2005, pp. 244–245.

[32] R. Headon and G. Coulouris, "Supporting gestural input for users on the move," in *Eurowearable, 2003. IEE*, 4-5 Sept. 2003, pp. 107–112.

[33] Jonny Farringdon, Andrew J. Moore, Nancy Tilbury, James Church, and Pieter D. Biemond, "Wearable sensor badge and sensor jacket for context awareness," in *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, Washington, DC, USA, 1999, p. 107, IEEE Computer Society.

[34] K. Van Laerhoven, A. Schmidt, and H.-W. Gellersen, "Multi-sensor context aware clothing," in *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on*, 7-10 Oct. 2002, pp. 49–56.

[35] K. Van Laerhoven, N. Kern, H.W. Gellersen, and B. Schiele, "Towards a wearable inertial sensor network," in *Eurowearable, 2003. IEE*, 4-5 Sept. 2003, pp. 125–130.

[36] Atmel microprocessors, "http://www.atmel.com/," .

[37] ST MEMS accelerometers
, "http://www.st.com/stonline/books/ascii/docs/10175.htm," .

[38] RFM company web site, "http://www.rfm.com," .

[39] M. Caccamo, L.Y. Zhang, Lui Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*, 3-5 Dec. 2002, pp. 39–48.

[40] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Troster, "Wearable sensing to annotate meeting recordings," in *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on*, 7-10 Oct. 2002, pp. 186–193.

[41] P. Bonato, "Advances in wearable technology and applications in physical medicine and rehabilitation," *J. Neuroengineering. Rehabil.*, vol. 2, pp. 2, 2005.

[42] J. Price J. Chapman A. Moore E. Jovanov, D. Raskovic and A. Krishnamurthy, "Patient monitoring using personal area networks of wireless intelligent sensors," *Biomed. Sci. Instrum.*, vol. 37, pp. 373–378, 2001.

[43] S.J. Morris J.A Paradiso, "Shoe-integrated sensor system for wireless gait analysis and real-time feedback.," *Proceedings of the Second Joint EMBS/BMES Conf. USA*, pp. 2468–2469, Oct. 2002.

[44] M. W. Horstinkand S. C. Gielen N. L. Keijsers, "Movement parameters that distinguish between voluntary movements and levodopa-induced dyskinesia in parkinson's disease," *Hum. Mov Sci.*, vol. 22, pp. 67–89, 2003.

[45] T. F. Budinger, "Biomonitoring with wireless communications," *2003Annu. Rev. Biomed. Eng*, vol. 5, pp. 383–412, 2003.

[46] S. Jayaraman S. Park, "e-health and quality of life: the role of the wearable motherboard," *Stud. Health Technol. Inform.*, vol. 108, pp. 239–252, 2004.

[47] W. Frasson D. Cattaneo, M. Ferrarin and A. Casiraghi, "Head control: volitional aspects of rehabilitation training in patients with multiple sclerosis compared with healthy subjects," *Arch. Phys. Med. Rehabil.*, vol. 86, pp. 1381–1388, 2005.

[48] F. Mayer J. Weller H. C. Heitkamp, T. Horstmann and H. H. Dickhuth, "Gain in strength and muscular balance after balance training," *Int. J. Sports Med.*, vol. 22, pp. 285–290, 2001.

[49] M. Reschke G. Clement and S. Wood, "Neurovestibular and sensorimotor studies in space and earth benefits," *Curr. Pharm. Biotechnol.*, vol. 6, pp. 267–283, 2005.

[50] D. Hilton P. A. Grace G. M. Lyons, K. M. Culhane and D. Lyons, "A description of an accelerometer-based mobility monitoring technique," *Med. Eng Phys.*, vol. 27, pp. 497–504, 2005.

[51] B. Chan L. Rocchi F. B. Horak M. Dozza, L. Chiari and A. Cappello, "Influence of a portable audio-biofeedback device on structural properties of postural sway," *J. Neuroengineering. Rehabil.*, vol. 2, pp. 13, 2005.

[52] C. Otto E. Jovanov, A. Milenkovic and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *J. Neuroengineering. Rehabil*, vol. 2, pp. 6, 2005.

[53] Eugene Shih, Seong-Hwan Cho, Nathan Ickes, Rex Min, Amit Sinha, Alice Wang, and Anantha Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA, 2001, pp. 272–287, ACM Press.

[54] L. Benini E. Farella, A. Acquaviva and B. Ricco', "A wearable gesture recognition system for natural navigation interfaces," in *Proceedings of EUROMEDIA2005*, Toulouse, 11-13 April 2005.

[55] E. Deutsch, *Ear and BrainEar and Brain: How we Make Sense of Sounds*, Springer-Verlag, 2003.

[56] I. Stark, "Invited talk: Thermal energy harvesting with thermo life," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 3-5 April 2006, pp. 19–22.

[57] M. Stordeur and I. Stark, "Low power thermoelectric generator-self-sufficient energy supply for micro systems," in *Thermoelectrics, 1997.*

Proceedings ICT '97. XVI International Conference on, 26-29 Aug. 1997, pp. 575–577.

[58] G.L. Solbrekken, K. Yazawa, and A. Bar-Cohen, "Thermal management of portable electronic equipment using thermoelectric energy conversion," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM '04. The Ninth Intersociety Conference on*, 1-4 June 2004, pp. 276–283Vol.1.

[59] Xiaofan Jiang, Joseph Polastre, and David E. Culler, "Perpetual environmentally powered sensor networks.," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, UCLA, Los Angeles, California, USA, April 25-27 2005, pp. 463–468.

[60] Farhan Simjee and Pai H. Chou, "Everlast: long-life, supercapacitor-operated wireless sensor node," in *ISLPED '06: Proceedings of the 2006 international symposium on Low power electronics and design*, New York, NY, USA, 2006, pp. 197–202, ACM Press.

[61] Chulsung Park and P.H. Chou, "Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *Proceedings of the Sensor and Ad Hoc Communications and Networks. SECON '06.*, 2006, vol. 1, pp. 168–177.

[62] M. Veefkind, "Industrial design and pv-power, challenges and barrieres," in *proceedings of the ISES Solar World Congress*, 2003.

[63] P. Fairley, "Solar-cell rollout," Technology Review, July-August 2004, pp.35-40.

[64] S. Roundy, B. Otis, Y.H. Chee, J.M. Rabaey, and P. Wrigh, "A 1.9GHz RF transmit beacon using environmentally scavenged energy," in *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED 2003)*, Aug. 2003, design contest.

[65] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. P. Chandrakasan, and J. H. Lang, "Vibration-to-electric energy conversion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 1, pp. 64–76, 2001.

[66] F. Moll and A. Rubio, "An approach to the analysis of wearable body-powered systems," in *In Proceedings MIXDES 2000*, June 2000.

[67] Nathan S. Shenck and Joseph A. Paradiso, "Energy scavenging with shoe-mounted piezoelectrics," *IEEE Micro*, vol. 21, no. 3, pp. 30–42, 2001.

[68] Joseph A. Paradiso and Mark Feldmeier, "A compact, wireless, self-powered pushbutton controller," in *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, London, UK, 2001, pp. 299–304, Springer-Verlag.

[69] Inc. radio switches EnOcean and sensors without batteries, "http://www.enocean.com/indexd.html," .

[70] microminiature sensor Microstrain, "http://www.microstrain.com/," .

[71] Convert vibrations into electricity Ferro Solution, "http://www.ferrosi.com/energy-harvesters.html," .

[72] D. Fye J. Zahnd S. Priya, CT. Chen, "Piezoelectric windmill: A novel solution to remote sensing," *Japanese Journal of Applied Physics*, vol. 44, pp. L104–L107, 2005.

[73] N. Mano, F. Mao, and A. Heller, "A miniature biofuel cell operating in a physiological buffer," *Journal of the American Chemical Society*, vol. 124, no. 44, pp. 12962–12963, 2002.

[74] Tae-Yeop Kim, Ho-Gyun Ahn, Seung Kyu Park, and Youn-Kyun Lee, "A novel maximum power point tracking control for photovoltaic power system under rapidly changing solar radiation," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, 12-16 June 2001, vol. 2, pp. 1011–1014vol.2.

[75] K. Kobayashi, H. Matsuo, and Y. Sekine, "A novel optimum operating point tracker of the solar cell power supply system," in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, 20-25 June 2004, vol. 3, pp. 2147–2151Vol.3.

[76] Chihchiang Hua and Chihming Shen, "Study of maximum power tracking techniques and control of dc/dc converters for photovoltaic power system," in *Power Electronics Specialists Conference, 1998. PESC 98 Record. 29th Annual IEEE*, 17-22 May 1998, vol. 1, pp. 86–93vol.1.

[77] Dong-Yun Lee, Hyeong-Ju Noh, Dong-Seok Hyun, and Ick Choy, "An improved mppt converter using current compensation method for small scaled pv-applications," in *Applied Power Electronics Conference and Exposition, 2003. APEC '03. Eighteenth Annual IEEE*, 9-13 Feb. 2003, vol. 1, pp. 540–545vol.1.

[78] Vijay Raghunathan, Aman Kansal, Jason Hsu, Jonathan Friedman, and Mani B. Srivastava, "Design considerations for solar energy harvest-

ing wireless embedded systems.," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, UCLA, Los Angeles, California, USA, April 25-27 2005, pp. 457–462.

[79] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan, "Harvesting aware power management for sensor networks," in *Automation Conference, 2006 43rd ACM/IEEE*, 24-28 July 2006, pp. 651–656.

[80] Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini, "Real-time scheduling with regenerative energy.," in *18th Euromicro Conference on Real-Time Systems, ECRTS 2006*, Dresden, Germany, July 5-7 2006.

[81] Yeong-Chau Kuo, Tsorng-Juu Liang, and Jiann-Fuh Chen, "Novel maximum-power-point-tracking controller for photovoltaic energy conversion system," *Industrial Electronics, IEEE Transactions on*, vol. 48, no. 3, pp. 594–601, June 2001.

[82] E. Koutroulis, K. Kalaitzakis, and N.C. Voulgaris, "Development of a microcontroller-based, photovoltaic maximum power point tracking control system," *Power Electronics, IEEE Transactions on*, vol. 16, no. 1, pp. 46–54, Jan. 2001.

[83] Clare, Inc., "Clare solar cells - cpc series data sheet," http://www.clare.com/Products/SolarCell.htm, August, 2005.

[84] Shad Roundy, Dan Steingart, Luc Frechette, Paul K. Wright, and Jan M. Rabaey, "Power sources for wireless sensor networks.," in *Wireless Sensor Networks, First European Workshop, EWSN 2004, Proceedings*, Berlin, Germany, January 19-21 2004, Lecture Notes in Computer Science, pp. 1–17, Springer.

[85] Yasser Ammar, Aurélien Buhrig, Marcin Marzencki, Benoît Charlot, Skandar Basrour, Karine Matou, and Marc Renaudin, "Wireless sensor network node with asynchronous architecture and vibration harvesting micro power generator," in *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, New York, NY, USA, 2005, pp. 287–292, ACM Press.

[86] Longbi Lin, Ness B. Shroff, and R.Srikant, "Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources.," in *Proceedings of IEEE INFOCOM 2005*, Miami, USA, March 13-17 2005, pp. 1262 – 1272.

[87] Thiemo Voigt, Hartmut Ritter, Jochen Schiller, Adam Dunkels, and Juan Alonso, "Solar-aware clustering in wireless sensor networks," in *Pro-*

ceedings of the Ninth IEEE Symposium on Computers and Communications, June 2004.

[88] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha, and J. Hou, "Real-time communication and coordination in embedded sensor networks.," in *Proceedings of the IEEE*, vol.91, no.7pp. 1002- 1022, July 2003.

[89] Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini, "Lazy scheduling for energy-harvesting sensor nodes.," in *Fifth Working Conference on Distributed and Parallel Embedded Systems, DIPES 2006*, Braga, Portugal, October 11-13 2006.

[90] Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini, "Real-time scheduling with regenerative energy.," in *18th Euromicro Conference on Real-Time Systems, ECRTS 2006*, Dresden, Germany, July 5-7 2006.

[91] Sanjoy K. Baruah, "Dynamic- and static-priority scheduling of recurring real-time tasks," *Real-Time Systems*, vol. 24, no. 1, pp. 93–128, 2003.

[92] Ernesto Wandeler, Alexander Maxiaguine, and Lothar Thiele, "Quantitative characterization of event streams in analysis of hard real-time applications.," *Real-Time Systems, Springer Science+Business Media B.V.*, vol. 9, no. 2, pp. 205–225, 2005.

[93] R. Ktz and M. Carlen, "Principles and applications of electrochemical capacitors.," in *Electrochimica Acta 45*. 2000, pp. 2483–2498, Elsevier Science Ltd.

[94] Maxwell technologies, Inc., "Boostcap ultracapacitor - pc series data sheet," http://www.maxwell.com/pdf/uc/datasheets/PC_Series.pdf, June, 2006.

[95] Moteiv Corporation., "Tmote sky - ultra low power ieee 802.15.4 compliant wireless sensor module, datasheet," http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf, June, 2006.

[96] Philippe Barrade and Alfred Rufer, "Current capability and power density of supercapacitors: Considerations on energy efficiency.," in *European Conference on Power Electronics and Applications (EPE), 2003*, Toulouse, France, September 2-4 2003.

[97] Aman Kansal, Dunny Potter, and Mani B. Srivastava, "Performance aware tasking for environmentally powered sensor networks.," in *Proceedings of the International Conference on Measurements and Modeling of*

Computer Systems, SIGMETRICS 2004, New York, NY, USA, June 10-14 2004, pp. 223–234, ACM Press.

[98] Cosmin Rusu, Rami G. Melhem, and Daniel Mosse, "Multi-version scheduling in rechargeable energy-aware real-time systems.," in *15th Euromicro Conference on Real-Time Systems, ECRTS 2003*, Porto, Portugal, July 2-4 2003, pp. 95–104.

[99] Andre Allavena and Daniel Mosse, "Scheduling of frame-based embedded systems with rechargeable batteries.," in *Workshop on Power Management for Real-Time and Embedded Systems (in conjunction with RTAS 2001)*, 2001.

[100] B. Prabhakar, Elif Uysal-Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling.," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Anchorage, AK, USA, April 2001, pp. 386–394.

[101] Xiaofan Jiang, Joseph Polastre, and David E. Culler, "Perpetual environmentally powered sensor networks.," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, UCLA, Los Angeles, California, USA, April 25-27 2005, pp. 463–468.

[102] Vijay Raghunathan, Aman Kansal, Jason Hsu, Jonathan Friedman, and Mani B. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems.," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, UCLA, Los Angeles, California, USA, April 25-27 2005, pp. 457–462.

[103] Farhan Simjee and Pai H. Chou, "Everlast: Long-life, supercapacitor-operated wireless sensor node," in *Proc. International Symposium on Low Power Electronics and Design (ISLPED)*, Oct. 2006.

[104] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava, "Power management in energy harvesting sensor networks," in *ACM Transactions on Embedded Computing Systems (in revision)*, May 2006, also available from: NESL Technical Report Number: TR-UCLA-NESL-200605-01.

[105] Aman Kansal, Jason Hsu, Mani B. Srivastava, and Vijay Raghunathan, "Harvesting aware power management for sensor networks," in *43rd Design Automation Conference (DAC)*, San Francisco, USA, July 2006, also available from: NESL Technical Report Number: TR-UCLA-NESL-200605-02.

[106] Alberto Bemporad, Francesco Borrelli, and Manfred Morari, "Model Predictive Control Based on Linear Programming - The Explicit Solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, Dec. 2002.

[107] F. Borrelli, A. Bemporad, and M. Morari, "A Geometric Algorithm for Multi-Parametric Linear Programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 515–540, Sept. 2003.

[108] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004.

[109] Sunergy, "Recordings of solar light intensity from 08/08/2005 to 04/09/2005, datasheet," http://sunergy.dmclf.org/cms/, June, 2006.

[110] J. Beutel, M. Dyer, M. Hinz, L. Meier, and M. Ringwald, "Next-generation prototyping of sensor networks," in *Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004)*. Nov. 2004, pp. 291–292, ACM Press, New York.