6-21-2011

Reconfigurable Sampling for Enhanced Energy Efficiency in Power-Constrained Wireless Systems

Timothy Kurp
timothy.kurp@engr.uconn.edu

Reconfigurable Sampling for Enhanced Energy Efficiency in Power-Constrained Wireless Systems

Timothy Daniel Kurp

B.S., Seton Hall University 2009

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2011

APPROVAL PAGE

Master of Science Thesis

Reconfigurable Sampling for Enhanced Energy Efficiency in Power-Constrained Wireless Systems

Presented by

Timothy Daniel Kurp, B.S.

Major Advisor

_____

Robert X. Gao

Associate Advisor

_____

Jiong Tang

Associate Advisor

_____

Nejat Olgac

University of Connecticut

2011

ACKNOWLEDGEMENTS

I am grateful to Professor Robert Gao for his continued advisement, guidance and support throughout the past two years, which I feel have significantly advanced my capabilities as a scientific researcher. His passion for high level research and endeavor to constantly improve has been a source of inspiration and also a reinforcement of strong work ethic. I believe that the training and experience I have received under his guidance are truly invaluable assets to my future career.

I would like to thank Professor Olgac and Professor Tang for serving on my thesis committee and providing insight into my research work.

I sincerely appreciate the support, feedback and help provided by my all of my colleagues and lab mates in the Electromechanical Systems Laboratory, especially Sripati Sah, Shaopeng Liu and Jinjiang Wang. Their advice and eagerness to share expertise has been instrumental in the development of my research work.

I extend a special gratitude to my parents, brothers and sisters who have supported and encouraged me throughout my life and graduate studies. Their attitude, ethic and example have formed my character and are the source of my desire to succeed.

Lastly, I would like to sincerely acknowledge the funding provided to this research by the National Science Foundation under grant EFRI-0735974.

ABSTRACT

RECONFIGURABLE SAMPLING FOR ENHANCED ENERGY EFFICIENCY IN POWER-CONSTRAINED WIRELESS SYSTEMS

JUNE 2011

TIMOTHY DANIEL KURP

B.S. PHYSICS, SETON HALL UNIVERSITY, SOUTH ORANGE, NJ

M.S. M.E, UNIVERSITY OF CONNECTICUT, STORRS, CT

Directed by: Professor Robert X. Gao

Rapid advancement in electronics for wireless transmission and computing has fostered the development of low cost wireless sensors used in a variety of applications. As the wireless sensor 'nodes' become more computationally powerful, yet scaled downward in terms of size, energy consumption and cost, they are able to not only replace wired sensing systems but reveal sensing applications previously not feasible. Many individual wireless sensor nodes can be densely deployed in the form of a wireless sensor network (WSN) to effectively monitor large spatio-temporal regions, while supporting ad-hoc communication between sensor nodes.

Although wireless sensing has found home in applications such as environmental, structural health, surveillance and industrial process monitoring, there are certain limitations and risks associated with a wireless versus a wired implementation. The limiting factor is most often the node or network lifetime, which is directly constrained by the limited onboard energy supply; a battery. Therefore, there is tremendous interest

in decreasing the power dissipation of these devices in order to enable more implementations, and extend lifetime in existing ones. The sensor node can be thought of as three synergistically acting subsystems: the data acquisition, processing and communication subsystems, where research has focused on addressing energy bottlenecks in each subsystem. Of the many techniques, a data driven approach which focuses on minimizing data load is capable of significantly decreasing power by reducing communications in transmitting data. The focus of this work is adaptive sensing; a data driven technique which extends this concept to further reducing energy consumption in all subsystems, as well as aiding in data load and bandwidth issues. In realizing such a framework, four research tasks have been identified and executed throughout the course of this research.

1) Formulation of an Efficient and Implementable Adaptive Sensing Method

Traditionally, data is acquired from individual sensors by the node at fixed time intervals. This is wasteful in that the sampling rate must accommodate the 'worst case' scenario; always sampling at a high rate. At times of low signal activity, this induces redundancy in the data, leading to unnecessary expenditure of energy in acquiring, processing and transmitting these redundant points. An *event-driven* sensing scheme in the form of an adaptive sampling algorithm is formulated in this work that dynamically adjusts the sampling rate to reduce redundancy and improve energy efficiency.

2) Formulation of Signal Recovery Procedure

Through incorporation of a non-uniform sampling scheme, the desired information is preserved but may be in a format which is not immediately usable. Once the adaptively

sampled data is received by the base station or end device, it is essential that the original signal be reconstructed from the received points. Interpolative techniques are leveraged to derive a signal reconstruction for samples acquired while using the adaptive sampling algorithm.

3) Algorithm Performance Analysis and Error Quantification

Simulations are performed which model the adaptive sampling process using real signals from various applications. An energy model for the sensor node is incorporated to quantify the effect that the algorithm implementation has on a commercially available platform's energy consumption. Due to a problem which will be discussed, traditional error metrics cannot effectively quantify error in context of the sampling algorithm's performance. Therefore, a custom error metric has been formulated to fully quantify the effect that the algorithm implementation will have on signal quality.

4) Implementation and Performance Verification

In addition to simulating the adaptive sensing process, two successful implementations are demonstrated which are scalable for other real sensing systems. A 'proof of concept' has first been demonstrated on an 8-bit MCU interfaced with a Bluetooth transceiver. Also, a more advanced adaptive sampling system has been realized on a Digital Signal Processor DSP based system with Zigbee communications. Performance and energy consumption measurements have been performed and are presented in this work, as well as a full discussion of the embedded design, implementation challenges and solutions.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Background

Utilization of sensors to measure naturally occurring or man-made phenomena enables monitoring and decision making processes previously unthinkable through pure human observation. In fact, sensing has become a vital component in an extensive amount of applications such as structural health [1][2], machining and manufacturing [3] , surveillance [4] and medical [5]. In incorporating sensors into any application there are many challenges which must be overcome. The two primary concerns are always (1) how to power the sensor unit and (2) how to retrieve the data or communicate findings to the desired destination.

Traditionally, this is accomplished by wiring the sensor and its electronics to a power source as well as wiring into a communication network or directly to a computer. This has several disadvantages, namely cost of installation and maintenance due to cabling which can deters sensor integration altogether. Additionally, cables limit motion; in a manufacturing setting or in medical monitoring applications for example, the sensed target being a moving machine or human subject cannot be functional when restricted in terms of their motion. Sensor integration with wires in this scenario requires additional planning in cable placement, while possibly compromising the functionality of the target being sensed. However, wires do provide a stable power source and communication medium.

The alternative to the wired approach is wireless sensing, which utilizes individual wireless sensor units, called wireless sensor 'nodes'. Technological advances in electronics hardware, downward scaling of transistor size and expanding technologies in MEMS sensors and batteries have facilitated a progression towards low cost, low power wireless sensor nodes which are very easily deployed [6]. The sensor nodes typically consist of a sensing, computing and communications subsystem as well as a power source [7] as depicted in Figure 1.1, along with a commercially available Imote2 sensor node. The sensing subsystem consists mainly of the sensors and an analog to digital conversion (ADC) unit, and is responsible for converting the physical phenomena of interest into digital signal form. The computing subsystem is comprised of a processing unit which is usually a microcontroller unit (MCU) or digital signal processor (DSP) chip, their supporting electronics and peripherals. It controls all sensor node activities as well as performing some local processing. The communication subsystem houses the radio transceiver with the amplifiers and corresponding electronics. The entire sensor node which is capable of performing sensing, data acquisition, localized processing and wireless communication is usually powered by battery.

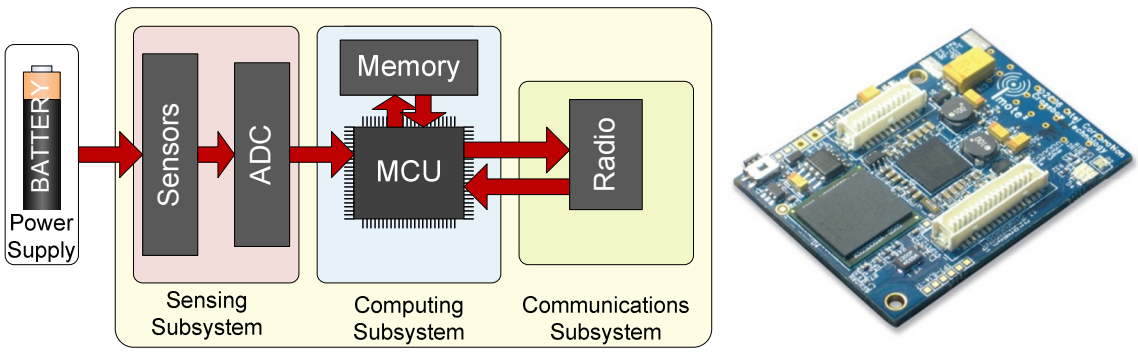


Figure 1.1 Wireless sensor node subsystems (Left) and commercially available node (Right)

Sensor nodes can be densely deployed and collaboratively monitor a target system, each sending information back to the base station or to other sensor nodes which relay information back to the base station [7] as depicted in Figure 1.2. This architecture is referred to as a wireless sensor network (WSN), where there is a large amount of research dedicated solely to investigating the communication and network topology in such a system [8]. Usage of a WSN to monitor some target system presents many opportunities over a wired implementation in addition to ease of deployment, such as the ability to leverage high node density to more accurately monitor the system, or to implement de-centralized decision making processes by forming clusters of sensor nodes.

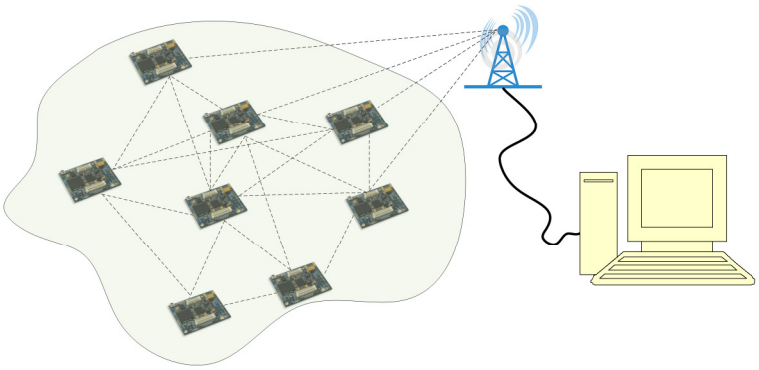


Figure 1.2. Adaptable topology in a wireless sensor network

These concepts have provoked an extensive amount of research in engineering and computer science disciplines, which seek to optimize the behavior of these wireless sensor networks and overcome obstacles. Research has led to development of efficient and secure communication protocols to be used in these networks, as well endless application of wireless sensing techniques. Additionally, small wireless sensing devices and sensor networks are finding place in areas previously inaccessible using wired sensors.

Although there are benefits to using a wireless over a wired approach, there is consistent demand and need to make these nodes smaller, more energy efficient and cost effective. Of these three needs, energy efficiency is of high priority; longevity is a deciding factor in many applications. Sensor nodes are typically powered by a low voltage battery which has limited energy supply, directly constraining the lifetime of the individual sensor nodes and the entire sensor network. The changing network topology and behavior within the network along with this energy constraint makes the WSN lifetime difficult to predict, introducing a concern of dependability.

Network lifetime and dependability are primary concerns especially in industry [9], which is estimated based on system behavior and node energy consumption properties [10]. In addition to dependability issues, limited node lifetime due to the power constraint poses significantly higher challenge when battery replacement is difficult. In heating, ventilation and air conditioning (HVAC) monitoring for example, sensor battery replacement requires unit disassembly which is timely and expensive [6]. Also, in medical applications sensors can be embedded in patient's bodies, requiring invasive surgery to replace batteries [11]. In environmental or military applications as well, sensor nodes may be dropped into a region without even having the possibility of replacing batteries.

There are existing and maturing technologies to increase the energy supply and node lifetime by scavenging energy from the environment through mechanical [12] or chemical [13] means. Although these sources may be capable of powering the sensor node at some times, the irregularity of their occurrence requires a battery to still be incorporated, and energy to still be used very sparingly. Minimization of energy usage in

wireless sensing devices therefore continues to be of high importance and to attract the attention of the research community.

1.2 Sensor node energy consumption

Before delving into techniques to improve energy efficiency of the sensor node, let us first examine where the energy is actually being consumed. The power consuming components are grouped based on functionality into the same subsystems displayed in Figure 1.1; the sensing, computing and communications subsystems. Figure 1.3 (Left) shows a distribution for power consumption amongst these subsystems for a typical sensor node [14]. The communications subsystem is usually the largest consumer of energy, which is expected to remain the case. This is because as Moore's law predicts the scaling down of power consumption of digital devices used in processing and data acquisition, the amount of power required for a transmitter correlates directly to desired transmission distance based on pure physics; efficiency improvements in radio circuitry can only go so far. The energy consumed by the communications subsystem can be estimated according to the transmission distance r, number of bits to be transmitted p_i during transmission event i, the sample resolution in bits per sample B, modulation setting for the transmitter b, and two hardware specific constants F and G are hardware constants as [16]:

$$E_{comm} = \frac{p_i \cdot B}{b}(F \cdot r^2(2^b - 1) + G) \tag{1.1}$$

The energy consumed by the computing subsystem is mainly due to the onboard processor such as a microcontroller, DSP chip or FPGA. These devices consume energy proportional to the amount of processing cycles as well as the processor frequency f,

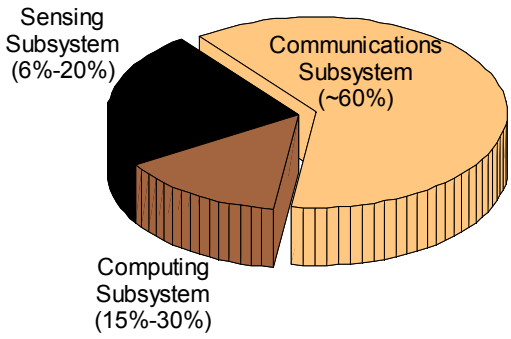
switching capacitance γ and hardware specific constants k and ε. The number of cycles required to perform a task on p samples are estimated according to the computational complexity $O(p)$, which describes how many basic operations i.e. additions, multiplications, etc. must be performed in executing the task. Knowing these parameters, the computational energy to complete a task can be calculated according to [17]:

$$E_{comp} = O(p) \cdot \gamma \left(\frac{f}{k} + \varepsilon \right) \tag{1.2}$$

In the sensing subsystem there are two possible consumers of energy; the ADC and the sensors themselves if they require power. The energy consumption of the ADC is typically proportional to the amount of samples acquired p and the sampling rate used [18]:

$$E_{ADC} = \frac{C_1 \cdot SR + C_2}{SR} \cdot p \tag{1.3}$$

Some power consumption properties for various sensing devices are shown in Figure 1.3 (Right) to give a general idea of the requirements. With a low rate ADC and passive sensors, the sensing subsystem will be one of the least consumers of energy. However, if higher rate ADC's or energy hungry sensors are required for the application, the power consumption of the sensing subsystem can quickly rival that of the communications subsystem [19].



Device	Function	Power
AD7789	16-Bit ADC, 16.6 SPS	400µW
AD7660	16-Bit ADC, 1KSPS	2
STCN75	Temperature Sensor	0.4mW
iMEMS	Accelerometer	30mW
T150	Humidity Sensor	90mW
CP18	Proximity	350mW

Figure 1.3. Typical power consumption distribution (Left) and power consumption of some off the shelf sensing devices (Right)

1.3 Approaches to power conservation

When it comes to reducing power consumption of wireless sensor nodes, the first initiative is to design efficient hardware. Downward scaling of transistors and advances in MEMs technology have been largely responsible for reducing the size and power usage of these devices [6]. However, incorporation of energy efficient techniques in designing the system's *behavior* further reduces each node's power needs. This is necessary to overcome fundamental limitations in reducing power usage of the hardware itself; there is a lower limit to reducing transmitter power, and components will always consume power if left in the 'ON' state. There is therefore a great deal of research aimed at optimizing sensor node activities to further reduce power consumption [19].

There are firstly techniques which are directly address optimization of the hardware itself, such as dynamic voltage scaling techniques (DVS) [20]. DVS reduces power usage of the processor by varying the processor supply voltage according to the computational load. The voltage controls the processor clock frequency, which is seen in Eq. (1.2) to directly affect energy consumption. A similar technique is applied to the

transmitter using dynamic modulation scaling (DMS) which varies the modulation setting seen in Eq. (1.1) according to the present data load.

In addition to developing more advanced hardware or directly optimizing its usage, there are many approaches which control sensor node activity as a means to reduce their power consumption. These techniques can be thought of as falling into three separate categories [15] as displayed in Figure 1.4. They are duty cycling, mobility based, and data driven techniques. Duty cycling techniques are effective in reducing power consumption by placing nodes or their individual components into sleep mode for periods of time. For example, exploiting node redundancy by placing certain nodes into sleep mode for periods of time can provide the same functionality while reducing the overall energy expenditure by two to three times [22]. Optimal wakeup protocols for the sleeping nodes must be introduced in this case in order to effectively communicate [23]. Energy efficient MAC protocols with low duty cycle also fall into this category [24].

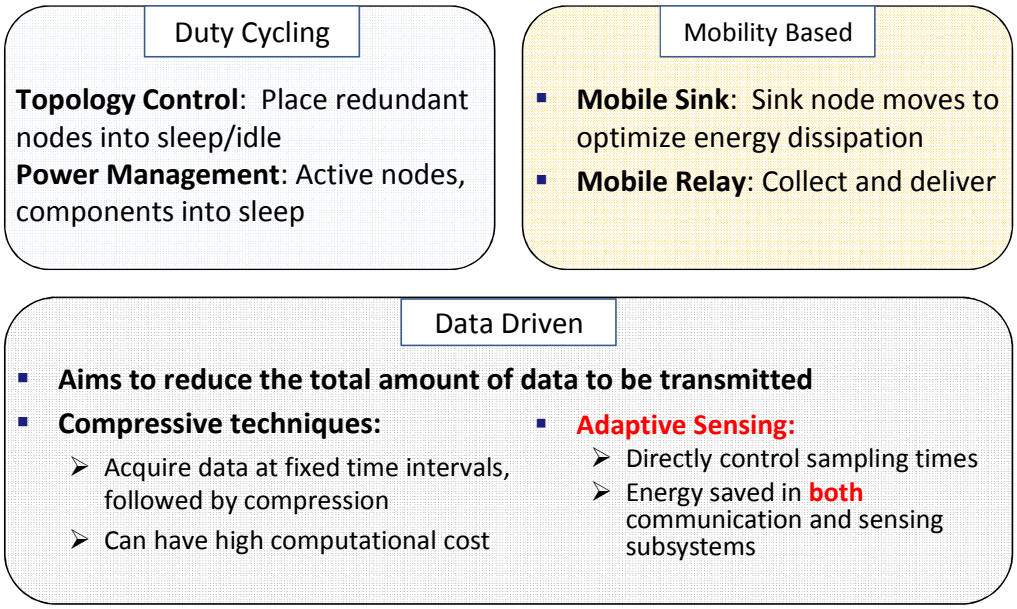


Figure 1.4. Three types of energy-efficient approaches

The approach taken in this thesis is referred to as a data driven approach, which mainly targets the largest consumer of energy on the sensor node; the radio and its circuitry. It is known that there is often redundancy in the form of spatial or temporal correlation of the acquired sample points [25], and the data driven approach focuses on removing this redundancy. In doing so, there are fewer samples to be transmitted by the radio, and hence less power is consumed.

There are two ways to accomplish this; compression and energy-efficient data acquisition, specifically adaptive sampling. In a compression technique, data is acquired at a fixed rate from the sensing subsystem and some process is performed to represent the signal with the smallest amount of data [26]. In this process, certain desired features can be extracted and transmitted rather than transmitting the entire signal [27], thus reducing energy expenditure. Adaptive sensing takes this one step further, with the mentality of reducing the amount of data directly at the acquisition stage. This allows further energy reduction in all sensor node subsystems if designed properly, because there are less samples acquired, processed and transmitted

CHAPTER 2

ENERGY-EFFICIENT DATA ACQUISITION

In a traditional data acquisition scheme data is acquired at fixed time intervals, referred to as fixed-rate sampling. The sampling rate is chosen according to Shannon's sampling theorem, which states that a signal must be sampled at a rate which is at least twice the maximum frequency present in the signal. On a sensor node, this is wasteful because of the varying levels of signal activity and changing frequency content; a fixed sampling rate must be chosen for the application's worst case scenario. Examining the sensor node's power consumption characteristics, it is evident that each subsystem would benefit in terms of energy efficiency if sampling rate could be reduced at times of low signal activity. This concept is realized through the development of alternative data acquisition, or sensing techniques.

2.1 Alternative Sensing Techniques

Efficient data acquisition for purpose of data or power reduction has been desirable since the dawn of the digital era. In wireless systems, it is unlike other approaches to energy efficiency such as using compression, data fusion or enhancing communication protocols; it is uniquely difficult in that decisions must be made regarding information which is not yet available. The problem presents a sort of paradox: in order to optimally sample an incoming signal, we need to know when samples should be taken as to reconstruct the signal from them. However, in order to know when samples need to be taken, we need to know the signal. Adaptive sampling and other alternative sensing techniques address this by making some inference from previously acquired samples in

order to choose future sampling rates, or making assumptions about the signal's content. There is of course some compromise, which involves a tradeoff between signal quality and energy efficiency.

In context of wireless sensing, the coined term 'adaptive sampling' is somewhat ubiquitous in that it encompasses a broad range of techniques. These techniques can be categorized into three groups by the type of problems they address. In WSNs, network level, *spatio-temporal* adaptive sampling [28]-[35] is employed to efficiently sense a given area by controlling node sampling times. At the node and network level, *compression-like* techniques [36]-[39] are also sometimes referred to as adaptive sampling. In these cases, fixed rate sampling is employed, but followed by a data reduction scheme to lessen the amount of data to be transmitted. Finally, there are techniques focused at the node level which control the sampling rate directly at the sensing stage; selecting times to activate the sensor and A/D converter. This thesis proposes an approach for this type of scenario, which will be referred to as a *node level adaptive sampling* technique [41]-[46].

2.1.1 Network Level Adaptive Sampling

At the *network* level, efforts to control node sampling are concerned with resource allocation and efficient sampling of a spatial region. In [28] for example, sensor nodes in a dense grid are activated adaptively using a recursive dyadic partitioning scheme in order to sense the field. This process, named Backcasting first obtains an estimate of the spatial field in a preview step, and then activates additional sensors in a refinement step to improve accuracy in boundary-containing regions. As a result, the data size is significantly reduced while maintaining error tolerances.

Comparable information-based techniques leverage a tradeoff between value and cost of sensing operations [29], and seek to optimize resource allocation within the network [29]-[32], [34]. Fisher information and Gaussian process regression-based error, for example is traded against energy consumption to select node sampling times [29]. Acquisition methods have been dually proposed with energy efficient data collection and multi-hop protocols [32], including for wireless sensor networks with real-time capability [30]. Recently, compressed sensing techniques have also been of interest for wireless sensor networks, largely because they are completely decentralized and do not require much a priori knowledge about the data [35].

2.1.2 Node Level Adaptive Sampling

At the node level, in adaptively sampling incoming signals from the sensors there are methods that either mimic compressive techniques, or are truly adaptive. *Compressive-like* techniques are also sometimes referred to as adaptive sampling. Several of these techniques such as send-on-sampling, integral sampling, gradient-based and predictor-based methods have been proposed for control applications [36]. In these methods, the actual sampling rate remains constant, but a sample is only transmitted when significant change has occurred. A similar methodology is used in model based techniques [37], [38], where an analytical model of the signal is built at the sensing node and sink node, and it is desired to be maintained. Information is only transmitted when the model begins to deviate from newly acquired data, at which point the model is updated.

In oscillatory signals, Shannon's theorem is used to remove unnecessary samples by analyzing bandwidth [39]. This was found useful in a hearing aid [39], where significant net reduction in data led to immediate savings in further processing. The benefit of these

techniques in terms of energy efficiency is clear, but leads one to beg the question, "Why not perform data reduction directly at the acquisition stage to maximize energy savings?"

This is the motivation for this work, which will be considered a *local adaptive sampling* technique. In the realm of this localized, real-time data acquisition reconfiguration, many adaptive schemes have been proposed. For signals with noticeable trends, a model based approach can sometimes be attractive. Subsequent sensor sampling times can be determined by the information value of the previous sample in terms of updating the model [29], [33].

For high information content signals, Nyquist's criterion is traditionally considered in selecting sampling rate. As formalized by Claude Shannon [40], discretization of band-limited signals in compliance with the sampling theorem is perfectly complemented by a reconstruction formula known as the cardinal series expansion, or sinc-interpolation. The theorem states that the sampling rate must be at least twice the highest frequency in the signal to ensure reconstruction. Knowing that frequency content varies in real signals, the concept of sampling at twice the short term, or 'instantaneous' frequency [42] is an intuitive means for adaptive sampling. This means that the sampling rate is continuously adapted according to an estimate of the current maximum signal frequency.

Hardware solutions to accomplish this have been proposed switch sampling rates based on signal frequency [42] and slope [43], as well as digital processing schemes to estimate maximal frequency from recently acquired samples and adjust sampling rate [41], [44], [45]. When using available samples to determine frequency, there are two obstacles, namely (1) the speed of the analysis, and most importantly (2) the inability to detect high frequencies after the sampling rate has been lowered. In [41], a Fourier transform reveals

small changes in maximum signal frequency in a snow monitoring application, but can only detect a rise within the slight amount of oversampling. [45] Proposes a similar approach leveraging a wavelet transform for the frequency analysis. The limitation can be illustrated by a simple example. Suppose the maximum signal frequency is 100Hz, and therefore the adapted sampling rate chosen to be 210Hz. If say, a 300Hz component emerges, it cannot be identified due to aliasing caused by sampling at 210Hz. This work delves into this challenge.

2.1.3 Compressed Sensing

Although considered non-adaptive, compressed sensing (CS) has attracted much attention from the research community in recent years as a way to address the data acquisition problem. It's development is connected with the 'Analog to Information' mindset, which reasons that if a signal is compressible, it should be compressed at acquisition to avoid wasting resources required to do so at a later stage [46]. Mathematical contributions have shown that this is possible to an extent, so long as the signal is sparse in some domain [47]-[49]. Essentially, if a signal is sparse or nearly-sparse in some domain Ψ and is measured in basis Φ, the number of randomly chosen samples needed to reasonably reconstruct it is based on its sparsity in Ψ, and the incoherence between Ψ and Φ [50]. Higher incoherence infers higher compressibility.

It is found that the reconstruction can be performed through methods such as l_1 minimization [52] and convex optimization [48], and the number of samples needed is often reduced below the Nyquist rate [48], [49], [51]. Surprising results from [48] indicate that perfect reconstruction is possible from a small number of frequency samples for a phantom test image, applicable to improve Magnetic Resonance Imaging techniques.

Compressive sensing may prove especially useful in high bandwidth applications where hardware limitations prevent sampling above the Nyquist rate, as backed by results using a proposed Random Modulated Pre-integration Sampling architecture [47], [49]. Radar applications for example, are target for these techniques [49]-[52], as well as streaming applications [53]. Selection in terms of the basis functions such that they best sparsify the signal [54][55] and remain incoherent is a topic of heavy research.

Although compressed sensing is a powerful solution in many applications, it does have its disadvantages. It is notable that it is non-adaptive to signal input, meaning that a priori knowledge about the signal in terms of its content must be available; an assumption of signal sparsity in a known domain is needed. Also, additional hardware is often needed in order to sample the signal. This is because if the signal is sparse or nearly sparse in the sensing domain, the signal must first be spread out in some other domain in order to capture it when sampling at lower rates. For signals with high information content or a changing amount of information content with time, compressed sensing is not the best solution. Therefore, the approach taken in this work is not compressive, but strives to maintain optimal sampling rate needed to preserve the active bandwidth, such that it can be analyzed at a later time.

2.2 Scope of Thesis

This thesis presents a new adaptive sampling technique to be used at the node level in adaptive sensing. Previously discussed limitations have been overcome through short bursts of high-rate sampling, which enable both reliable analyses to estimate appropriate future sampling rates, and net reduction in acquired samples. An analysis for this sampling rate selection has been formulated to be highly efficient in terms of

computational cost, which is shown to be implementable on real wireless sensing systems. Presented simulation and experimental trials confirm its usability, as well as performance in terms of data and power consumption reduction. Implementation challenges and solutions are also heavily investigated.

For the designed sampling technique, the resultant samples will not be spaced evenly in time. This is problematic, since most signal processing techniques require a signal sampled on an evenly spaced grid. This issue is addressed through the derivation of a signal reconstruction technique to produce a fixed rate signal from the non-uniformly acquired samples. The technique is shown to provide near-perfect reconstruction for this special case of non-uniform sampling, allowing the signal to be retrieved. Additionally, a post analysis has been formulated to quantify error in sampling rate selection, and therefore information loss.

CHAPTER 3

ADAPTIVE SAMPLING ALGORITHM FORMULATION

3.1 Algorithm Methodology

To further improve energy efficiency of wireless sensor nodes, adaptive sampling or more generally techniques focused on optimizing data acquisition are effective in that they reduce power consumption in sensing, data acquisition, processing and transmission. Varying sampling rate according to the qualities of the input signal is an intuitive approach, however it is very difficult in practice due to the fact that signal activity is not known until after sampling. Existing techniques aim to approximate signal activity in either a model-based approach or by tracking signal bandwidth. Several techniques have been proposed to track signal bandwidth and change sampling rate accordingly [41], [39][45]. However, a significant limiting factor is that once sampling rate is lowered, a sudden appearance of relatively high frequency content cannot be detected. For most applications, this is a serious limitation because frequency qualities of the signal may be very different at different times, and also change suddenly with change in events.

This problem can be addressed by introducing short, recurrent bursts of high rate sampling, and analyzing these samples to estimate signal frequency content. Doing so allows the entire bandwidth to be probed, as opposed to checking only the bandwidth visible when sampling at a lower, adjusted rate. Each burst of high rate sampling of length T_H is followed by a short interval of adjusted rate sampling of length T_L, where the sampling rate used here is calculated based on findings in the samples acquired at the higher rate. Using this method allows an ongoing estimate of frequency content to be

maintained and subsequently a better sampling rate adjustment, while still reducing the amount of data acquired. This results in lower power consumption[28] by the sensor node, and therefore longer lifetime.

This process is formalized into an adaptive sampling algorithm structure illustrated in Figure 3.1. In the adaptive sampling algorithm, data acquisition is performed in a total of i time intervals. Each interval begins with a short investigative sub-interval (referred to as the High Sampling Rate Period, or *HSRP* of duration T_H, in which a high sampling rate R_H is adopted. R_H is the sampling rate used if fixed rate sampling the signal. An analysis is then performed on the data points acquired during the *HSRP* to estimate the highest frequency component F_{max} in the signal. This analysis is performed while the system continues to sample at R_H. With the identification of F_{max}, the minimum required sampling rate R_{Req} is determined as $R_{Req}=c \cdot F_{max}$, where $c>2$ is a confidence factor chosen to satisfy Shannon's sampling theorem. The remainder of the interval i, is referred to as the Low Sampling Rate Period (*LSRP)* of duration T_L, in which the adjusted sampling rate R_L is adopted based on the minimum required rate R_{Req}. For increased robustness, R_L can into account the sampling rate in the k previous intervals:

$$R_L = \max(R_{Req}(i-l), l = 0..k-1 \tag{3.1}$$

A flowchart of the algorithm and an illustration of the sampling intervals, sub-intervals and resulting dynamic sampling rate are presented in Figure 3.1.

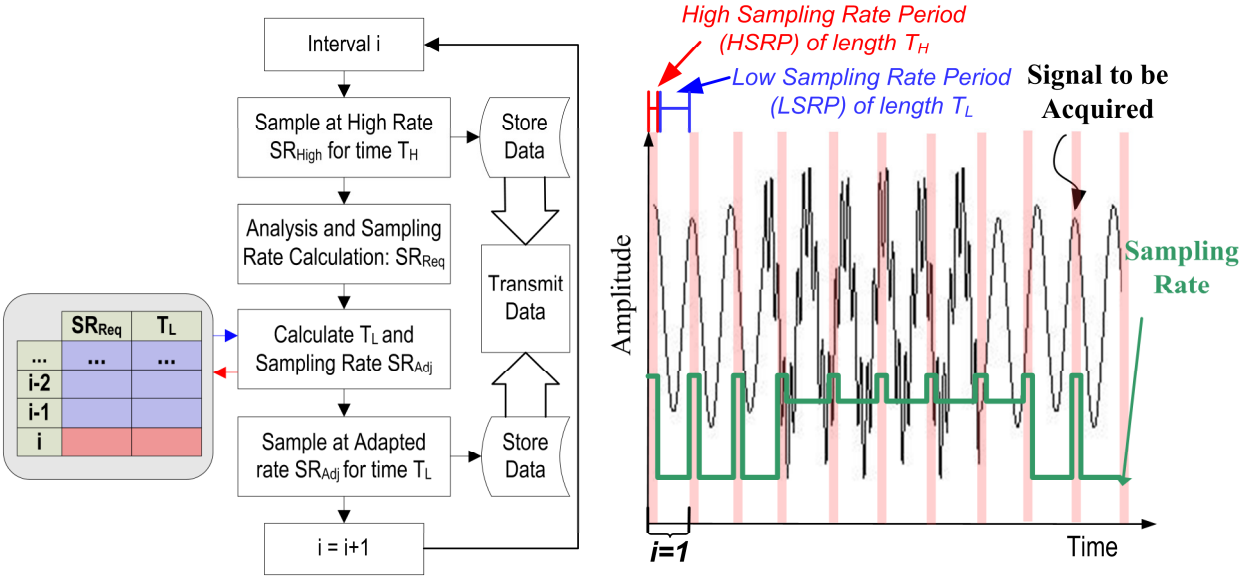


Figure 3.1. Flowchart and exemplified desired algorithm performance

With this structure in place, there are now two separate problems to address. The first is to formulate the analysis to be performed on the samples acquired at the high rate, which chooses the appropriate sampling rate for the remainder of the interval. This problem will be referred to as sampling rate selection. Secondly, the length of the *LSRP*, T_L must also be chosen, which governs how often the sampling rate is updated. By shortening this interval, performance in sampling rate adjustment may improve, but at the cost of lower energy efficiency due to the fact that more samples are acquired. This problem is referred to as analysis triggering.

3.2 Sampling Rate Selection

The basis for the algorithm is that sampling rate can be adapted during the acquisition process based on the signal's frequency content. Maximum frequency present in the signal during successive *HSRPs* of duration T_H is used to adjust the sampling rate for the

following *LSRP* of duration (T_L). The analysis performed on the samples acquired at the high rate has been designed to estimate maximal signal frequency and compute sampling rate in a fast, efficient manner.

3.2.1 Separation of signal into frequency bands

Following the acquisition of samples at this rate, wavelet packet decomposition is leveraged to break the signal into frequency bands. Shown in Figure 3.2, the wavelet packet transform decomposes the set of samples into separate frequency bands by recursively applying highpass and lowpass filters *G(z)* and *H(z),* where *G(z)* and *H(z)* are the z-transforms of finite impulse response (FIR) filters *g[n]* and *h[n]* [56]. The filters are desired to be power complementary, meaning the summation of their responses are equal to one over the frequency domain. This ensures that the all signal information is being examined equally.

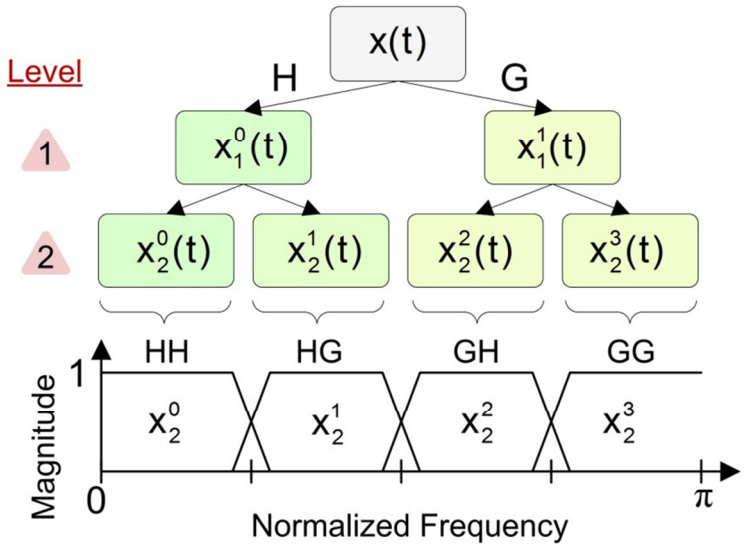


Figure 3.2. Illustration of a 2-level wavelet packet decomposition

This decomposition can continue downward for as many levels j as desired, generating $b = 2^j$ frequency sub-bands. These indexes j and b are used to identify each sub-band in the wavelet packet tree; $x_j^b(t)$. The lowpass and highpass filtering operations which generate the sub-bands are described mathematically as discrete convolutions:

$$x_{j+1}^{2b}[n] = \sum_m h[2n-m]\, x_j^b$$
$$x_{j+1}^{2b+1}[n] = \sum_m g[2n-m]\, x_j^b \tag{3.2}$$

Notice that the signal is decimated by a factor of two after each filtering operation, indicated by the $2n$ in each equation. This means that half of the coefficients are discarded after filtering, and the total number of coefficients in all bands at each level is about the same for each level. This decimated wavelet packet formulation enables greater computational efficiency by keeping the number of coefficients about the same for each level of decomposition, although the number of bands increases by a factor of two. This decimation is made possible because each filtering operation essentially halves the bandwidth of the previous band, allowing half the samples to be removed without loss of information [56].

However, the filters are never perfect halfband filters. Therefore, proper construction of the filters is critical to truly represent the signal in these bands, due to the aliasing effect induced by decimation and the slightly overlapping frequency responses as illustrated in Figure 3.2. Specifically, filters should be selected as the decomposition filters of a quadrature mirror filter (QMF) bank. These are two out of a set of four filters;

two decomposition and two reconstruction filters which meet the criterion of being power complementary as well as cancelling out any aliasing due to decimation [57]. A quadrature mirror filter bank in a one level decomposition-reconstruction structure is shown in Figure 3.3.

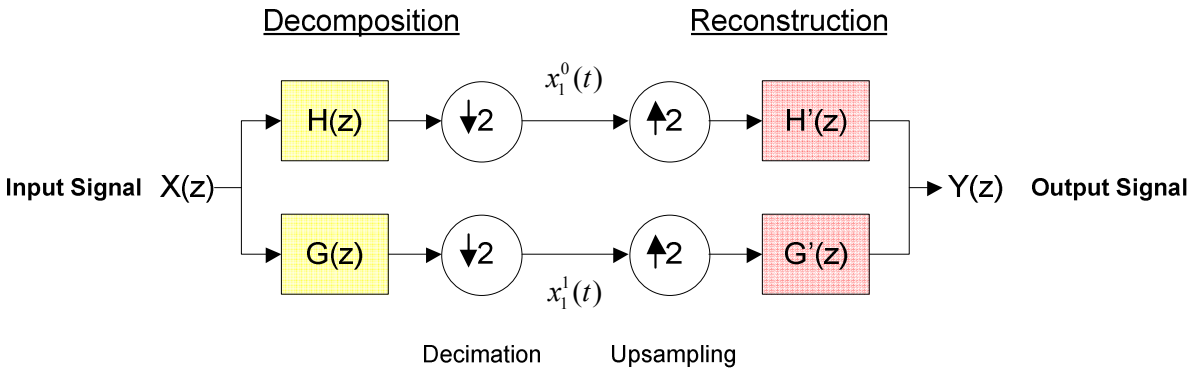


Figure 3.3. One-level wavelet decomposition and reconstruction

In designing these filters, the criteria need to be enforced. Let us analyze this filter bank shown in Figure 3.3 to see how this is accomplished. Recall that the z-transforms of the factor of *M* decimator and the factor of *L* upsampler shown in Figure 3.4 are given by [56]:

$$Y_d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X_d(z^{1/M} W_M^{-k}) \tag{3.3}$$

$$Y_u(z) = X_u(z^L)$$

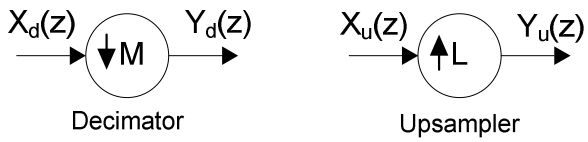


Figure 3.4. Factor of M decimator and factor of L upsampler

Here, $W_M = e^{-j2\pi/N}$. Let us first analyze the top half of the system shown in Figure 3.3, which is broken down as shown in Figure 3.5.

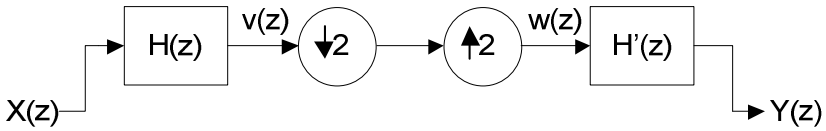


Figure 3.5. Top half of system to analyze

This system uses a factor of 2 decimator and factor of 2 upsampler, i.e. *L=M=2* in Eq. (3.3). Using this and substituting into Eq. (3.3), the relation between *v(z)* and *w(z)* is found:

$$w(z) = \frac{1}{2}\sum_{k=0}^{2-1} v\left(z^{\frac{L}{M}} W_M^{-k}\right) = \frac{1}{2}\left[v(zW_2^0) + v(zW_2^{-1})\right] = \frac{1}{2}\left[v(z) + v(ze^{j\pi})\right] \tag{3.4}$$

Since $e^{j\pi} = cos\pi + jsin\pi = -1$:

$$w(z) = \frac{1}{2}\left[v(z) + v(-z)\right] \tag{3.5}$$

Now, from Figure 3.5 it is evident that $Y(z) = w(z)H'(z)$ and $v(z) = X(z)H(z)$. Combining these with Eq. (3.5) solves the input output relation for the simplified system in Figure 3.5:

$$Y(z) = \frac{1}{2}\left[X(z)H(z) + X(-z)H(-z)\right] \cdot H'(z) \tag{3.6}$$

Seeing that the full structure in Figure 3.3 is symmetric i.e. the bottom half is the same as the top but with different filters, Eq. (3.6) is applied to solve for the whole system input output relation:

$$Y(z) = \frac{1}{2}[X(z)H(z) + X(-z)H(-z)] \cdot H'(z) + \frac{1}{2}[X(z)G(z) + X(-z)G(-z)] \cdot G'(z) \tag{3.7}$$

Rearranging to combine terms *X(z)* and *X(-z)*:

$$Y(z) = \frac{1}{2}[H(z)H'(z) + G(z)G'(z)] \cdot X(z) + \frac{1}{2}[H(-z)H'(z) + G(-z)G'(z)] \cdot X(-z) \tag{3.8}$$

Using the system response given in Eq. (3.7) the perfect reconstruction conditions can be enforced, ensuring that all information will be preserved in implementing this filter bank, even with the decimation. This is accomplished by first noticing that the second term with the X(-z) is the aliasing term and is desired to be zero. Therefore, the filters must be constructed to satisfy:

$$H(-z)H'(z) + G(-z)G'(z) = 0 \tag{3.9}$$

It is also desired that the output *Y(z)* be exactly the same as the input *X(z)*, with only some slack in the form of a time delay. This requires that:

$$H(z)H'(z) + G(z)G'(z) = 2z^{-l} \tag{3.10}$$

Here, l is any integer. Eq. (3.9) and (3.10) are referred to as the perfect reconstruction conditions for the quadrature mirror filter bank [58] whose implementation structure is shown in Figure. 3.3.

3.2.2 Thresholding and determination of sampling rate

The filters used to decompose the signal in the wavelet packet transform for our purposes will be the decomposition filters from a quadrature mirror filter (QMF) bank, as

to preserve the signal information. These filters will break the signal into 2^J frequency bands as previously discussed using the wavelet packet decomposition shown in Figure 3.2. Once the bands have been generated for some J level decomposition, where there will be 2^J bands, coefficients in all bands are subjected to a threshold δ which is a function of the data size p and noise variance σ taken from the first high pass band as [59]:

$$\delta = \sigma\sqrt{2logp}$$
$$\sigma = \frac{median(\{|x_1^1(T)|: \forall T \in x_1^1\})}{.6745} \tag{3.11}$$

Any coefficient not above this threshold is set to zero. Now, the highest band index b with any non-zero coefficient, denoted b_{max} is identified, which indicates an estimate of the highest frequency present in the signal. Using this index, an appropriate sampling rate for the signal, R_L is now computed where c is a constant greater than or equal to two:

$$R_L = c \cdot \frac{R_H}{2^J} \cdot (b_{max} + 1) \tag{3.12}$$

3.2.3 Sampling rate calculation algorithm for increased efficiency

Seeing that only one band needs to be identified i.e. the highest band with any content in it, b_{max}, the computational load induced by determining an appropriate sampling rate can been significantly reduced by avoiding unnecessary computations. The algorithm shown in Figure 3.6 accomplishes this task by selectively computing bands; computing only two bands at each level instead of 2^j bands at each level j. Essentially, at each level, one band is broken down into two bands; a low pass and high pass band. If there is content in the high pass band, the high pass band will be passed on and split again. If not,

the low pass band will be further split. Doing so allows the wavelet packet tree to be traversed; only computing two bands at each level and completely avoiding computing unnecessary branches.

A real time filtering implementation structure will be presented in Chapter 6 which allows for continuous execution of sampling rate calculation using this algorithm, as opposed to waiting until all high rate samples are acquired to begin computation. This has obvious advantages, namely that it gives a 'head start' in computation such that the result will be available even before the first sample at the adjusted rate needs to be taken.

Sampling Rate Calculation Algorithm

$$x_0^0(t) = S(t)$$

While $j < J$ && endloop = false

{

$$x_{j+1}^{2b}[n] = \sum_{m=-\infty}^{\infty} h[2n-m]x_j^b[m]$$

$$x_{j+1}^{2b+1}[n] = \sum_{m=-\infty}^{\infty} g[2n-m]x_j^b[m]$$

if $x_{j+1}^{2b+1}(t) > \delta$ for any t

$$x_J^b(t) = x_{J+1}^{2b+1}(t)$$

elseif f $x_{j+1}^{2b+1}(t) > \delta$ for any t

$$x_J^b(t) = x_{J+1}^{2b}(t)$$

else endloop = true

}

$$SR = c\frac{R_H}{2^j} \cdot (b+1)$$

Figure 3.6. Algorithm to efficiently compute appropriate sampling rate

A visual explanation of the sampling rate calculation algorithm is shown in Figure 3.7.

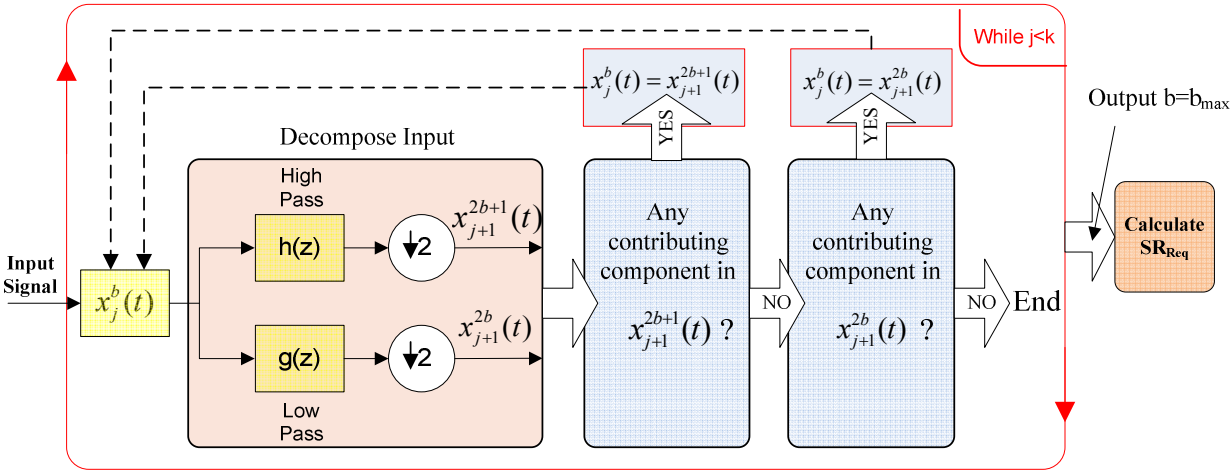


Figure 3.7. Visual illustration of efficient sampling rate calculation algorithm

By implementing this scheme, computational complexity is now linearly related to the number of samples needed to be processed. The computational complexity in actual filtering becomes only that of the discrete wavelet transform as approximated by:

$$O(p) \approx \sum_{j=0}^{J} 2\left[\frac{p}{2^{j-1}}C + C(C-1)\right] \tag{3.13}$$

In Eq. (3.13), p is the size of the data set, and C is the length of the QMF filter chosen. This result is obtained by counting the basic operations needed to perform the filtering, such as addition and multiplication. The advantage of selectively computing wavelet coefficients as opposed to computing the entire wavelet packet decomposition when calculating sampling rate is examined in Figure 3.8. The computational complexity and therefore execution time increases linearly with decomposition level in the WPT, due to the fact that the about the same number of coefficients are in each decomposition level (because of the decimation as previously discussed, it does not increase exponentially). By selectively computing only two bands in each level as proposed in this work, the

overall complexity and execution time is significantly reduced, as seen in the figure. This is important because with each successive level increase, the number of rates we have to choose from increases by a factor of two, allowing for greater energy savings.

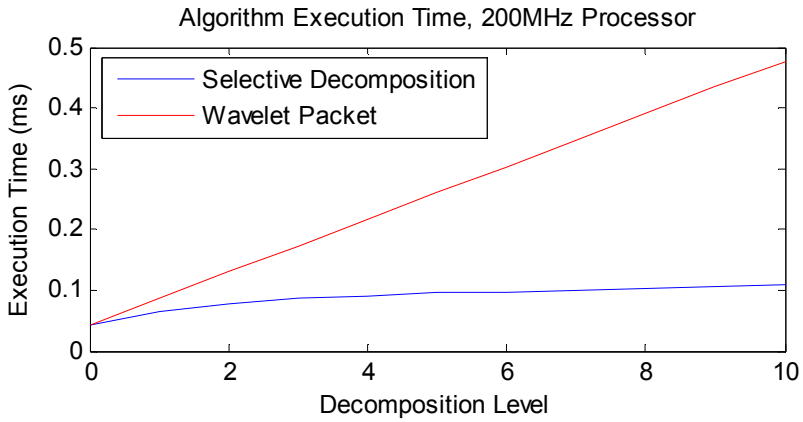


Figure 3.8. Execution time reduction through proposed sampling rate calculation scheme

3.3 Analysis Triggering

After an appropriate sampling rate for the signal R_L has been determined, the sampling rate is adjusted, or lowered, to this rate for some amount of time T_L. With some a priori knowledge about the physical phenomena being monitored, the first option is to set T_L beforehand to make sure that the sampling rate is updated enough based on how fast the signal is expected to change. A second option is to dynamically change the duration T_L according to how fast the signal is changing. Should changes in frequency be detected, shortening T_L enables allows the algorithm to respond quickly to further frequency changes in the signal. On the other hand increasing the time interval T_L lowers the data load and energy consumption. This is because increased T_L reduces the number of times the *HSRP* (which consumes more energy by acquiring and processing more samples) occurs in a given data collection event. However, the improved energy usage

comes at the cost of the possibility of overlooking high frequency events due to the larger time intervals between analyses.

The optimal *LSRP* duration T_L is therefore as long as possible to conserve energy, constrained such that the analysis is performed frequently enough to avoid missing high frequency events. Based on the application, the size of the interval can be limited such that it is less than the expected length of the event to be captured, guaranteeing that it will not be completely missed.

To do this, the *LSRP* interval duration T_L is updated in an on-line manner based how frequently the sampling rate needs to be adjusted. If the target signal behaves like a stationary signal the sampling rate selection algorithm is repeatedly suggesting the same adjusted sampling rate R_L, the *LSRP* duration T_L is extended to further conserve energy. Conversely, if fast changes in frequency content are detected, the *LSRP* interval length T_L decreases and HSRP periods occur more often. This issue is analogous to that of packet collision in communication protocols, in which the packet transmission rate is monitored and controlled to maximize usage of available bandwidth. In many industry standard protocols such as Transmission Control Protocol (TCP) [60] the Additive Increase Multiplicative Decrease (*AIMD)* method is used to vary the length of the congestion window [61]. The congestion window is a protocol paradigm that allows direct control of time spacing between packet transmissions.

Essentially, the length of the congestion window is additively increased until a packet collision is detected, at which point the window is multiplicatively decreased by some factor. This linear control algorithm has been extensively analyzed and has been shown

to converge to fairness [60], meaning that it evenly allocates bandwidth while minimizing collisions. The fairness index (F) is defined as [60]:

$$F(x) = \frac{(\sum x_i)^2}{n(\sum x_i^2)} \tag{3.14}$$

The fairness index (F) varies between 0 and 1, 0 being the least fair and 1 being the most fair. Here, x is the throughput from transmitter i and n is the total number of transmitters. In addition to AIMD other linear control algorithms have also been researched in prior work. These alternative algorithms include Multiplicative Increase Additive Decrease (*MIAD*), Multiplicative Increase Multiplicative Decrease (*MIMD*), and Additive Increase Additive Decrease (*AIAD*). Among these algorithms the AIMD control scheme was found to be optimal [62].

In this work, the AIMD algorithm is used to increase the *LSRP* duration T_L if the low sampling rate adjustment has been sufficient, and multiplicatively decreased if it has not. The low sampling rate adjustment is sufficient if sampling rate has changed more than the amount covered by confidence factor c, if not the adjustment is deemed insufficient. To determine sufficiency the required rates SR_{Req} of most recent intervals are compared. Criterion to indicate sufficiency combined with the AIMD control statement is mathematically stated as:

$$\begin{aligned} T_L(i) = \alpha \cdot I\left(R_L(i) > c \cdot \frac{R_L(i-1)}{2}\right) \cdot T_L(i-1) + \beta \\ \cdot \mathrm{I}\left(R_L(i) \leq c \cdot \frac{R_L(i-1)}{2}\right) \end{aligned} \tag{3.15}$$

Here, I is the indicator function whose value is 0 or 1 according to the Boolean statement inside the parentheses, c is the constant from Eq. (3.12), α and β are constants which control the amount of increase or decrease of the *LSRP* duration T_L. To illustrate the behavior of this interval length changing, an example is given in Figure 3.9 for a signal with arbitrarily increasing and decreasing sampling rate.

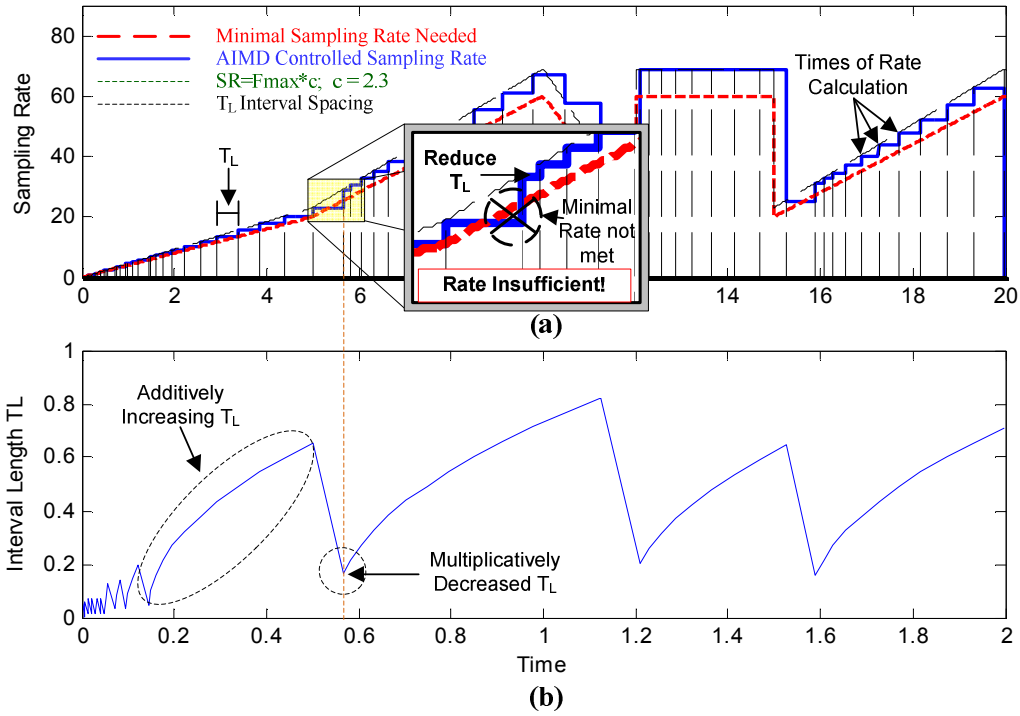


Figure 3.9. AIMD interval control: a) Sampling rate adjusted at the end of each interval (Top) b) Length of time T_L in each interval (Bottom)

In Figure 3.9, the green dashed line represents the value which the sampling rate is set to at the end of each interval. The blue line corresponds to the actual sampling rate which is adjusted at times governed by the AIMD algorithm. The red line represents the minimum sampling rate required to avoid aliasing, or slightly above to give some

confidence window. Any time the sampling rate used falls below this line, it is deemed insufficient.

It is evident that the time between sampling rate changes slowly increases when the sampling rate is sufficient, and decrease rapidly when the required criterion is not met. A rapid decrease is required to avoid missing information by analyzing the signal more frequently; so that the blue line will not cross the red for the example in Figure 3.8. This process results in the saw tooth behavior seen in the bottom part of the Figure 3.8. In the example the AIMD algorithm performs the desired function, ensuring that the sampling rate will be at least the minimum rate needed.

The estimated computation time for the entire adaptive sampling algorithm, consisting of sampling rate selection and interval length selection through AIMD, is estimated to be:

$$T \approx \sum_{j=0}^{J} 2\left[\frac{3p}{2^{j-1}}C + C(C-1)\right]T_m \tag{3.13}$$

Here, p is the number of samples acquired in the *HSRP*, C is the filter length, J is the number of levels in the wavelet decomposition and T_m is time to execute one machine cycle.

CHAPTER 4

SIGNAL RECOVERY

Through the adaptive sampling of any signal, it is desired that sufficient information is retained in order to reconstruct the continuous signal using its discrete samples. The adaptive sampling technique presented in this paper yields a signal sampled in a piecewise fixed rate manner, using uniform sampling rate for given time intervals in which the bandwidth-sampling rate criterion holds. This presents an interesting problem in reconstruction in that the sampling is not uniform, but is also not completely non-uniform or random. Utilizing this information, a reconstruction procedure is proposed which leverages properties of fixed rate sampled signals in conjunction with a non-uniform reconstruction framework. In this unique linear algebra-based approach, a near exact reconstruction is derived which complements the proposed adaptive sampling technique.

4.1 Signal Reconstruction and Interpolative Techniques

In a digital system an analog signal or function $f(t)$ is represented by a finite number of values; N samples taken at times t_n in a vector of the form $f(t) = \vec{\boldsymbol{f}} = \langle f(t_1), f(t_2), .., f(t_N) \rangle \in R^N$. This vector $\vec{\boldsymbol{f}}$ is obtained through a sampling process $\vec{\boldsymbol{f}} = \Phi f$ where Φ describes the specific sampling operation. It is of course desired to design this sampling process be reversible where $f(t) \leftrightarrow \vec{\boldsymbol{f}}$ such that the finite set of samples contains enough information to completely represent and reconstruct the original function. As is intuitive, the lower limit of samples needed to accurately

reconstruct the signal is dependent on the signal information content, and also the sampling method Φ [50]. For a bandlimited signal sampled at a fixed rate for example, Shannon's theorem perfect reconstruction by cardinal series expansion [40].

In a typical sampling-reconstruction framework, it is assumed that the continuous signal can be represented by the sum of basis functions which have been shifted and weighted according to the sample values. Correctly choosing the basis functions enables the signal value $f(t)$ to be computed for any time *t* from the set of samples $\vec{\boldsymbol{f}}$, referred to as *interpolation*. To reconstruct a signal acquired at a fixed rate this is expressed mathematically where *N* is the number of samples and n is sample number where $\forall n \in Z^N$ as:

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT)\varphi(t - nT) = \sum_{n=-\infty}^{\infty} \boldsymbol{f}[n]h_s(t - nT) \tag{4.1}$$

In a generalized sense, this can represent any reconstruction in a shift invariant space $V_\varphi = Span(\{\varphi(t-k)\}_{k\in Z})$, where V_φ designates the space of functions that can be represented by a linear combination of shifted and scaled basis functions φ. For the case of fixed rate sampling, it is proven from Shannon's theorem [40] that for bandlimited signals this equation can reconstruct the entire continuous signal where *T* is the sampling period and h_s(x) is a Sinc function given by:

$$h_s(x) = \operatorname{sinc}\left(\frac{\pi x}{T}\right) = \frac{\sin(\frac{\pi x}{T})}{(\frac{\pi x}{T})} \tag{4.2}$$

It is notable that this procedure which is also referred to as a cardinal series expansion is equivalent to an up-sampling of $\boldsymbol{f}[n]$ followed by a low pass filtering using the ideal

lowpass filter specified in Eq. (4.2). In either explanation, each interpolated point is merely the sum of Sinc functions, each one shifted and scaled to each sampling time and sample amplitude.

However, in practice this method of interpolation is not able to be implemented due to the finite length of $\boldsymbol{f}[n]$ and infinite support of the Sinc function. Thus, an approximation is developed by truncating filter h_s to be of finite length. However, by simply truncating the ideal lowpass filter; applying a rectangular window function, a ripple effect occurs which is known as Gibbs phenomenon [56]. In this phenomenon which can be seen in Figure 4.3, ripple occurs near the cutoff frequency and there is poor stop-band attenuation, even with a relatively high filter length. This cause unwanted artifacts in the interpolation, inducing large errors. The response and interpolation can be improved by applying a fixed window function around the Sinc function by $\bar{h}_s[n] = h_s[n] \cdot w[n]$, where $h_s[n]$ is the discrete form of the filter in Eq. (4.2) and *w[n]* is the fixed window function. This process is referred to as apodization. Figure 4.3 shows some common window functions (left) as well as their effect on the frequency response of the truncated Sinc function (right).

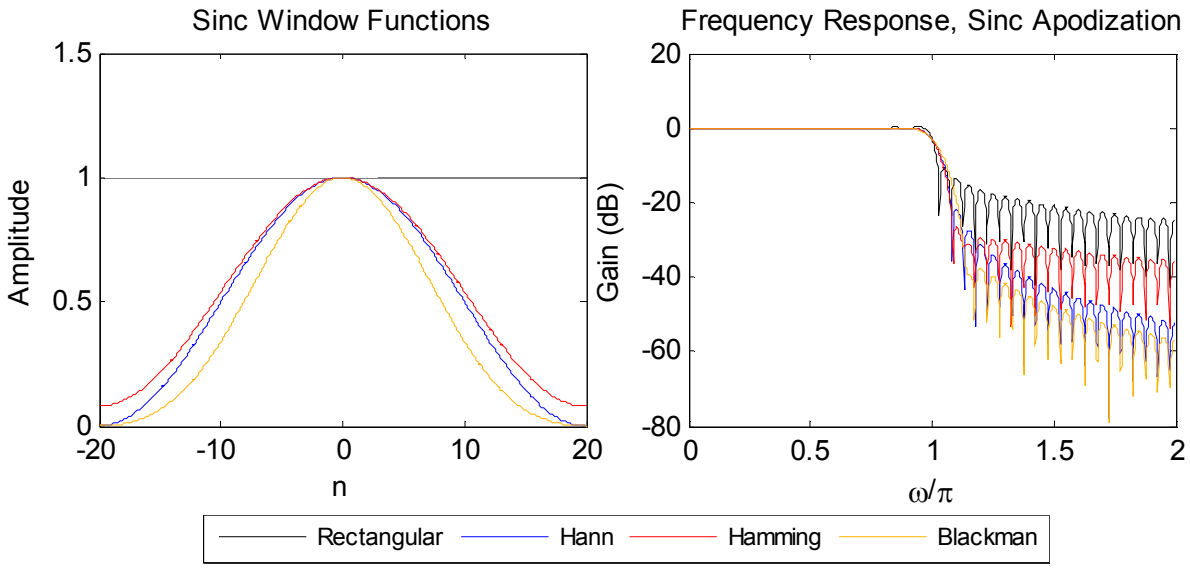


Figure 4.3. Common window functions (left) and their effect on frequency response of a truncated Sinc function, for support N=41 (right)

As seen in Figure 4.3 (right), application of the window function significantly improves the frequency response of the truncated Sinc function, therefore giving a better interpolation. In most cases, this is still sub-optimal due to the long decay and infinite support of the Sinc filter. The long decay leads to 'ringing' artifacts when the filter is truncated to short length, usually leading to the requirement of a large window size and subsequent filter length. This induces heavy computational load and still has limited performance. As an alternative, extensive research has been directed towards developing interpolation techniques with a better 'bang for their buck'. This means finding the most accurate interpolation, with the smallest possible basis support which directly affects computational complexity, and exploiting the tradeoff between these two.

To accomplish this, it is desirable to expand the selection of basis functions in searching for those with the coveted properties. It makes sense that the error at the

sampling values should be zero in the reconstructed signal, i. e. the interpolation constraint be imposed, however the classical formulation in Eq. (4.1) requires that φ be an interpolant, meaning $\varphi(0) = 1$ and $\varphi(nT) = 0, \forall n \neq 0$. The Sinc function in Eq. (4.2) exhibits this property, but the selection of other basis functions is very limited under this constraint. To get around this, generalized interpolation [63] extends the class of possible basis functions φ to those which are may be non-interpolating, while retaining the desired interpolation constraint in the signal reconstruction. This is accomplished by reformulating Eq. (4.1):

$$f(x) = \sum_{n=-\infty}^{\infty} d_n \varphi(\frac{x}{T} - n) \tag{4.3}$$

The evaluation of this equation requires an additional step, because the samples f_n have been replaced by coefficients d_n, thus the coefficients d_n must be determined first. Now, there is freedom to choose φ to be non-interpolating and to enforce the interpolation constraint when calculating coefficients d_n. To determine the coefficients d_n, the samples taken f_n are simply substituted into the left side of Eq. (4.3) with the selected basis function φ on the right and sampling times *x*. Now, the coefficients d_n can be found by solving the system of linear equations, which can be represented in matrix form as $\boldsymbol{f} = \boldsymbol{\varphi d}$, where $\boldsymbol{\varphi}$ is a square matrix with basis functions φ along the diagonal, forming a band-diagonal matrix. The coefficients can be obtained by solving the inverse problem $\boldsymbol{d} = \boldsymbol{\varphi^{-1} f}$. Efficient techniques have been developed to perform this specific matrix inversion with basis functions $\boldsymbol{\varphi}$ forming a band-diagonal [64].

Recently, [65] has proposed an alternative to matrix inversion by realizing that this procedure leveraging Eq. (4.3) is merely a convolution $f_n = \sum_{n=-\infty}^{\infty} d_n \varphi(t-n)$, which can be written $f_n = (d_n * \varphi)_n$. Subsequently, the convolution inverse φ^{-1} is leveraged to obtain the coefficients $d_n = ((\varphi)^{-1} * f)_n$, which is implementable through discrete filtering. The z-transform of such a filter is takes the form [63]:

$$\Phi^{-1}(z) = \frac{1}{(\varphi_0 + \sum_{n \in N} \varphi_k (z^k + z^{-k})} \propto z^m \prod_{i=1}^{m} (z - z_i)^{-1} (z - z_i^{-1})^{-1} \tag{4.4}$$

Following the determination of all coefficients d_n which enforce interpolation of the samples f_n, they are substituted back into Eq. (4.3) which can now be evaluated for any point $f(x)$. This is the procedure of generalized interpolation. A powerful result of this formulation is the equivalent interpolant of the non-interpolating basis function; an infinite support interpolating basis function can be handled by a non-interpolating, finite support one [63]. This is evident by substituting the equation for the coefficients back into Eq. (4.3), resulting in [63]:

$$\begin{gathered} f(x) = (\varphi)^{-1} * f * \varphi = \varphi_{int} * \boldsymbol{f} \\ \varphi_{int} = (\varphi)^{-1} * \varphi \end{gathered} \tag{4.5}$$

Thus, the operation performed by the finite length basis function is equivalent to an interpolating, infinite counterpart. This relation asserts some interesting properties which testify to the higher performance of kernels in comparison to the sinc kernel, namely that in the case of cardinal splines such as B-splines the equivalent interpolant can be shown to converge to a sinc function as the order increases [68].

4.2 Choice of Basis Function

Proper selection of the basis function has the most critical effect on the performance of the interpolation, in terms of both the reconstruction error and the computational cost. Although directly interpolating functions which are 'close' to the Sinc function would seem to be the best choice, the desirable constraint of small support is too restrictive to allow the desired performance [68]. Also, 'close' is not specific, either quantitatively or qualitatively.

Through the generalized interpolation formulation, it is assumed that the interpolator itself can be represented as a sum of scaled and shifted versions of some finite, not necessarily interpolating basis functions. This sum is referred to as an equivalent interpolator. Using the generalized form will result in very high approximation order with smallest possible basis support.

With this newly found freedom revealed by a generalized formulation, there exists an opportunity for designing new basis functions. Currently, the best known interpolation techniques leverage polynomial splines, in particular B-splines and Maximum Order of Minimal Support (MOMS) interpolation [67][68]. B-splines have a plethora of interesting properties; one which is especially useful here is that they provide maximal order of approximation for a given support [63]. The family of B-splines, each of order *n* and function β^n is given by [63]:

$$\beta^n(x) = \sum_{k=0}^{n+1} \frac{(-1)^k (n+1)}{(n+1-k)!\,k!} \left(\frac{n+1}{2} + x - k\right)_+^n \tag{4.6}$$

These are the centered B-spline functions, where $(x)_+^n$ is the one sided power function defined as $(x)_+^n = \max(0, x)^n$ [63]. The width or support of the basis function of degree *n* is *n+1*, and the first 4 degree B-spline functions are shown in Figure 4.4. The first two degrees are extremely simple computationally; degree 0 corresponding to the 'nearest neighbor', and degree 1 to 'linear' interpolations. However the price is paid in terms of poor reconstruction quality. As degree increases, the B-spline curve converges to a Gaussian curve, and its equivalent interpolant to a Sinc function. For degrees greater than 1 the B-spline basis functions are non-interpolating as seen in Figure 4.4, meaning they must be used with the generalized interpolation in Eq. (4.3) to enforce the interpolation constraint. This provides much better performance than using a windowed sinc through apodization [63].

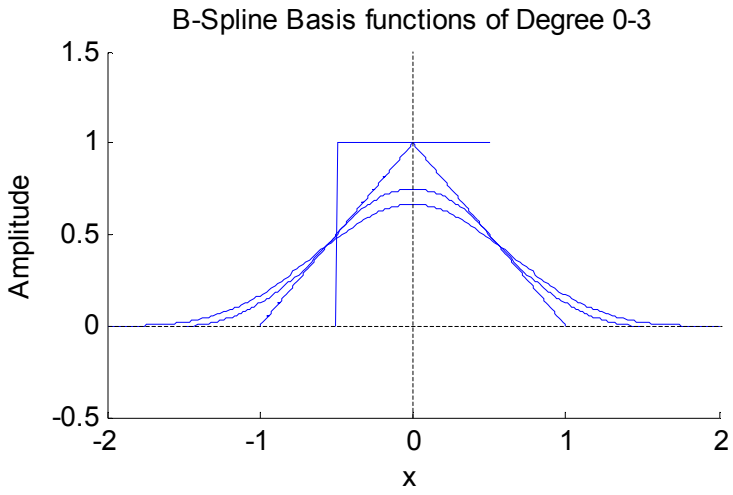


Figure 4.4. First 4 B-Spline Basis functions, degree 0-3

Further improving performance, MOMS, or Maximum Order Minimal Support kernels have been developed to be used in the same generalized interpolation implementation structure. These kernels minimize the support needed to attain a given

order of approximation L, and are the set of functions which are the sum of B-splines and their derivatives [68]:

$$\varphi(x) = \sum_{n=0}^{L-1} \lambda_n \frac{d^n}{dx^n} \beta^{L-1}(x - a) \tag{4.7}$$

Construction of these functions is investigated in [68], which presents optimal maximal order with minimal support functions (O-MOMS) and also a formulation which is directly interpolating (I-MOMS). As discussed previously, directly interpolating kernels are restrictive and sub-optimal; it is advantageous to instead use a non-interpolating kernel in the generalized interpolation formulation.

4.3 Reconstruction from the Adaptively Sampled Signals

Let us now investigate the unique reconstruction problem which results from sampling a signal adaptively using the previously formulated adaptive sampling scheme. First, recall that the sampling scheme consists of recurrent bursts of high rate sampling each followed by intervals of adjusted rate sampling. The adjusted sampling rate used in these intervals is calculated according to the frequency content which is identified through the presented analysis. Therefore, we will assume that at all times, in both the high and adjusted rate sampling intervals that the signal is nearly bandlimited to a frequency below half the sampling rate. An example sampling rate curve is shown in Figure 4.5, which illustrates the resultant sample spacing.

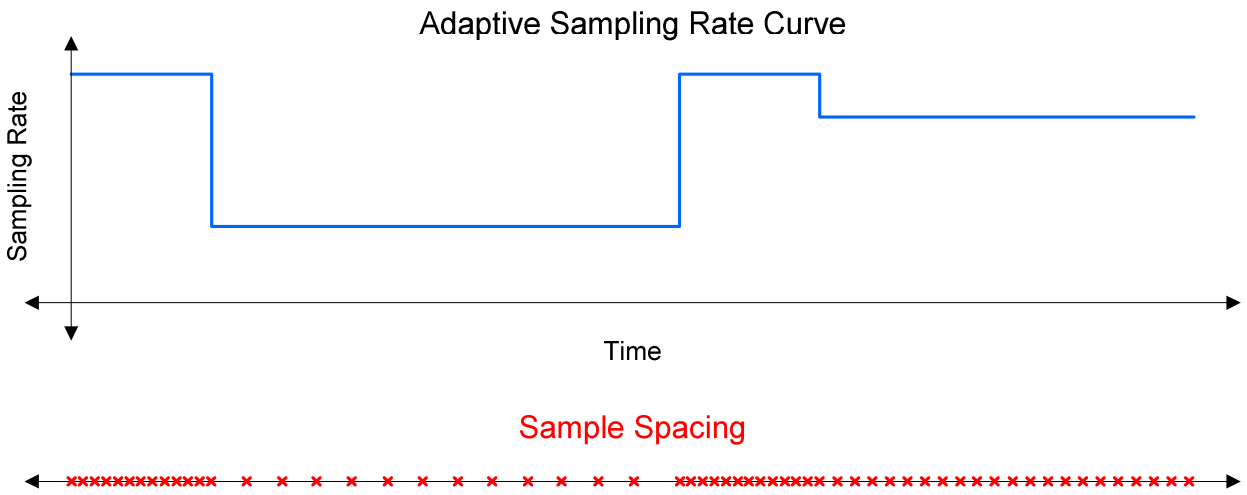


Figure 4.5. Sample spacing when using adaptive sampling algorithm

As depicted in the figure, the sampling rate is changing, but is constant for short periods of time. Being that the signal has been sampled according to the instantaneous frequency content, the signal content is preserved and the signal can be up-sampled back to be of one uniform sampling rate. However, how should this up-sampling, termed reconstruction, be accomplished? The sampling scheme supports reconstruction in a shift-invariant space as previously discussed, however the interpolation methods require a signal with fixed sampling rate to successfully perform reconstruction to a continuous signal.

In a sense, this specific case can be considered as non-uniform sampling. There are developed interpolation techniques for reconstruction of a signal from non-uniform samples, such as Lagrange or non-uniform B-spline interpolation, which are polynomial approximations. However, directly utilizing a non-uniform approach does not take advantage of the additional knowledge of frequency content which has been used in sampling the signal, and the reconstruction will not be precise. An alternative is to use

the interpolation kernels discussed in Section 4.2 to reconstruct the signal in a piecewise manner; to up-sample each section separately and then combine the results to yield one uniform signal. This is satisfactory when using kernels with small support, except for the distortion that occurs at the boundaries; at the beginning and end of the sample set the interpolated points may contain large error due to the lack of points on one side.

Recall that the reconstruction consists of summing scaled and shifted basis functions as shown in Figure 4.6. As visible in the figure, taking only the sample set with uniform rate will result in the lack of shifted basis functions to the left or right side. The error will affect a region within the support width of the basis function, which becomes significant when the high or adjusted rate intervals become smaller, and problematic in inducing noise every time the sampling rate changes. Unwanted artifacts and discontinuities will be at the boundaries, which are sampling rate change times in the final reconstructed signal.

In formulating a solution for this problem, consider the case shown in Figure 4.6 with one boundary, where the sampling rate changes from a high rate R_H to a lower rate R_L. There are N samples in the second interval which need to be up-sampled to the same rate as in the first interval, R_H; this is the reconstruction. The samples are spaced at time intervals denoted by the 'x's, and the reconstruction is in a shift invariant space using some basis function φ. The basis functions or their equivalent interpolants φ are shown scaled to the sample values. As discussed previously, if the samples to the left of the boundary are not available error will be induced. Examining the picture, the circles depict these needed samples, which are not directly available at these times because of

the change in sampling rate. However, these points are attainable by using the surrounding samples.

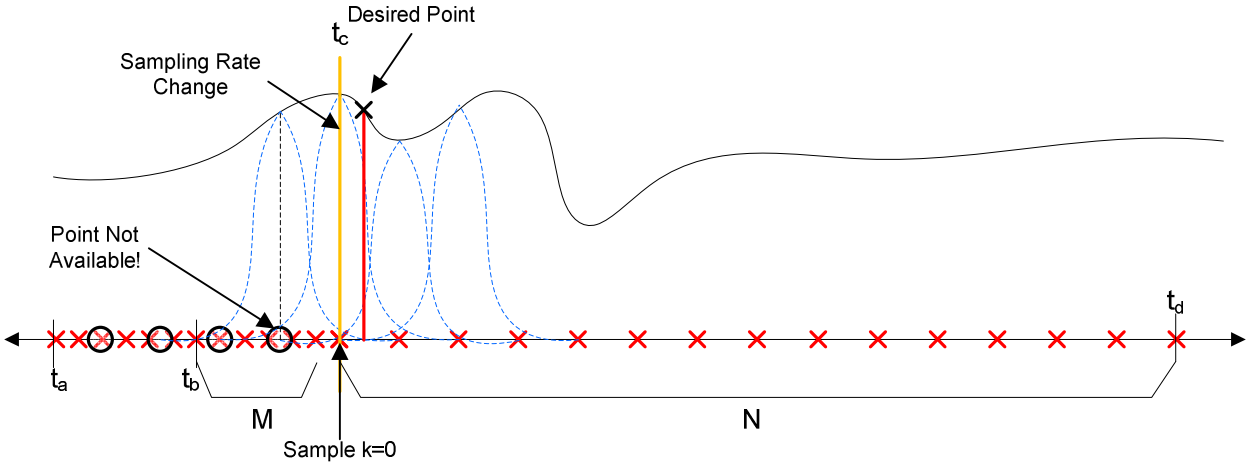


Figure 4.6. Signal reconstruction problem near boundary

Instead of taking the *N* samples alone to perform the reconstruction, we take also the M previous samples, where $M = \lceil (s-1)R_H/2R_L \rceil$ for odd basis support *s*. Now, consider this set of samples as a vector $\boldsymbol{x}^T = \langle x(t_1), x(t_2), .., x(t_L) \rangle$ which is of length *L=N+M*. The first step is to obtain a set of uniformly spaced samples, or coefficients sampled at the low rate R_L from time t_b to time t_d. By expanding Eq. (4.3), this can be accomplished by performing the interpolation in reverse. Expanding Eq. (4.3):

$$
\begin{aligned}
x(t_1) &= x[k]\varphi(t_1 - kT) \\
&\quad + x[k+1]\varphi(t_1 - (k+1)T) + \ldots x[k+s-1]\varphi(t_1 \\
&\quad - (k+s-1)T), k = \left\lfloor \frac{-2b}{T} + \frac{t_1}{T} \right\rfloor \\
x(t_2) &= x[l]\varphi(t_2 - lT) + x[l+1]\varphi(t_2 - (l+1)T) \\
&\quad + \cdots x[l+s-1]\varphi(t_2 - (l+s-1)T), l = \left\lfloor \frac{-2b}{T} + \frac{t_2}{T} \right\rfloor \\
x(t_3) &= \cdots \\
&\ldots
\end{aligned}
\tag{4.8}
$$

Here, T=1/R_L and $b = (s-1)2R_L$. In matrix form, this yields:

$$
\begin{bmatrix} x(t_1) \\ x(t_2) \\ x(t_3) \\ \vdots \\ x(t_L) \end{bmatrix}_{Lx1} = \begin{bmatrix} — & — & & & \\ — & — & — & & \\ & — & — & — & \\ & — & — & — & \\ & & : & : & : \\ & & & — & — \end{bmatrix}_{Lxm} \begin{bmatrix} \hat{x}[0] \\ \hat{x}[1] \\ \vdots \\ \hat{x}[m] \end{bmatrix}_{mx1} \tag{4.9}
$$

$$
or\ \boldsymbol{x}_{Lx1} = \Phi_{\mathrm{Lxm}} \hat{\boldsymbol{x}}_{mx1}
$$

The coefficients $\hat{x}$ are evenly spaced from time t_b to t_d with sampling period $1/R_L$, and the elements of matrix Φ are values of φ evaluated according to Eq. (4.3) and times at the corresponding row on the left hand side of the Eq. (4.9). Essentially, this means that the acquired samples are the result of a linear combination of scaled and shifted basis functions, which are governed by some coefficients on a fixed sampling grid. From the method through which the signal is sampled, it is known that $L>m$ and also that the maximal gap between any two samples is $1/R_L$. Therefore, Eq. 4.9 is simply an over-determined system of linear equations which is solvable through a matrix inversion $\hat{\boldsymbol{x}} =$

$\Phi^{-1}\boldsymbol{x}$. Since the matrix is not square, a pseudo-inverse can be implemented, or sampling points can be removed to make the matrix square since the system is over-determined.

This approach fits directly into the generalized sampling formulation of Eq. (4.3), where the coefficients $\hat{\boldsymbol{x}}$ are substituted as d_n's. Now, the interpolation is performed using Eq. (4.3) with the coefficients and basis function as inputs, giving the signal up-sampled to a uniform sampling rate of R_H. The result is combined with original samples between t_a and t_b to yield the complete signal sampled at one uniform, high sampling rate R_H to complete the reconstruction. Using the generalized reconstruction formulation, the selection of basis functions φ for this procedure is extended to the higher performance non-interpolating ones such as B-splines or O-MOMS as discussed. Also, implementing the case depicted in Figure 4.6 to the adaptively sampled signal requires addressing two boundaries; the sampling rate change on the right back to a higher sampling rate. This is done by simply taking an additional M samples on the right side of the sample sequence and performing the procedure in the same way.

4.4 Performance Demonstration and Comparison

An analytical test signal made up of several harmonic components is sampled using changing sampling rate, which varies in a way that represents the behavior of the adaptive sampling algorithm. The test signal is shown in Figure 4.7 (Top) with the sampling points. As seen in the figure, there is first an area with dense concentration of samples, signifying high rate sampling as verified in Figure 4.7 (Bottom) in the sampling rate curve. Then, there is an interval of lower rate sampling, and this process repeats.

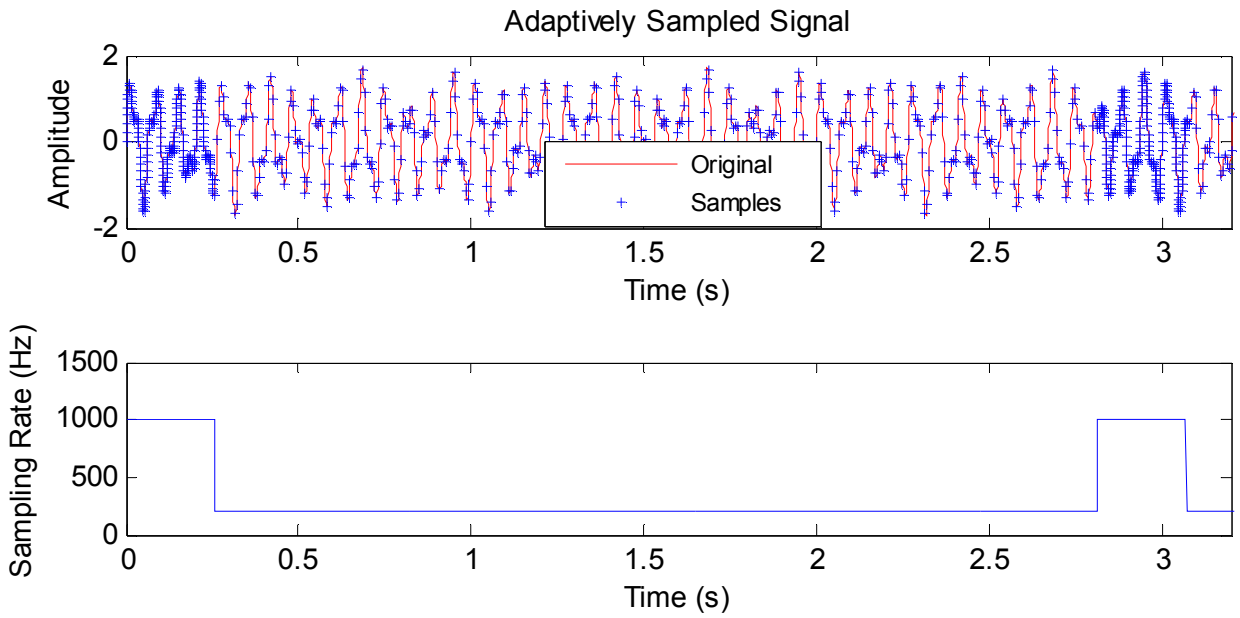


Figure 4.7. Example signal sampled adaptively (Top), and sampling rate used (Bottom)

From the samples, it is desired to up-sample the signal to be of one uniform rate. The first method investigated is a piecewise interpolation, where each section of uniformly sampled signal is up-sampled separately, using a sinc kernel given by Eq. (4.2), then combined to yield one uniform signal. The result is displayed in Figure 4.8. From an initial glance at Figure 4.8 (Top), the two signals seem indistinguishable. However, zooming in on the boundary area or examining the reconstruction error in Figure 4.8 (Bottom) reveals significant error. Far away from the boundaries and during the high rate sampling areas the error is very low, but increases drastically as approaching the boundaries. For this signal, the overall reconstruction error yields a root mean square error (RMSE) of 2.96e-4.

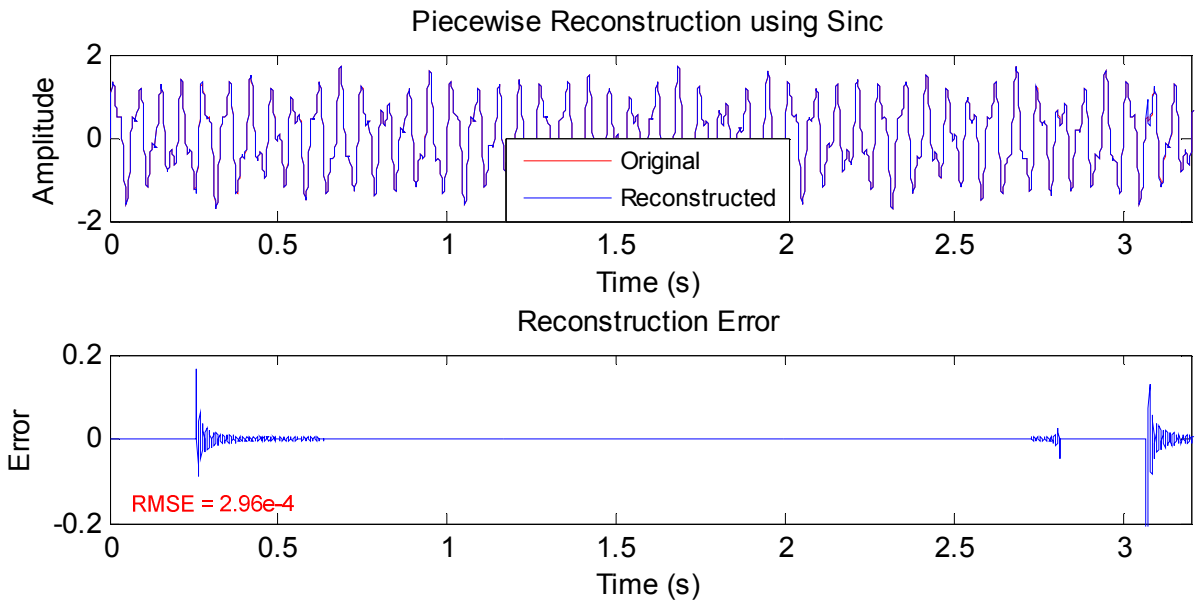


Figure 4.8. Signal reconstruction using piecewise Sinc interpolation

The new method presented in Section 4.3 addresses this issue at the boundaries and improves error by using a generalized interpolation approach. The results from this interpolation are given in Figure 4.9, using a B-Spline basis function of degree 15.

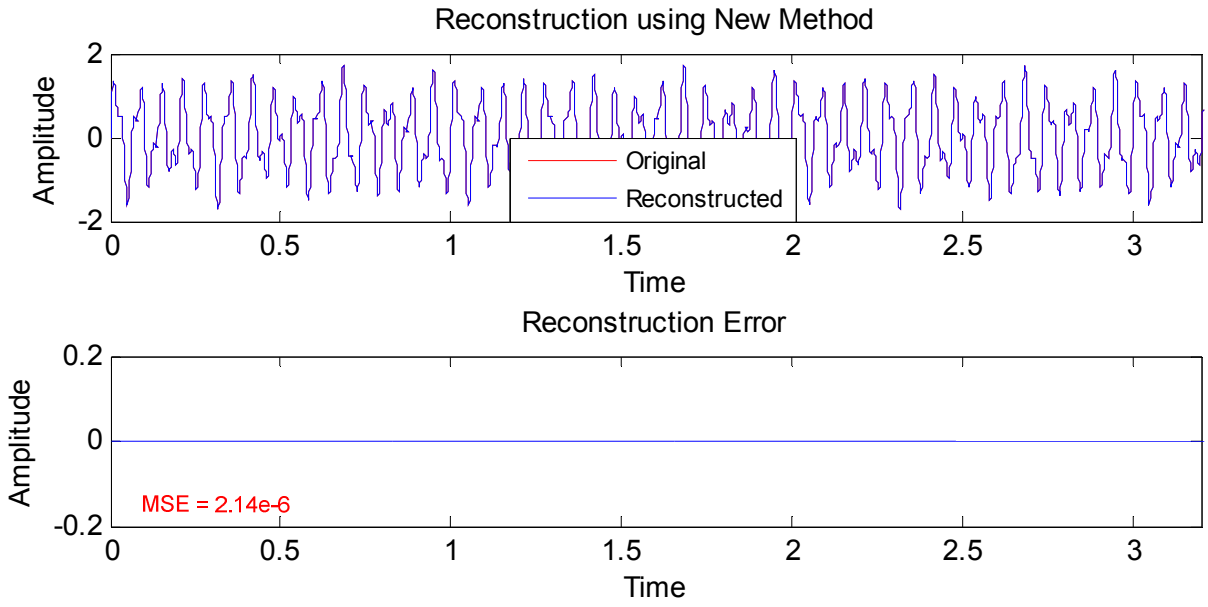


Figure 4.9. Signal reconstruction using new method

Using the new method, the issues at the boundaries are almost nonexistent and RMSE improved to 2.14e-6; about 2 orders of magnitude. This is due to the fact that using the new method, an additional M samples are able to be used at each boundary, as discussed in Section 3.4. To better see the difference at the boundaries, both reconstruction techniques are displayed in Figure 4.10 against the original signal; zoomed in at the sampling rate change times. Since the piecewise Sinc reconstruction is only able to use the samples acquired at a uniform rate, there is a hard boundary where the sampling rate changes, which subsequently leads to the visible Sinc response at this time as seen in Figure 4.10. Alternatively, since the new method first generates a set of coefficients beyond these boundaries, and the smaller support basis functions can be used, the reconstruction near the boundary is almost exact.

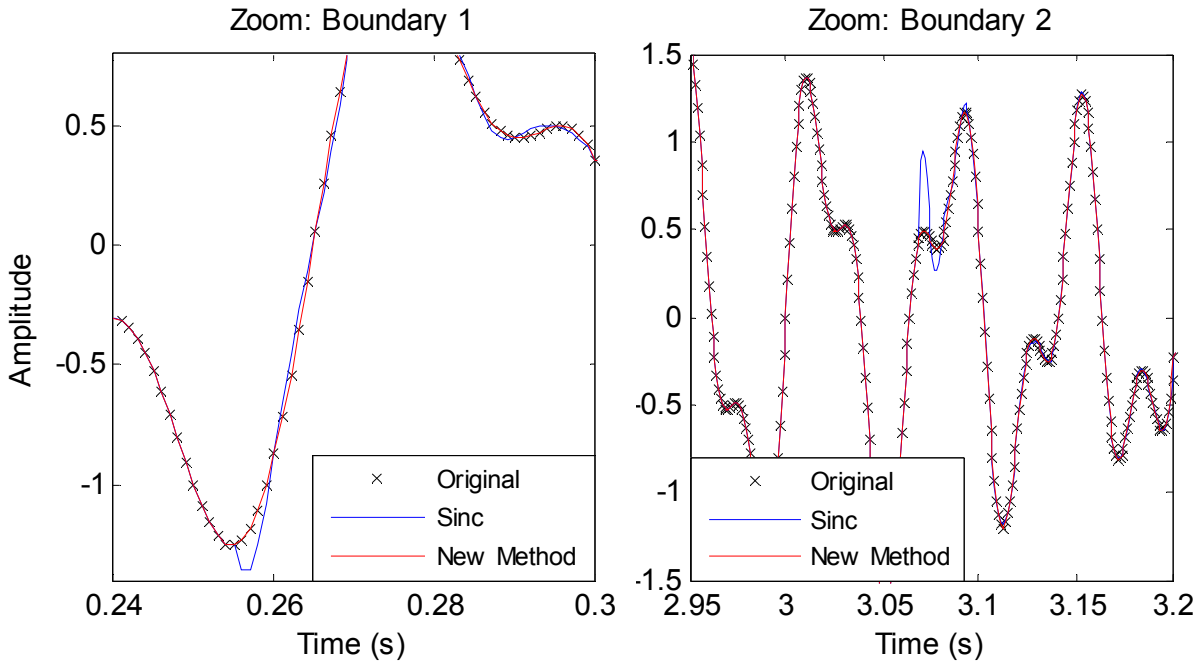


Figure 4.10. Zoomed-in view of the reconstruction techniques near sampling rate change boundaries

The new method is clearly advantageous over the piecewise Sinc reconstruction, both in terms of performance and computational complexity. For the Sinc reconstruction, maximal support is used; the convolution sum in Eq. (4.1) incorporates all samples in the interval in each summation. This is expensive computationally, and still does not give as good of performance. Alternatively the example leverages a B-Spline curve of degree 15. The smaller support of this basis function is only $n+1=16$, but gives remarkably better performance in the generalized interpolation. As discussed previously, this is due to the powerful result of the generalized interpolation formulation; using it, the response of an infinite support basis function can be achieved using only a small support, finite basis function.

For this test signal, the RMSE for this signal reconstruction using B-Spline basis functions of several different degrees is displayed in Table 4.1. Using a B-Spline of at least degree 2 in this formulation achieves substantial performance over the piecewise Sinc method. The improvement in reconstruction error with B-Spline order can be seen in Figure 4.11, which shows the signal reconstruction between 0.1s-1.0s using B-Spline basis functions of degree 0, 1, and 2. Degree of zero corresponds to nearest neighbor interpolation, which provides a very efficient but very poor interpolation. The error is seen to decrease with degree 1, corresponding to linear interpolation, and substantially with degree 2.

Table 4.1. Reconstruction error using B-Splines of different degree

	B-Spline Order				
	n=0	n=1	n=2	n=5	n=10
RMSE	2.40e-3	6.25e-4	6.42e-5	6.57e-7	1.129e-6

4.5 Summary of Results

A new method has been presented to reconstruct signals which exhibit fixed sampling rates for short periods of time. This method which leverages generalized interpolation techniques enables signals acquired using the proposed adaptive sampling to be accurately reconstructed, or up-sampled, to one uniform rate. The method is compared against the traditional bandlimited, Sinc signal reconstruction formula, which can only be implemented in a piecewise manner in this case of adaptive sampling. The new method is shown to have several advantages over the piecewise reconstruction, namely the elimination of boundary effects, lower error and smaller basis support.

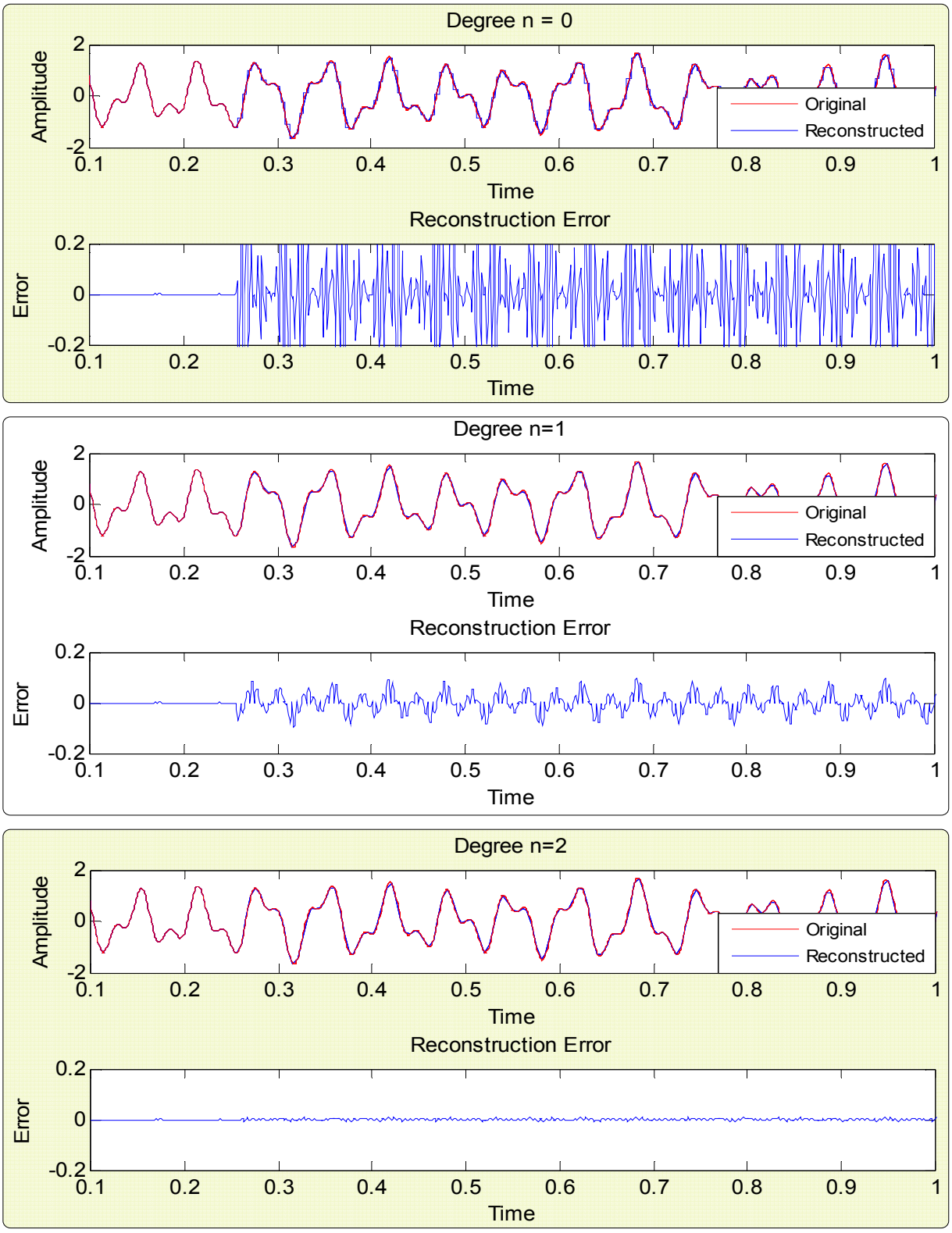


Figure 4.11. Reconstruction using new method, B-Splines of degree 0, 1, 2

CHAPTER 5

PERFORMANCE EVALUATION

Now that the adaptive sampling algorithm has been formulated, and a formulation has been presented to reconstruct signals acquired in this manner, the performance of the proposed adaptive sampling algorithm is evaluated for two separate cases. In the first case, synthetic vibration signals exhibiting milling chatter are subjected to adaptive sampling, where energy consumption is compared against the fixed rate sampling case. An energy model given in Section 1.2 is incorporated to calculate energy consumption of a commercially available sensor node performing the adaptive sampling. Secondly, a case of human physical activity monitoring is investigated, using a separate energy model for a wireless integrated measurement system.

In both cases, a custom error metric is needed in order to quantify the performance in terms of sampling rate adjustment. Root Mean Square Error (RMSE) and other standard indicators are effective in quantifying error in signal reconstruction, because they indicate the time domain difference between the original and new signals. However, quantifying performance of sampling rate adjustment is not as straightforward. This is because the signal contains some noise, which does not necessarily need to be included in the final signal for the sampling to be successful. By lowering the sampling rate when only desired low frequency components are present, some high frequency noise will be absent in the final signal because the bandwidth has essentially been cut in this area. This is fine since the desired signal is preserved, but the signal sampled at a higher rate can look

different in time domain; the noise filtering can cause a time domain discrepancy even when the desired signal is preserved.

Therefore, a custom method has been formulated to extract the desirable signal in time-frequency domain from the original signal, and generate a curve which specifies the minimal required sampling rate as a function of time. The sampling rates selected by the adaptive sampling algorithm are compared against this minimal curve, and the error metric is calculated, which specifies the percentage of time which the sampling rate selection has been sufficient.

5.1 Error Metric Formulation

The first step in obtaining the error value is to extract the desirable information from the original signal and determine what sampling rates are needed as a function of time. It can be thought of as extracting the 'instantaneous' frequency content, and calculating a sampling rate curve based on this; the rate needed at any time is dependent on the signal frequency content at that time. The desired time-frequency content is identified with the aid of a continuous wavelet transform (CWT). In this transform, a wavelet is scaled to different widths and convolved with the signal. The wavelet acts as a bandpass filter, where the center frequency is dependent on the scaling of the wavelet. The result is an intensity map with scale on the y-axis and shift parameter on the x-axis. Mathematically, the CWT is represented for scale a_j and shift b_k values [69]:

$$X_{WT}(b_k, a_j) = a_j^{-1/2} \int_{-\infty}^{\infty} x(t) \Psi^* \left(\frac{t-k}{a_j} \right) dt \tag{5.1}$$

Here, X_{WT} the are the resultant wavelet coefficients, $x(t)$ is the signal, and Ψ is the wavelet function. In this standard formulation, the actual shifting governed by k will be

different at different scales. It is desired that the transform be evaluated on an even sampling grid by using only parameter k, i.e. evaluated at the same times for all scales. This is for use with the test signal which is originally sampled at a high, uniform rate, which requires the substitution:

$$a_j = a_0^j b_k = k b_0 a_0^j \tag{5.2}$$

Variable a_0 defines the base scale, whose value affects the center frequency of the wavelet with respect to scales chosen a_j. Shift and initial shift b_k and b_0 are similarly related. In computing the CWT, evaluation on an even sampling grid is now achieved by simple iterations through a_j and b_k. Through the substitution, the transform is now defined as:

$$X_{WT}(b_k, a_j) = a_0^{-j/2} \int_{-\infty}^{\infty} x(t) \Psi^*(a_0^{-j} t - k b_0) dt \tag{5.3}$$

We will use the complex Morlet wavelet for its tight bandwidth, useful in isolating frequency components. If there are certain signal features that are desired to be preserved, entropy based techniques can be applied to select a wavelet that will best detect their presence. Knowledge of the wavelet's center frequency and scaling allows frequency components to be identified, and subsequently sampling rate calculated. First, recall the complex Morlet wavelet [69]:

$$\Psi(y) = \frac{1}{\sqrt{\pi f_b}} e^{2j\pi f_c y} e^{-\frac{1}{f_b} y^2} \tag{5.4}$$

Here, f_c is the wavelet center frequency and f_b is the bandwidth parameter commonly selected as 2. In the example analysis, the bandwidth parameter is increased to 10 to increase resolution in frequency domain. Essentially, this widens the window function

around the sinusoidal terms, narrowing the bandwidth of the resultant wavelet function. The frequency centered in this bandwidth is referred to as the center frequency. Since the wavelet will be scaled in the transform, the center frequency will change. The new center frequency will be referred to as the prominent frequency. We must know this change in order to correlate findings in the wavelet coefficients to frequency, and from viewing Eq. (5.1) it is evident that scale and frequency are inversely related. In an exact relation, from Eqns. (5.3) and (5.4) it is evident that the prominent frequency at scale a_j, which is F_c, wavelet center frequency f_c and base scale a_0 are related through:

$$F_c = f_c a_0^{-j} \tag{5.5}$$

After some manipulation, prominent frequency at scale a_j can be related to the wavelet center frequency, scale and original fixed sampling rate of the test signal SR_t:

$$F_c = \frac{f_c}{a_j} SR_t \tag{5.6}$$

At all times, the minimal sampling rate is determined by the maximum frequency present in the signal, which corresponds to the minimal scale. To find the minimal scale which is present at all times, the energy content of each wavelet coefficient from X_{WT} in Eq. (5.3) is investigated, which is determined by:

$$E\big(X_{WT}(b_k, a_j\big) = \big\|X_{WT}\big(b_k, a_j\big)\big\| \tag{5.7}$$

Using a threshold δ, the prominent wavelet coefficients are extracted by setting all coefficients with energy below the threshold to zero, representing the desired signal content. It is desired to find the highest frequency components at each time of evaluation b_k, which requires us to look at the energy content in each scale at that value b_k. Recall that the prominent frequencies F_c at each scale are inversely related to the scale value

through Eq. (5.6). Therefore, we must find the minimal scale at each b_k for which a coefficient above threshold δ exists. Choosing b_0=1, shift values b_k correspond directly to times t. Now, a function of minimal scale at each time t, deemed $A_m(t)$, can be derived as:

$$A_m(t) = minimize(a_j) subject\ to: E(X_{WT}(t, a_j)) \geq \delta(a_j) \tag{5.8}$$

The minimal sampling rate requirement as a function of time $SR_{Req}(t)$ is then derived through the help of Eq. (5.6) which relates scale to prominent frequency. First, Eq. (5.5) is rewritten to yield:

$$j = \frac{\log_{10}(a_j)}{\log_{10}(a_0)} \tag{5.9}$$

SR_{Req} is now presented by using Eq. (5.8) and j from Eq. (5.9), adhering to Nyquist's criterion:

$$SR_{Req}(t) = 2f_c a_0^{-j(t)} \tag{5.10}$$

The adaptive sampling rate guided by the algorithm SR_{Alg} is compared against the required rate SR_{Req} of Eq. (5.10) from the fixed-rate sampled signal $x(t)$. For all discrete times at which SR_{Req} is evaluated, it is determined whether the required sampling rate is maintained, i.e. $SR_{Alg} \geq SR_{Req}$. The time this inequality is satisfied is Co_{sat}, and violated Co_{vio}. Percentage of time proper sampling rate was not maintained given by Eq. (5.11) multiplied by the compromised bandwidth serves as the error metric.

$$\%SR_{Er} = \frac{Co_{vio}}{Co_{vio} + Co_{sat}} * 100 \tag{5.11}$$

This process of calculating sampling rate selection error and bandwidth loss is displayed using an example in Figure 5.1. Figure 5.1 (Top Left) shows the simple original signal which consists of a 2Hz signal with an enveloped 70Hz signal between 5 and 10 seconds. Figure 5.1 (Top Right) and (Middle Right) show the CWT before and after thresholding, and (Middle Left) displays the thresholded CWT which has been converted from scale to frequency on the y-axis using Eq. 5.6. (Bottom Left) shows the envelope over this frequency information multiplied by a constant 2, giving the minimal required sampling rate curve. This curve is copied into (Bottom Right) where it is compared against the sampling rates used by the adaptive sampling algorithm. The times where the used rate drops below the required rate signifies error. The area made by the crossing of the two curves for the whole process is calculated, indicating the amount of bandwidth that the signal occupies which was compromised. For this simple case, this compromise indicates 0.8% total loss. This method will be used in quantifying error in sampling rate adjustment for the test cases.

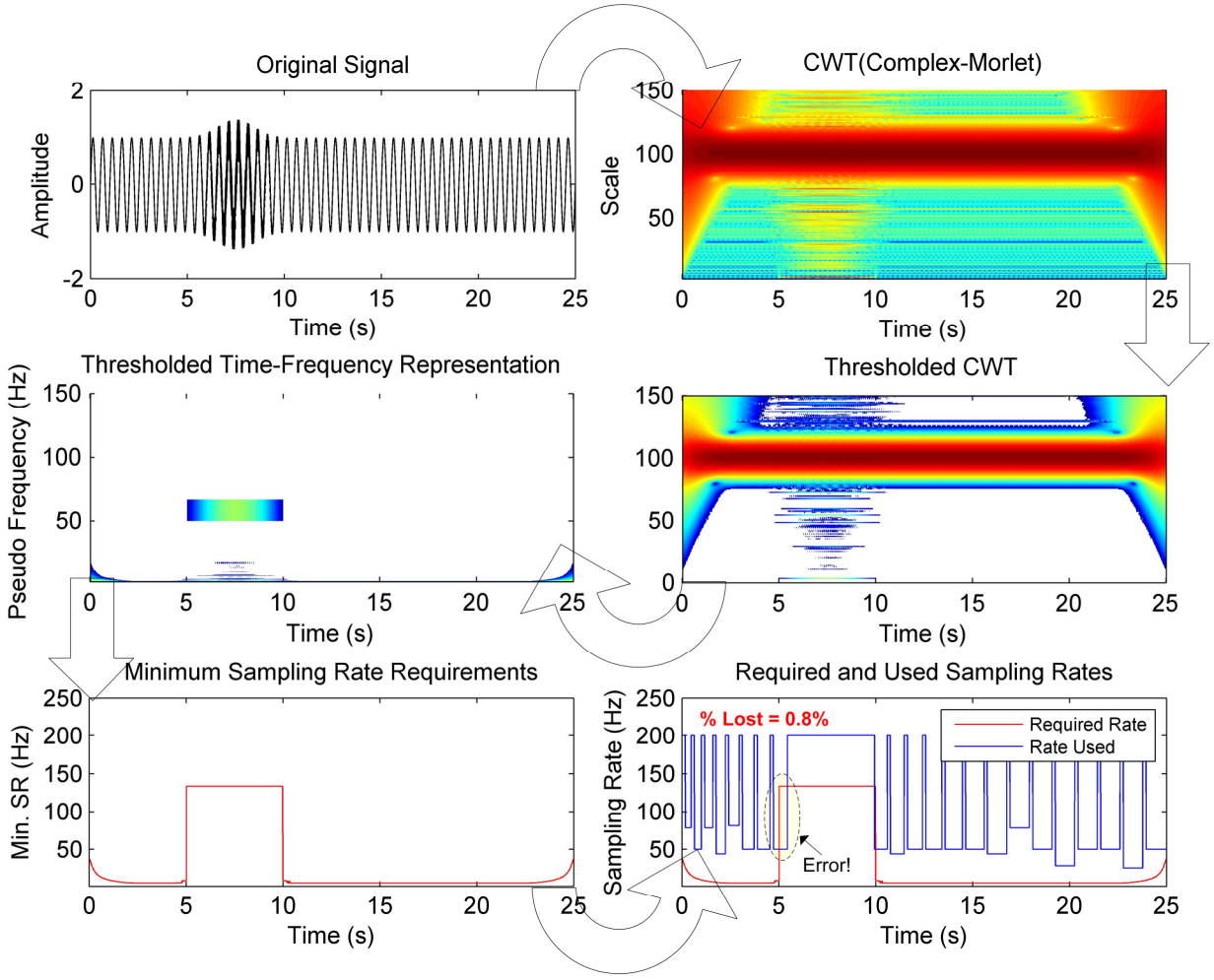


Figure 5.1. Example calculation of sampling rate selection error and bandwidth loss

Based on the algorithm description in Chapter 3, we can see that this error can be introduced due to two separate reasons. The first of which is error in sampling rate selection. This means that during an interval of high rate sampling and analysis, some desired signal content was not detected and the sampling rate was not high enough for the following low sampling interval *LSRP*. This can be avoided by proper threshold setting to detect these components, introducing further statistical methods if necessary in high noise situations as an alternative to hard thresholding.

The second possible cause of error is due to the 'one step behind' nature of every adaptive sampling algorithm; we can only make decisions on future sampling rate usage based on the samples that have previously been acquired. In the proposed algorithm, the sampling rate is adjusted for some small amount of time in the *LSRP*, which is then quickly followed by another short burst of high rate sampling to re-analyze the signal for new frequency components. This allows for a net reduction in energy in samples taken and as a result, energy consumption. The net reduction results in extension of node lifetime.

However, this is at the cost of a known amount of latency in detecting newly emerging frequency components; the amount of time between the analyses, which is the length of the *LSRP*. This amount of time is either fixed due to user settings, or allowed to fluctuate based on signal activity using the previously discussed AIMD scheme, which still has known upper and lower bounds. The user must choose an acceptable latency for the target application, manipulating the tradeoff between detection latency and energy reduction.

Figure 5.2 displays this tradeoff for a commercially available system, operating in an example scenario with high rate of 5 kHz. Average SR specifies the average sampling rate used throughout the entire acquisition; which is specific to the input signal. On the second axis, ratio of T_L to T_H reflects the amount of time in adjusted sampling mode to in analysis mode. This is directly related to the latency; a higher ratio indicates more time is spent in adjusted sampling mode, i.e. the *LSRP* is longer and latency is higher. The tradeoff is easily noticed in this figure: Although the average required sampling rate may low, adaptive sampling cannot lead to energy savings if the ratio is low because there is

little time spent in adjusted rate sampling mode. However, the detection latency is minimal. Alternatively, if the low sampling interval can be extended and some latency allowed, energy savings in excess of 5x can be achieved. The tradeoff between latency and energy savings should be carefully evaluated for each case.

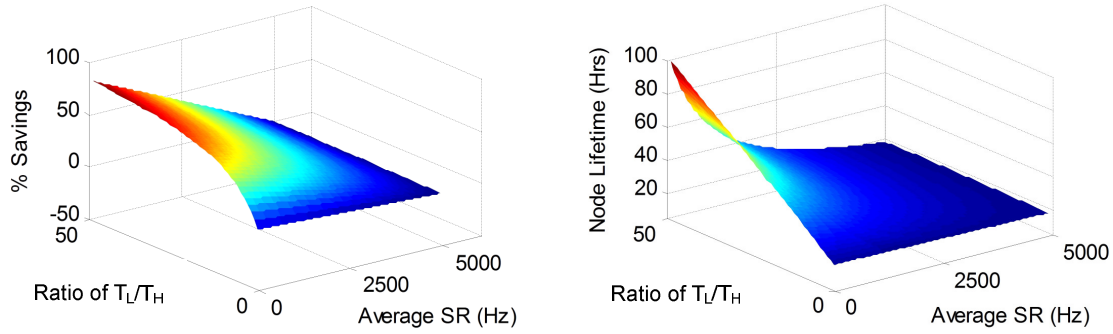


Figure 5.2. Energy/Latency tradeoff for commercially available node

5.2 Simulation Method Overview

Now that an error metric has been developed, the adaptive sampling algorithm is initially tested through simulation by adaptively sampling test signals and evaluating performance. The simulation mimics the real time sampling performed by a sensor node by taking consecutive 'samples' from an already acquired signal. Instead of the ADC taking a sample, the operation is replaced by interpolation; reconstructing the test signal and taking samples at requested points.

The proposed adaptive sampling algorithm is used to guide the sampling rate in resampling the signal, where all decisions are made causally. The simulation methodology is presented in Figure 5.2, which displays this first step of adaptively sampling the test signal. Based on the adaptively sampled signal and knowledge of sampling rate used as a function of time, energy consumption for a sensor node

performing this task is evaluated through the incorporation of a sensor node energy model. Error analysis is performed by the custom error metric describing how much of the used bandwidth occupied by the signal was compromised due to inadequate sampling rate adjustment. Figure 5.3 gives a visual idea of this process

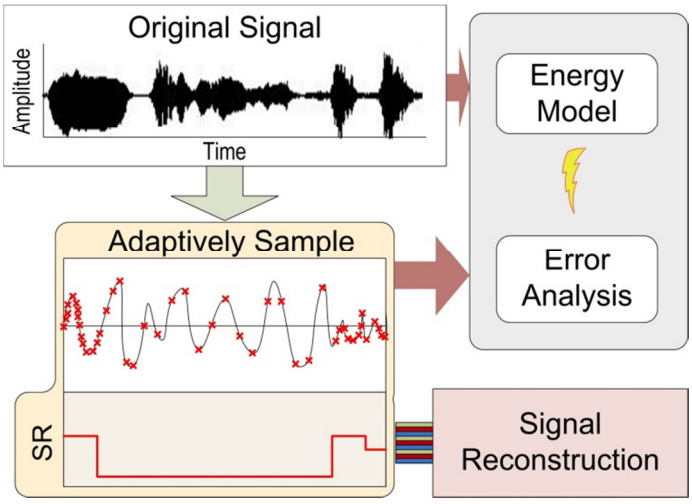


Figure 5.3. Illustrated Simulation Method

5.3 Adaptive sampling of real signals

The adaptive sampling algorithm has been evaluated in simulation for two separate scenarios. Synthetic signals modeled after a milling process exhibiting chatter are used as input [70], as well as experimentally acquired human physical activity monitoring data [71]. The frequency-varying nature of these signals makes them ideal candidates for adaptive sampling. Filter coefficients tabulated in [72] are the Quadrature Mirror Filters used for the sampling rate selection mechanism, and are of length 16.

5.3.1 Case 1: Milling Chatter

Synthetic signals modeled after milling processes exhibiting chatter were used to test the algorithm, simulating stochastic emergence of unwanted high frequency vibrations.

Tool chatter in machining processes is a unpredictable occurrence causing damage to the work piece and tool, continuing to be a topic of research. The regenerative effect has become a widely accepted explanation for chatter, but more advanced analytical modeling has led to the identification of other instability behavior [73]. Monitoring of these processes for purpose of instability identification, vibration detection or vibration damping is still crucial, attracting the development of wireless monitoring systems. Recently, a wireless tool tip vibration sensor was developed for milling which is capable of transmitting data via Bluetooth to a computer [70]. With the implementation of wireless systems in such a high data rate application, the node's limited energy source quickly becomes a limitation and inconvenience. However, the frequency-varying nature of milling vibrations and need for increased energy efficiency makes it an ideal candidate for adaptive sampling.

Since analytical modeling of the milling process is not the focus of this work, synthetic signals are modeled after those seen by a wireless tool tip sensor [70]. Each signal is constructed by summing three separate components, resulting in a signal containing a single chatter event.

$$y(t) = y_1(t) + y_2(t) + y_3(t) \tag{5.12}$$

The first component consists of the spindle and tooth pass frequencies, along with their first harmonic.

$$y_1(t) = \sum_{i=1}^{4} A_1(i) \cdot \sin[2\pi \cdot f_1(i)] \tag{5.13}$$

The second component includes the dominant components emerging with chatter, which rise exponentially until maximum vibration is reached.

$$y_2(t) = \sum_{i=1}^{3} A_2(t) \cdot \Theta(t - t_{start}) \cdot \sin[2\pi f_2(i) \cdot (t - t_{start}) + \theta(i)] \tag{5.14}$$

$$A_2(t) = \begin{cases} \eta(\mathrm{i}) \cdot \mathrm{e}^{-\Lambda \cdot (\mathrm{t} - \mathrm{t_{start}})} & t < t_{start} + t_{rise} \\ D_i & t \geq t_{start} + t_{rise} \end{cases} \tag{5.15}$$

$$\Theta(t - t_{start}) = \begin{cases} 1 & 0 < t - t_{start} \leq t_{chat} \\ 0 & elsewhere \end{cases} \tag{5.16}$$

Here, t_{rise} denotes the time from chatter emergence to full vibration, t_{start} represents when the vibration begins, t_{chat} defines how long the entire chatter event lasts, f is each vibration frequency, η is the initial chatter amplitude and θ is phase lag. Finally, the third term is made up of harmonics from the chatter vibration.

$$y_3(t) = \sum_{j=1}^{3} A_3(i) \cdot \Theta(t - t_{start} - t_{rise}) \cdot \sin[2\pi \cdot f_1(i)] \tag{5.17}$$

$$\Theta(t - t_{start} - t_{rise}) = \begin{cases} 1 & 0 < t - t_{start} \leq t_{start} + t_{chat} \\ 0 & elsewhere \end{cases} \tag{5.18}$$

Similarly, $\Theta(t - t_{start} - t_{rise})$ causes the chatter harmonics to emerge only once the main chatter mode has reached maximum vibration. The frequency and amplitude parameters used for signal construction are presented in Table 5.1. Each test signal contains a single chatter event with random length, rise time, and event length. Noise is added to the signal at signal to noise ratios of 20, 10, and 0 dB.

$$S(t) = y(t) + e(t) \tag{25}$$

Table 5.1. Test Signal Constants

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
f_1	63.7 Hz	127.30 Hz	190.9 Hz	254.0 Hz
f_2	560.0 Hz	900.0 Hz	960.0 Hz	-
f_3	2580 Hz	2700.0 Hz	2980.0 Hz	-
A_1	$6.5*10^{-2}$	$3.5*10^{-2}$	$8.0*10^{-3}$	$2.5*10^{-2}$
A_3	$1.2*10^{-2}$	$9.0*10^{-3}$	$1.1*10^{-2}$	-
D_1	$8.0*10^{-2}$	$1.8*10^{-2}$	$5.0*10^{-3}$	-
η	$1.0*10^{-2}$	$1.0*10^{-1}$	$8.0*10^{-3}$	-

The signals which would have been sampled at a fixed rate of 8 kHz are sampled adaptively using rates causally determined by the proposed adaptive sampling algorithm. Once complete, the sampling rate used by the algorithm is compared against the ideal case as specified in Section 5.2, which gives an error value. Energy consumption during adaptive sampling is continuously calculated using Eqns (1.1-1.3) and summed to give total energy consumption. The hardware constants for the processor and transceiver are from the Intel Xscale PXA271[74] , and those of the A/D converter are of a Maxim converter defined for multiple sampling rates [18]. The hardware constants used in the simulations are tabulated in Table 5.2.

In order to achieve a realistic simulation, delay induced by processing time is included in sub-interval length T_H. Estimated computation time is estimated using Eq. (3.13) and processor speed given in Table 5.2. Here the data set analyzed in each interval T_H is taken to be 256 data points. From Eq. (3.13), the processer at 13 MHz will complete the task in an estimated 0.2 ms. Since the sampling rate during this interval is 8 kHz, the task is completed within the time 2 samples are taken, requiring 258 total data points in T_H.

Table 5.2 Hardware Constants.

Constant	Value	Constant	Value
Γ	$1.5*10^{-9}$ *nF*	*F*	$67*10^{-12}$ *J/sym/m²*
f	13 *MHz*	*G*	$10*10^{-9}$
k	$870*10^{6}$ *MHz/V*	*r*	50 *m*
ε	0.83 *V*	C_1	.0027
B	16 *bits/sample*	C_2	.2572
b	6		

An adaptively sampled signal is illustrated in Figure 5.4, along with the original signal and corresponding spectrograms. In the right hand figures, the adjusted sampling rate is plotted in red color. Comparing spectrograms of the reconstructed and original signal, it is clear that adaptive sampling has performed successfully in that the desired signal components are preserved. The spikes in the sampling rate indicate the execution of high rate sampling, analysis, and sampling rate adjustment. The time *LSRD* between analyses is also adjusted through the AIMD algorithm as seen by difference in time between spikes.

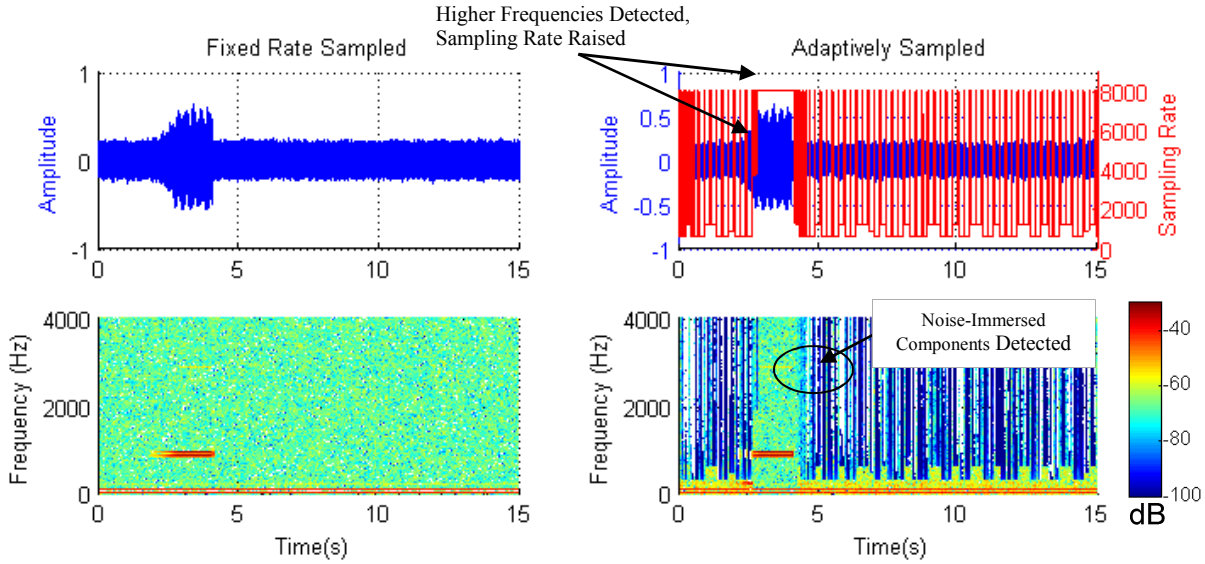


Figure 5.4. Adaptive sampling of a milling chatter signal

When high frequency events occur, the algorithm successfully detected them and the sampling rate and *LSRP* length T_L were adjusted accordingly. This is best observed by examining the sampling rate plot. Before the chatter event occurred, the high sampling rate periods (*HSRP*) were followed by relatively long low sampling rate periods (*LSRP*), with low sampling rates (0 to 1500 Hz). Once the chatter event started it was detected by the algorithm. This led to an increase in the adjusted sampling rate (7000 to 8000 Hz). Furthermore, the duration of the *LSRP* periods T_L was shortened according to the AIMD protocol. Once the chatter event has completed, the sampling conditions are again relaxed. As a result, a 68.4% decrease in energy consumption is achieved with only a 1.3% error in sampling rate adjustment according to the error metric described in Section 5.1. For three separate test signals with random chatter events, noise is added at SNR levels of 0, 10, and 20dB. Results indicate 47.8%-75.6% energy savings for the system using this signal set.

5.3.2 Case 2: Physical Activity Monitoring System

Physical activity (PA) monitoring applications can also benefit from adaptive sampling in terms of reduced power consumption of the monitoring devices. Wearable, wireless integrated measurement systems (IMSs) are desired to measure physical activity of human subjects using various sensors, since they are least restrictive. Such a system was designed [71] which acquired data from three tri-axial accelerometers, two displacement sensors and an ultraviolet photodiode. To conserve energy, an adaptive scheduling technique switches components into sleep mode between samples, drawing less current in comparison to active mode. This technique led to tremendous power savings, which can be amplified with the introduction of adaptive sampling. By lowering

average sampling rate, components are allowed to remain in sleep mode longer. To quantify energy savings, a separate energy model based on the sleep/active mode energy consumption and times spent in each [71] is used.

A set of 9 signals acquired during an activity session for a human test subject [71] are used. For the system, a fixed sampling rate of 30Hz is normally used on all 9 channels of data acquisition. The subject performs various activities. Some unknown activity is recorded (0-86 sec), followed by computer work (86-505 sec), jogging at 3mph (505-923 sec) at 0% grade, then at 5% grade (923-1341 seconds). This is seen in the signals displayed in Fig. 5.5, which are the y-axis of waist accelerometer and abdomen piezoelectric respiration sensor. Spectrograms reveal the frequency changes with time, which are taken advantage of by adaptive sampling.

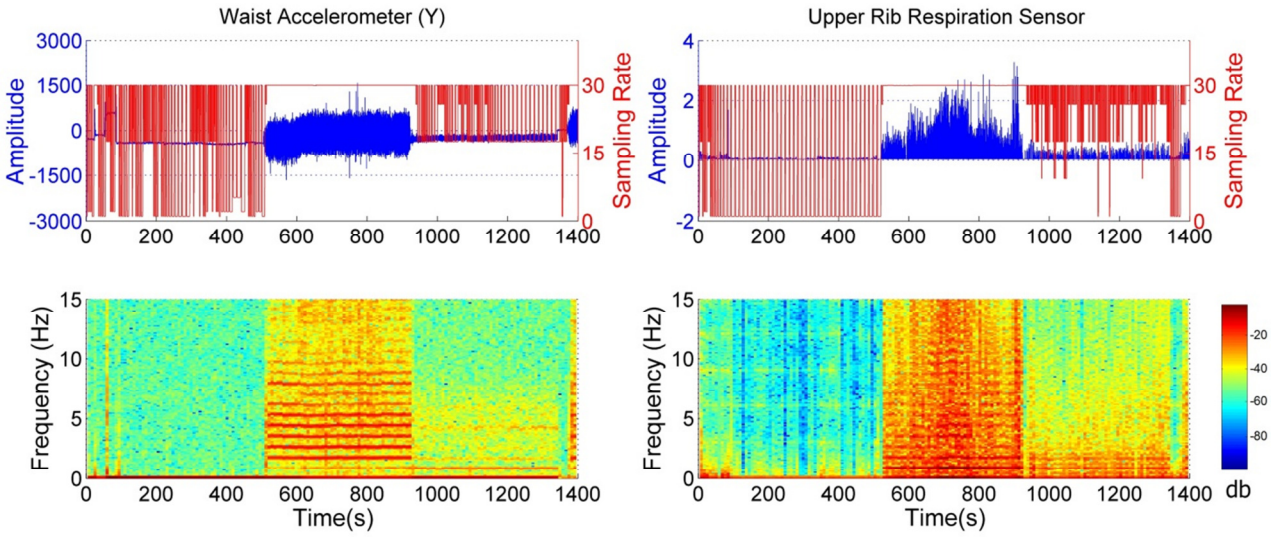


Figure 5.5. Physical Activity Signals and Adaptive Sampling Rate

As with the milling chatter event, the sampling rate is successfully varied with maximum signal frequency content. The adaptive sampling was performed on all 9

signals and results displayed in Table 5.3. From Figure 5.5 and Table 2 it is evident that the amount of data can be significantly reduced based on the changing bandwidth requirements of the sensors. For the UV sensor, a reduction of 72% was seen in comparison to the 30Hz fixed sampling rate; the high frequency part is mainly noise, allowing the sampling rate reduction. Reduction in accelerometer data is large during activities requiring lower bandwidth, seen when the subject was performing computer work or jogging at certain grades. Adaptive sampling enabled net data reduction of 33.9% for the entire acquisition event, translating to a predicted 22.7% energy savings for the integrated measurement system. This reduction is due to decreased hardware activity, and translates to a reduction in power consumption of the system from 22.6mW to 17.5mW.

Table 5.3. Adaptive Sampling for Physical Activity Monitoring

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Measurement	XYZ Waist Accelerometer			Upper Rib	Abdomen	XYZ Wrist Accelerometer			UV Sensor
Percent Error	0.25%	0.51%	0.52%	3.9%	4.3%	6.6%	3.4%	4.23%	1.6%
Data Reduction	28%	27%	32%	33%	10%	37%	27%	40%	72%
Energy Savings	**22.74%**								

CHAPTER 6

IMPLEMENTATION AND PERFORMANCE VERIFICATION

To demonstrate the usability and effectiveness of the proposed adaptive sampling algorithm, its implementation on two separate systems has been investigated. The challenges involved in implementation are investigated and solutions are verified. Although the algorithm structure is relatively simple, incorporating the algorithm and its components into an embedded system to guide sampling rate in real time presents some challenges. Namely, how to design the firmware to implement the scheme, as well as designing a custom filtering structure to efficiently compute the sampling rate selection portion of the algorithm.

The first system serves as a proof of concept, demonstrating successful performance on a system based around an 8-bit MCU and Bluetooth transceiver. Through the implementation of the adaptive sampling algorithm, the node is able to automatically adjust sampling rate according to the input signal, subsequently reducing energy expenditure. The second system demonstrates a more powerful implementation built around a DSP core, leveraging Zigbee communications. The Zigbee transceiver is much more efficient and is more flexible in terms of configuration, allowing further reduction of energy consumption through algorithm implementation. The increased computational power allows for more levels of decomposition in sampling rate selection portion, and higher sampling rates can be achieved. Additionally, a filtering structure has been designed for the DSP core to enable continuous sampling rate computation.

6.1 MCU implementation with Bluetooth Data Transmission

A Bluetooth based wireless sensor node platform has been designed and prototyped to experimentally investigate the feasibility and performance of the developed adaptive sampling algorithm. The wireless sensor node platform is based on the Freescale 8-bit microcontroller evaluation board (MC9S08LC60) that is serially interfaced to a generic Class 2 Bluetooth module. The A/D converter onboard the microcontroller samples the target signal and transmits it, via the Bluetooth transmitter to a laptop computer. The schematic and circuits of the experimental setup are illustrated in Figure 6.1.

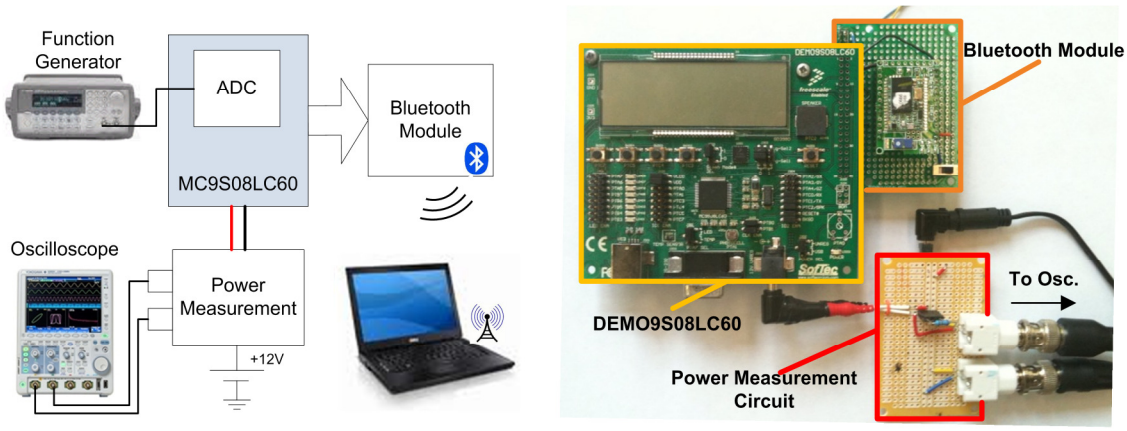


Figure 6.1. Experimental setup 1 for adaptive sampling algorithm evaluation

6.1.1 Firmware Organization

The program structure for implementing the adaptive sampling algorithm on the wireless sensor node is illustrated by means of pseudo-code in Figure 6.2. The ADC sampling is performed within an interrupt routine and the main program performs the sampling rate selection calculations and AIMD.

At system initialization, the signal is sampled at the high sampling rate SR_{High}. When enough points have been acquired, the analysis as previously described is performed in

the main program to select the sampling rate. As soon as the calculations are complete, the low sampling rate period *LSRP* begins, after which the new sampling rate SR_{adj} is used for duration T_L. Data is transmitted after each sample is taken in both periods. It should be noted that at the end of each high sampling rate period (*HSRP*) the new sampling rate SR_{adj} is transmitted via the Bluetooth to the laptop. This information is necessary for correct reconstruction of the signal on the receiver side.

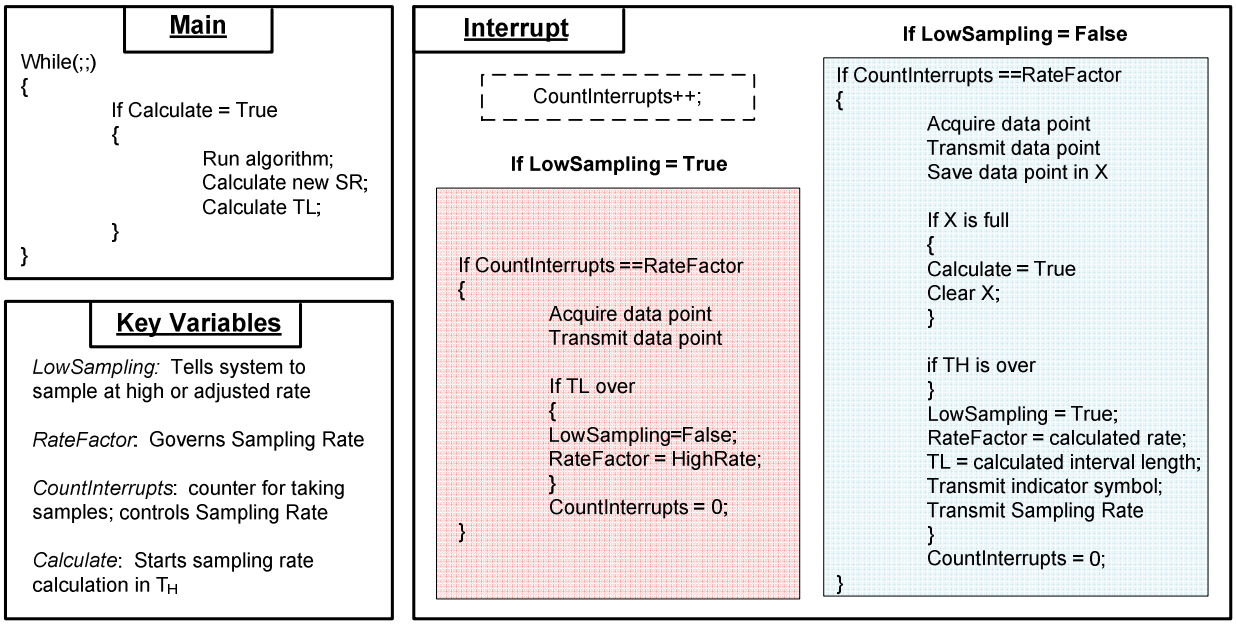


Figure 6.2. Implementation of the adaptive sampling scheme described by C Pseudo-code

6.1.2 Adaptive sampling on the Wireless Sensor Node

Three capabilities have been demonstrated experimentally, namely: 1) Lower sampling rates lead to a decrease in overall power consumption, 2) The proposed algorithm running on a WSN can identify the correct sampling rates for a target signal, and 3) The node is

capable of dynamically adjusting the sampling rates in a way that allows for reliable signal reconstruction.

The first issue has been addressed by measuring the net power consumption of the sensor node operating at different sampling rates. The results are illustrated in Figure 6.3, showing about a 90% difference in net node power consumption for different sampling rates between 10 Hz and 8 kHz; baseline power consumption excluded. The baseline power consumption is relatively large, which can be improved by selecting more efficient hardware and leveraging sleep modes. This is shown in the second system.

Baseline Power Consumption

Device, Status	Power Usage
Demo Board on, BlueTooth off	439mW
Demo Board on, BlueTooth on	792mW

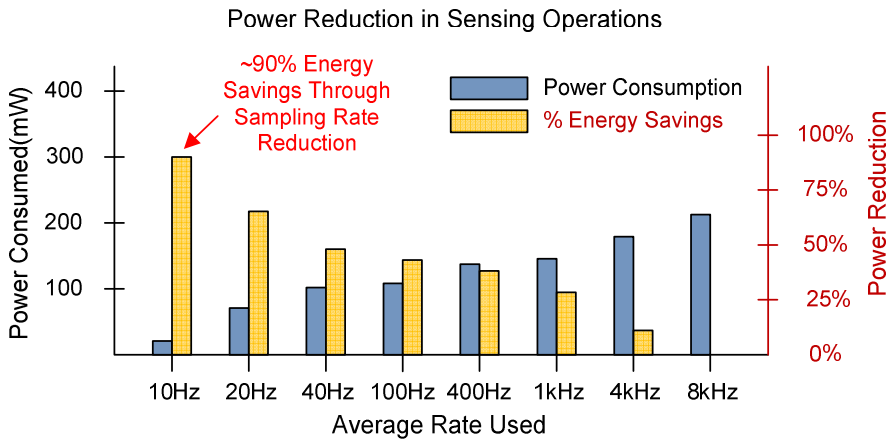


Figure 6.3. System baseline power consumption (Left), Energy savings by lowering sampling rate (Right)

As discussed previously, the sampling rate selection algorithm processes the high pass side first, while traversing the wavelet packet tree. If the coefficient values in the high pass side are zero, the algorithm proceeds by decomposing the low pass band. As a result, it requires different amount of filtering operations to evaluate certain frequency bands in comparison to others. Thus detection of different frequencies involves different processing latencies. Figure 6.4 illustrates the results for measured processing latencies

for the sampling rate selection algorithm to reach each frequency sub-band in the two levels (using a filter length of 4 and signal length of 16). For an input signal spread over the entire bandwidth, the wavelet decomposition takes 2.81 ms. In the worst case scenario, the analysis takes 4.59 ms. With 2 levels of decomposition the advantage of the selective decomposition over using a full wavelet packet transform is evident, as computing all bands and thresholding using a 2 level WPT is clocked at 7.8 ms.

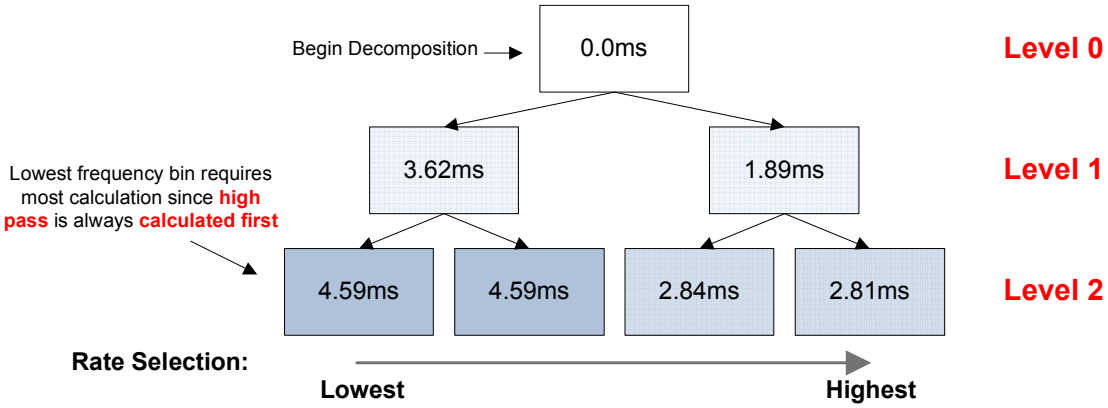


Figure 6.4. Execution time measured for the SDA to end at each band for 2 levels of decomposition, length 4 filters, length 16 input.

The node's ability to compute the adjusted sampling rate has been investigated. The sampling rate SR_{High} is selected to be 333Hz. With this rate, the 4 frequency bins contain the bandpass regions 0-42Hz, 42-83Hz, 83-125Hz, and 125-166Hz. The adjusted sampling rates selected corresponding to these bins, as calculated are 83Hz, 166Hz, 250Hz and 333Hz. The function generator is used to generate sine waves, and these test signals are fed to the ADC of the sensor node as illustrated in Figure 6.1. The sensor node adaptively samples these signals and transmits them to the laptop. The received data is analyzed to see if the sensor node chose the correct frequency bin and corresponding sampling rate.

Figure 6.5 presents the results for the test signals, each of which is a sine wave in the range of each frequency bin. It is seen that for each input, the correct frequency bin and sampling rate is selected by the algorithm. Examining the received samples (left hand plots) reveals the workings of the adaptive sampling algorithm. The periodic stretched-out sections indicate high sampling rate periods (*HSRP*). Each *HSRP* section is followed by 0 and 255 values, which look like a spike, followed by the sampling rate value. This is part of a protocol event that indicates sampling rate change. The *LSRP* of length T_L follows this protocol event. From the received samples, the sampling rates are extracted, and reconstruction is performed. The reconstructed test signals are illustrated on the right side.

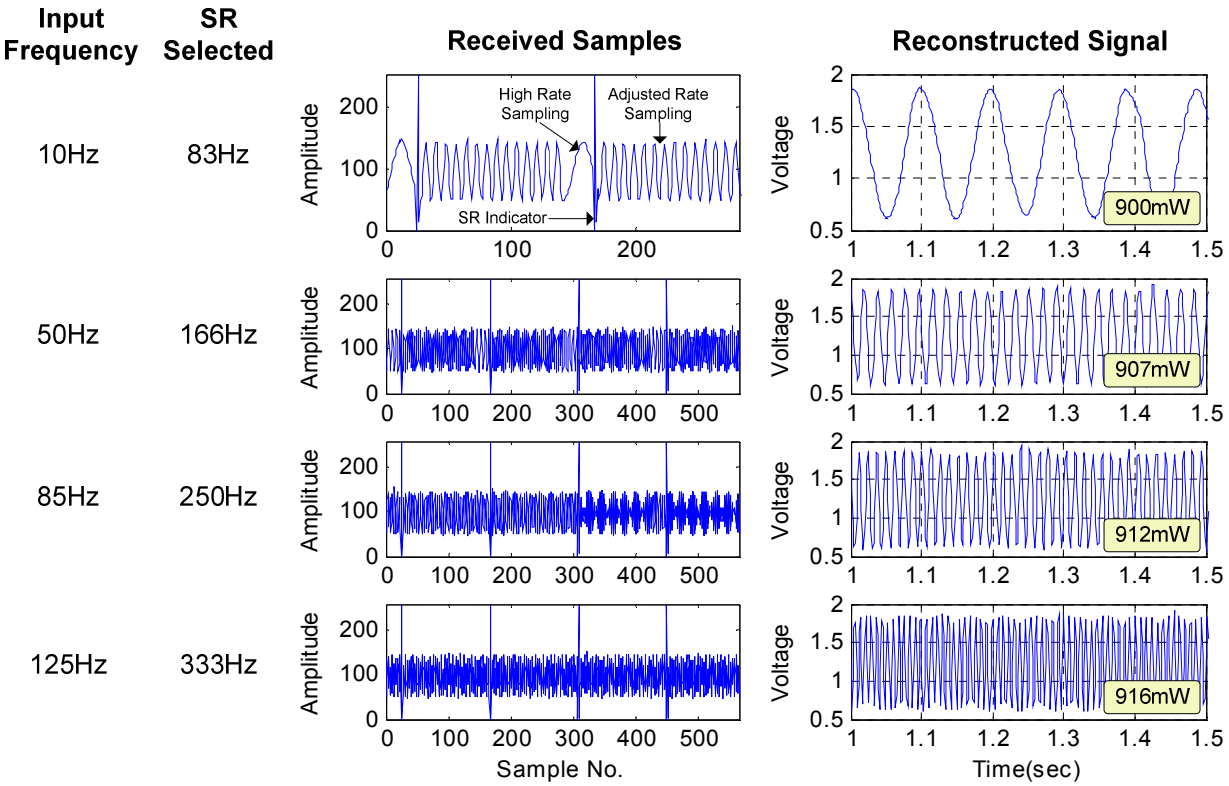


Figure 6.5. Automatic sampling rate adjustment by the sensor node for different input sine waves

6.2 DSP Implementation with Zigbee Communications

Though the 8-bit MCU implementation serves as a proof of concept, its performance is limited due to the 8-bit operations, low clock speed and limited memory. Many sensor nodes have incorporated more powerful processing units which facilitate greater capabilities in localized processing and faster communications, as well as increasing data load capability. A custom wireless sensor node based around a Texas Instruments C5500 Ultra-low power DSP has been prototyped to incorporate adaptive sampling and investigate opportunities presented by a more powerful system.

The DSP based system is shown in Figure 6.6. The Texas Instruments 32-bit C5510DSK evaluation board is seen top left, which houses the DSP chip and its peripherals. Bottom left is the low power ADC and its circuitry, which is interfaced through a multichannel buffered serial port (McBSP). The Zigbee transmitter is seen on the right hand side, along with three power resistors with which to measure energy consumption of the three components. The Zigbee module is interfaced also through an McBSP with a software SCI protocol.

The 32-bit, 200Mhz DSP core is capable of handling substantially greater computational load, allowing for higher data acquisition rates and also levels of decomposition in sampling rate selection; which results in finer rate selection. Digital signal processors are optimized for filtering which consists mainly of shift-add operations. To take full advantage of this, a filtering architecture is proposed which continuously computes sampling rate during the high rate sampling period (HSRP).

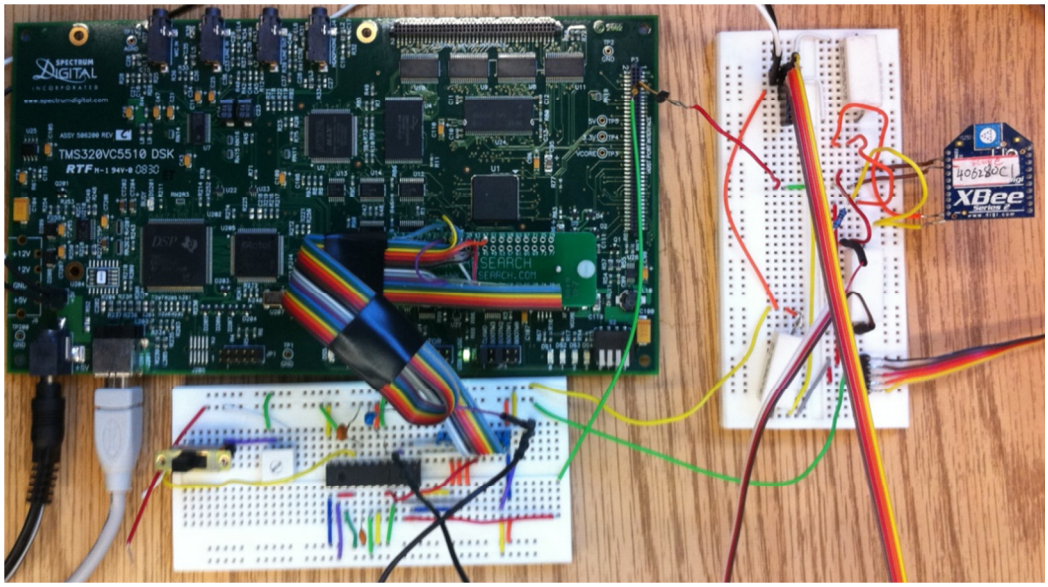


Figure 6.6. Experimental setup 2, DSP with Xbee transmission

Figure 6.7 demonstrates this improvement. Originally, as shown in filtering scheme 1 (Top), the *HSRP* is completed and then the data block generated during this period is analyzed, to calculate sampling rate. As formulated in Chapter 3, the wavelet packet algorithm leverages a recursive decomposition to determine frequency range. This algorithm uses decimated filtering, meaning that each successive level of decomposition has roughly half the amount of samples; 2 samples are needed at one level to generate 1 sample in the next. Therefore, for a J level decomposition, every 2^J samples acquired lead to generation of 1 coefficient. The new scheme, filtering scheme 2 (bottom) uses this by performing the filtering operations after every block of 2^J coefficients. This allows the algorithm computation to be spread out over the *HSRP*, dramatically reducing the latency in sampling rate adjustment following the end of the interval. For example, clocking the computation for a 3 level decomposition with length 16 filters, translating to blocks of size 8, the computation time is only 68μs.

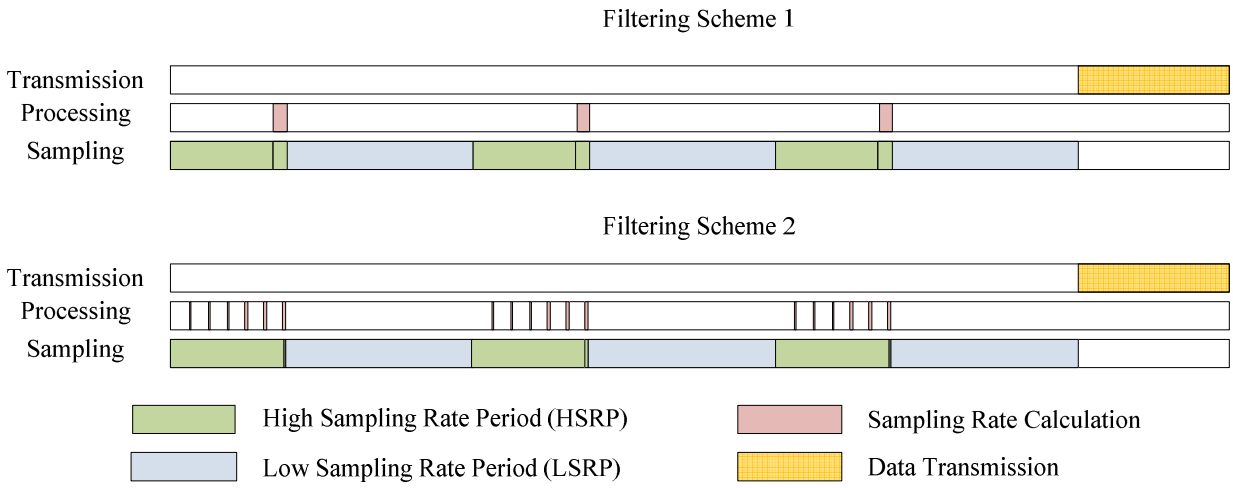


Figure 6.7. Execution timing using the two filtering schemes

Implementing a filter structure for this scheme is somewhat complex, since in filtering the previous *L-1* samples are needed as well as the current sample block, where *L* is the length of the filter. Also, the algorithm may take different paths down the wavelet tree as explained in Chapter 3. One possibility is to store all computed bands, allocating a memory block for each band. This requires excessive memory, seeing that for even a 3 level decomposition, $2^3+2^2+2^1+1=15$ separate blocks are needed.

To make it more efficient in terms of memory usage, a multiple buffered filtering scheme shown in Figure 6.8 is proposed. In this scheme, whenever the next block of samples is acquired, they are placed into an allocated memory block B_{00} and the previous samples are shifted to the left into the buffer. Then for each successive level, the current path on the wavelet tree is compared to the previous path. If it is different from the previous one, the buffer must also be updated before the new coefficients are computed. This is because the samples stored here are actually from a different band; only the highest path will be kept in order to compute sampling rate. If the path is the same as the previous one, the old coefficients are shifted left into the buffer and the new coefficients are

computed. In doing this, only 2 bands need memory allocation at each level of decomposition; whose sizes have also been decreased. For example, a 3 level scheme with a *HSRP* of 128 points would have required 516 memory units. With this multiple buffered filtering structure, using a length 16 filter, the memory requirement is reduced to 166 memory units; a reduction of 67%.

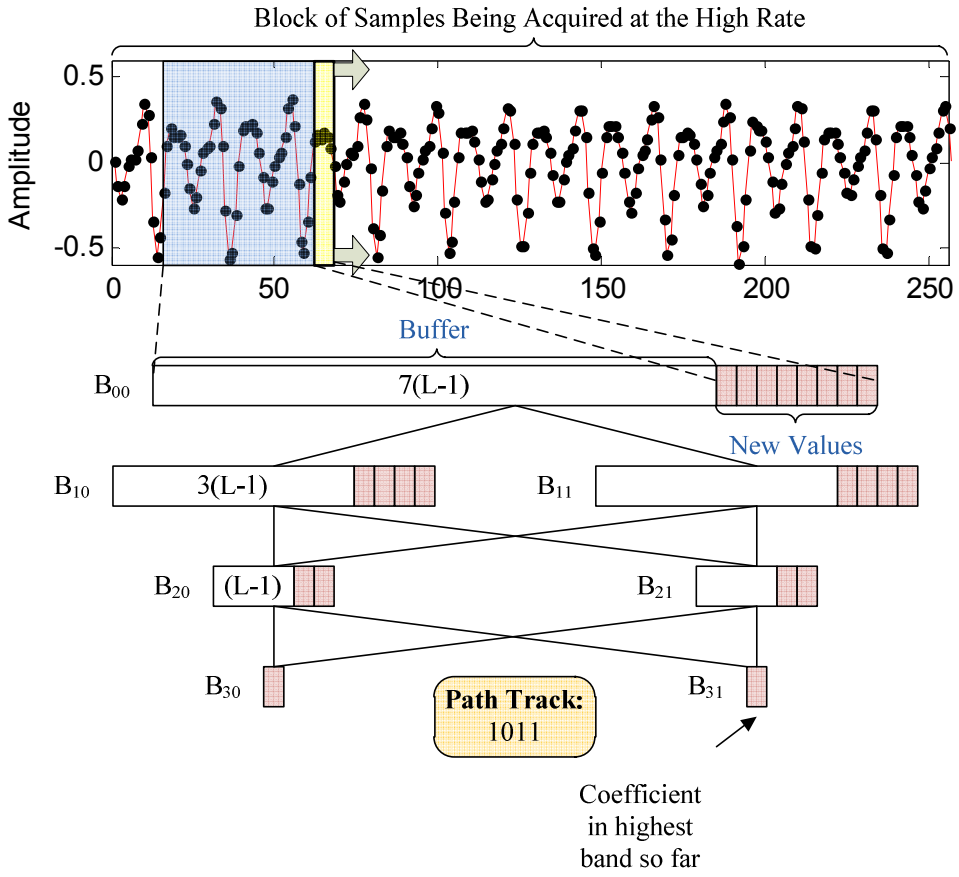


Figure 6.8. Proposed multiple buffered filtering scheme

The energy savings via adaptive sampling for this system can be viewed in Figure 6.9, which shows the energy consumptions for different average sampling rates used during a 0.5s length acquisition period. For signals which may vary between 0-1KHz for example, a fixed sampling rate would be chosen of 2KHz, consuming about 0.715J of energy. By

incorporating adaptive sampling, if the average sampling rate can be reduced to 250Hz for example, where it would reach 2KHz temporarily to capture high frequency events, this consumption could be reduced to about 0.11J; saving about 85% in terms of energy consumption. The components in this system consume less baseline power than in the previous system, and this 85% represents total energy expenditure reduction.

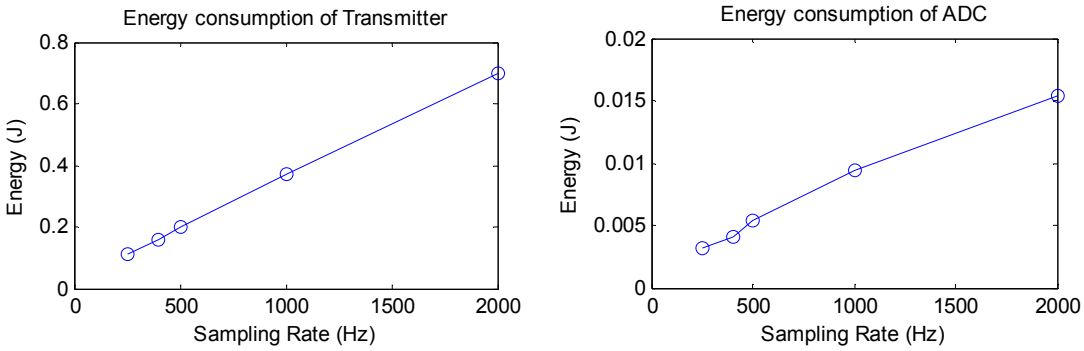


Figure 6.9. Energy consumption at different rates, 0.5 second acquisition

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Summary

In this thesis, a technique for adaptive data acquisition on a wireless node has been formulated as an alternative to the traditional, fixed-rate sampling method. This technique is used to dynamically adjust the sampling rate by means of short bursts of high rate sampling and analysis. This analysis is outlined in detail in Chapter 3, which tracks the signal spectral properties utilizing an effective filtering scheme. This includes selectively computing coefficients of wavelet packet decomposition and utilizing the outcome as the basis for selecting an appropriate sampling rate.

A signal reconstruction procedure has been derived for the special case of non-uniform sampling in Chapter 4, which allows the signal to be retrieved from the adaptively acquired samples. This linear algebra based procedure utilizes a generalized interpolation approach to solve the problem of reconstruction artifacts at sampling rate change times. Additionally, a post-analysis has been formulated in Chapter 5 to quantify error in sampling rate adjustment and signal loss, which isolates error in rate adjustment from that of signal reconstruction.

The developed algorithm has been evaluated through simulations for a variety of test signals in Chapter 5, to evaluate its validity. An energy model has also been incorporated to estimate changes in sensor node energy consumption. Results indicate data reduction enabled by adaptive sampling, which leads to significant reduction in energy

consumption for trial both cases, while maintaining low error. Performance has also been verified through implementation on two test beds in Chapter 6: one based on an 8-Bit microcontroller and another based on a high speed 32-bit DSP system. All necessary implementation issues in algorithm embedment such as firmware organization, filter architecture design and evaluation are addressed and presented.

7.2 Intellectual Contributions

In this work, several contributions have been made in the field of wireless monitoring, specifically adaptive sampling. Firstly, a new adaptive sampling technique has been proposed which leverages bursts of high rate sampling to overcome previous limitations in sampling rate adjustment. In previous works, it was proposed that sampling rate be continually adjusted based on signal frequency content. This principle has theoretical basis, but in practice is only possible to implement for small, gradual change in frequency content, which is almost never the case. The reason for this is that once sampling rate is lowered, higher frequency events cannot be detected. By performing these bursts of high rate sampling, the entire signal bandwidth is probed in order to identify the 'instantaneous' signal bandwidth. This enables higher frequencies to be detected, even after the sampling rate has been lowered. This results in a net reduction in samples acquired and energy consumption, whilst maintaining sampling rate necessary to preserve signal quality.

In addition to this simple concept, an improved theoretical framework has been presented to select sampling rate based on signal frequency content. Previously proposed techniques leverage a Fourier transform [41], or wavelet transform to do this [45]. The advantage of time-frequency analysis over standard Fourier transform is known

extensively in the literature; however the wavelet transform does not investigate the high frequency range. A wavelet packet transform enables greater resolution and therefore more sampling rates to choose from, however induces greater computational complexity. This work has presented a selective decomposition algorithm to calculate sampling rate, using the structure of the wavelet packet decomposition. This gives the higher resolution while adding significantly less computational complexity and therefore execution time, as shown in Figure 3.7.

A contribution has also been made in the area of signal reconstruction. The literature is extensive in terms of interpolative techniques for both fixed rate and non-uniformly sampled signals. However, the case of a signal sampled at switching rates, where there are sampling rate change points, has not been previously investigated. The reconstruction procedure derived in this work provides a theoretical solution for this problem, and is shown to be optimal over previously available techniques. This procedure allows signals sampled using the proposed algorithm to be recovered.

The literature is also lacking in investigation of the implementation issues associated with incorporating adaptive sampling into real embedded systems. This is most likely due to the fact that it is well established that reducing samples taken will reduce energy consumption. Thus, most effort is spent in production of theory for adaptive sampling and compressed sensing. This thesis has delved into the associated challenges, suggesting framework and solutions for implementation issues, specifically to optimize speed and minimize the algorithm's computational footprint on system resources.

7.3 Future Work

This work lays the groundwork for a new adaptive sampling technique, from which there are several possible research directions. The first of which is to improve the criteria for triggering analyses in the adaptive sampling. The proposed technique uses timed analyses, which can be fixed based on knowledge of the application, or also adjusted adaptively using an AIMD method. There is opportunity to improve on this problem, which includes the class of problems in node activation at the network level as well as analysis triggering at the node level. In short, improved methods are needed to suggest when to update the system model in terms of sampling rate. The system should be self-aware of whether or not it is accurately sensing the independent signal, or spatial field correctly. Tools developed in the fields of statistics and mathematics should be incorporated to accomplish this goal. Among these include change detection, or event detection algorithms developed for complex signals such as audio or spatio-temporal field monitoring, which can be used to trigger the system to reconfigure its parameters, including sampling times.

In cases such as field monitoring, a network level approach in sampling is necessary to estimate the two dimensional field. However, in cases such as manufacturing monitoring, it is the individual signals that are of interest; vibration, temperature fluctuation of a machine, for example. In these cases, a node level adaptive sampling scheme such as the one proposed is ideal as they preserve the quality of each signal; it is not a field estimation problem. However, effects of node level adaptive sampling on the network level performance must be investigated. With each node adjusting the sampling rate according to each input signal, the network will experience varying levels of

communication and data load. Handling of the changing requirements must be addressed in terms of bandwidth management.

At the network level, some of these signals can be spatially correlated. In cases where there is correlation between two sensor readings, nodes can share information about what is happening and come to some consensus about what sampling rates to use. If one sensor has missed some signal information due to lowering of sampling rate, it may be possible to improve the resultant signal using information from the other sensors.

7.4 Broader Impact

The presented work on node-level reconfigurable sampling for information acquisition is applicable to many real world sensing systems which can benefit from extended sensor node lifetime. Example applications include manufacturing, structural health, surveillance, and cyber physical systems such as smart buildings and smart grid. These applications require monitoring of systems which exhibit signals of varying activity and frequency content, where reconfigurable sampling can be used to conserve sensor node energy whilst retaining signal fidelity.

BIBLIOGRAPHY

[1] T. Becker et al., "Autonomous sensor nodes for aircraft structural health monitoring," IEEE Sensors Journal, vol. 9, no. 11, pp. 1589-1595, 2009.

[2] N. Elvin, N. Lajnef, and A. Elvin, "Feasibility of structural monitoring with vibration powered sensors," Smart Materials and Structures, vol. 15, pp. 977-986, June 2006.

[3] R. Teti, K. Jemielniak, G. O'Donnel, and D. Dornfeld, "Advanced monitoring of machining operations," Annals of the CIRP, vol. 59, no. 2, pp. 717-739, 2010.

[4] T. He et al., "Energy-efficient surveillance system using wireless sensor networks," Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys), pp. 270-283, Boston, MA, June, 2004.

[5] L. Schwiebert, S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," Mobile Computing and Networking, pp. 151-165, 2001.

[6] G. Chen, S. Hanson, D. Blaauw, D. Sylvester, "Circuit Design Advances for Wireless Sensing Applications," Proceedings of the IEEE, vol. 78, no. 11, pp. 1808-1827, 2010.

[7] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, 2002.

[8] J. Pan, Y. Hou, L. Cai, Y. Shi, S. Shen, "Topology Control for Wireless Sensor Networks," Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pp. 298-299, San Diego, CA, September 2003.

[9] S. Mukhopadhyay, C. Schurgers, D. Panigrahi, and S. Dey, "Model-Based Techniques for Data Reliability in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, vol. 8, no. 4, pp. 528-543, 2009.

[10] D. Bruneo, A. Puliafito, and M. Scarpa, "Dependability Evaluation of Wireless Sensor Networks: Redundancy and Topological Aspects," Sensors, 2010 IEEE, pp. 1827-1831, November 2010.

[11] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc sensor network infrastructure for emergency medical care," Proceedings of the International Workshop on Wearable Implantable Body Sensor Networks, June 2004.

[12] E. Sazonov, H. Li, D. Curry, P. Pillay, "Self-Powered Sensors for Monitoring of Highway Bridges," IEEE Sensors Journal, vol. 9, no. 11, November 2009.

[13] Y. Yu, G. Qiao, and J. Ou, "Self-Powered Wireless Corrosion Monitoring Sensors and Networks," IEEE Sensors Journal, vol. 10, no. 12, December 2010.

[14] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," IEEE Transactions on Signal Processing, pp. 40-50, March 2002.

[15] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," Ad Hoc Networks, vol. 7, no. 3, pp. 537-568, 2009.

[16] C. Schurgers, V. Ragunathan, and M. Srivastava, "Power Management for Energy-Aware Communication Systems," ACM Transactions on Embedded Computing Systems, vol. 2, no. 3, pp. 431-447, 2003.

[17] V. Gutnik and A. Chandrakasan, “Embedded Power Supply for Low-Power DSP,” IEEE Transactions VLSI System, vol. 12, pp. 425-435, 1997.
[18] Application Notes Maxim Corporation, “12-Bit Sampling analog to digital converter conserves power,” http://www.maxim-ic.com/appnotes.cfm/an_pk/43, 1998.
[19] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, “Energy management in wireless sensor networks with energy-hungry sensors,” IEEE Instrumentation & Measurement Magazine, vol. 12, no. 2, pp. 16-23, 2009.
[20] B. Zhai, D. Blaauw, D. Sylvester, K. Flautner, “The Limits of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 11, November 2005.
[21] C. Yeh, Z. Fan, and R. Gao, “Energy-Aware Data Acquisition in Wireless Sensor Networks,” Proceedings of the Instrumentation and Measurement Technology Conference (IMTC), May 2007.
[22] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, “Networking Issues in Wireless Sensor Networks,” Journal of Parallel and Distributed Computing, vol. 64, no. 7, pp. 799-814, July 2004.
[23] C. Schurgers, V. Tsiatsis, M. B. Srivastava, “ STEM: Topology Management for Energy Efficient Sensor Networks,” IEEE Aerospace Conference, Big Sky, MT, March 2002.
[24] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Sensor Networks,” Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2004.
[25] M. C. Vuran, and I. F. Akyildiz, “Spatial Correlation-based Collaborative Medium Access Control in Wireless Sensor Networks,” IEEE/ACM Transactions on Networking, vol. 14, no. 2, pp. 316-329, April 2006.
[26] A. Wegener, “Compression of Medical Sensor Data,” IEEE Signal Processing Magazine, vol. 27, no. 4, pp. 125-130, June 2010.
[27] Y. Lu, J. E. Michaels, “Feature Extraction and Sensor Fusion for Ultrasonic Structural Health Monitoring Under Changing Environmental Conditions,” IEEE Sensors Journal, vol. 9, no. 11, September 2009.
[28] R. Willet, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," Proceedings of the3rd International Symposium on Information Processing in Sensor Networks (IPSN), pp. 124- 133, Berkeley, CA, April 2004.
[29] J. Kho, A. Rogers, and N. Jennings, “Decentralized control of adaptive sampling in wireless sensor networks,” ACM Transactions on Sensor Networks, vol. 5, no. 3, article 19, pp. 1-35, May 2009.
[30] X. Liu, Q. Wang, W. He, M. Caccamo, and L. Sha, “Optimal Real-Time Sampling Rate Assignment for Wireless Sensor Networks,” ACM Transactions on Sensor Networks, vol. 2, no. 2, pp. 263-295, May 2006.
[31] J. Wang, Y. Liu, and S. K. Das, “Energy-Efficient Data Gathering in Wireless Sensor Networks with Asynchronous Sampling,” ACM Transactions on Sensor Networks, vol. 6, no. 3, article 22, pp. 1-37, June 2010.
[32] S. Bandyopadhyay, Q. Tian, and E. J. Coyle, “Spatio-Temporal Sampling Rates and Energy Efficiency in Wireless Sensor Networks,” IEEE/ACM Transactions on Networking, vol. 13, no. 6, pp. 1339-1352, December 2005.

[33] P. Padhy, R. Dash, K. Martinez, and N. Jennings, “A Utility-Based Adaptive Sensing and Multihop Communication Protocol for Wireless Sensor Networks,” ACM Transactions on Sensor Networks, vol. 6, No. 3, article 27, pp. 1-39, June 2010.
[34] A. Jain, and E. Y. Chang, “ Adaptive sampling for sensor networks,” Proceeedings of the 1st international Workshop on Data Management For Sensor Networks (DMSN), vol. 72, pp. 10-16, Toronto, Canada, August 2004.
[35] J. Haupt, W.U. Bajwa, M. Rabbat, and R. Nowak, “Compressed Sensing for Networked Data,” IEEE Signal Processing Magazine, vol. 25, no.2, pp. 92-101, March 2008.
[36] J. Ploennigs, V. Vasyutynskyy, and K. Dabitzsch, “Comparative Study of Energy-Efficient Sampling Approaches for Wireless Control Networks,” IEEE Transactions on Industrial Informatics, vol. 6, no. 3, pp. 416-424, June 2010.
[37] Y. Borgne, S. Santini, and G. Bontempi, “Adaptive Model Selection for time series prediction in wireless sensor networks,” Signal Processing, vol. 87, no. 12, pp. 3010-3020, December 2007.
[38] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, “Model-Driven Data Acquisition in Sensor Networks,” Proceedings of the 30th International Conference on Very Large Databases (VLDB), vol. 30, pp. 588-599, Toronto, Canada, September 2004.
[39] W. R. Dieter, S. Datta, and W. Key Kai, "Power Reduction by Varying Sampling Rate," Proceedings of the 2005 International Symposium on Low Power Electronics and Design (ISLPED), pp. 227- 232, August 2005.
[40] M. Unser, “Sampling-50 years after Shannon,” Proceedings of the IEEE, vol. 88, no. 4, pp. 569-587, April 2000.
[41] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, “An Adaptive sampling algorithm for effective energy management in wireless sensor networks with energy hungry sensors,” IEEE Transactions on Instrumentation and Measurement, vol. 59, no. 2, pp. 335-344, 2010.
[42] J. Dunlop, and N. C. Changkakati, "Transmission of speech by adaptive sampling," Radio and Electronic Engineer , vol.50, no.3, pp.127-132, March 1980
[43] R. Rieger, and J.T. Taylor, “An Adaptive Sampling System for Sensor Nodes in Body Area Networks,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 17, no. 2, pp. 183-189, April 2009.
[44] T. Kurp, R. Gao, and S. Sah, “An adaptive sampling scheme for improved energy utilization in wireless sensor networks,” Proceedings of the International Instrumentation and Measurement Technology Conference (I2MTC), pp. 93-98, Austin, TX, May, 2010.
[45] P. Hall and S. Penev, “Wavelet based estimation with multiple sampling rates,” The Annals of Statistics, vol. 32, no. 5, pp. 1933-1956, 2004.
[46] D. Healy, and D. J. Brady, “Compression at the Physical Interface,” IEEE Signal Processing Magazine, vol. 25, no. 2 pp. 67-71, March 2008.
[47] E. J. Candès, and M. B. Wakin, "An Introduction to Compressive Sampling," IEEE Signal Processing Magazine, vol.25, no.2, pp.21-30, March 2008

[48] E. J. Candès, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," IEEE Transactions on Information Theory, vol.52, no.2, pp. 489- 509, February 2006.
[49] E. J. Candès, M.B. Wakin, and S.P. Boyd, "Enhancing Sparsity by Reweighted l1 Minimization," Journal of Fourier Analysis and Applications, vol. 14, pp. 877-905, December 2008.
[50] E. J. Candès, and J. Romberg, "Sparsity and Incoherence in Compressive Sampling," Inverse Problems, vol. 23, no. 3, pp. 969-985, April 2007.
[51] D. Donoho, "Compressed Sensing," IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006.
[52] M.A. Herman, and T. Strohmer, "High Resolution Radar via Compressed Sensing," IEEE Transactions on Signal Processing, vol. 57, no. 6, pp. 2275-2284, June 2009.
[53] P. Boufounos, and M.S. Asif, "Compressive Sampling for Streaming Signals with Sparse Frequency Content," The 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1-6, Princeton, NJ, March 2010.
[54] G. Peyre, "Best Basis Compressed Sensing", IEEE Transaction Signal Processing, vol. 58, no. 5, pp. 2613-2622, May 2010.
[55] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data Acquisition through Joint Compressive Sensing and Principle Component Analysis," Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-6, Honolulu, HI, December 2009.
[56] S. Mitra, "Digital Signal Processing: A Computer Based Approach, McGraw-Hill, New York," 3rd Ed, 2006.
[57] F. Elfouly, M. Mahmoud, M. Dessouky, and S. Deyab, "Comparison between haar and daubechies wavelet transformations on FPGA technology," International Journal of Computer, Information, and Systems Science and Engineering, vol. 2, no. 1, pp. 37-42, 2008.
[58] C. Chen, J. Lee, "Design of Quadrature Mirror Filters with Linear Phase in the Frequency Domain," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 39. no. 9, pp. 593-605, 1993.
[59] D. Donoho and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," Biometrica, vol. 81, no. 3, pp. 425-455, 1994.
[60] H. Jasem, Z. Zukarnain, M. Othman, S. Subramaniam, "Fairness of the TCP-based new AIMD congestion control algorithm", Journal of Theoretical and Applied Information Technology, vol. 5, pp. 568-576, 2009.
[61] S. Cai, Y. Liu, and W. Gong, "Analysis of an AIMD based collision avoidance protocol in wireless data networks," Proceedings of the 42nd IEEE Conference on Decision and Control, vol.1, pp. 104-109, Maui, HI, December, 2003.
[62] S. Kumar, M. Gupta, V. Srivastav, K. Agarwal, "On the efficiency and fairness of congestion control algorithms", Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pp. 405-407, 2007.
[63] P. Thevenaz, T. Blu, M. Unser, "Interpolation Revisited," IEEE Transactions on Medical Imaging, pp. 739-758, vol. 19, no. 7, July 2000.
[64] M. Liou, "Spline Fit Made Easy," IEEE Transactions on Computers, vol. C-25, no. 5, pp. 522, May 1976.

[65] B. Migeon, and P. Marché, "In Vitro 3D Reconstruction of Long Bones Using B-Scan Image Processing," Medical and Biological Engineering and Computing, p. 369-372, vol. 35, no. 4, 1997.
[66] A. Aldroubi, M. Unser, and M. Eden, "Cardinal Spline Filters: Stability and Convergence to the Ideal Sinc Interpolator," Signal Processing, vol. 28, no. 2, August 1992.
[67] M. Unser, "Cardinal Exponential Splines: Part II- Think Analog, Act Digital," IEEE Transactions on Signal Processing, pp. 1439-1449, vol. 53, no. 4, 2005.
[68] T. Blu, P. Thévenaz, and M. Unser, "MOMS: Maximal-Order Interpolation of Minimal Support," IEEE Transactions of Image Processing, vol. 10, no. 7, July 2001.
[69] S. Yunhui, R. Qiuqi, "Continuous Wavelet Transforms", Proceedings of the International Conference on Signal Processing (ICSP), vol.1, pp. 207-210, 2005.
[70] C. Suprock, B.K. Fussell, R.B. Jerard, and R.Z. Hassan, "A low cost wireless tool tip vibration sensor for milling," Proceedings of the 3rd ASME International Manufacturing Science and Engineering Conference (MSEC), Paper No. 72492, Evanston, IL, October, 2008.
[71] S. Liu, R. Gao, and P. Freedson, "Design of a Wearable Multi-Sensor System for Physical Activity Assessment," Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 254-259, Montreal, Canada, July 2010.
[72] C. Chen, J. Lee, "Design of Quadrature Mirror Filters with Linear Phase in the Frequency Domain," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 39, no. 9, pp. 593-605, 1992.
[73] T. Insperger, G. Stepan, P. Bayly, and B. Mann, "Multiple chatter frequencies in milling processes," Journal of Sound and Vibration, vol. 262, no. 2, pp. 333-345, 2003.
[74] C. Yeh, "Dynamic reconfiguration techniques for wireless sensor networks," Master's Thesis, University of Massachusetts, Amherst, 2008.