

AN ENERGY AUDIT OF AUTOMOTIVE VIBRATION SOURCES FOR ENERGY HARVESTING AND APPLIED COMPUTATION IN WIRELESS SENSOR NETWORKS

Andrew Pullin

UC Berkeley

Berkeley, CA, USA

ABSTRACT

Sensor networks powered entirely from harvested energy are becoming increasingly feasible with developments in low power radios and harvesting devices. The a priori knowledge of the energy availability in applications is important in the choice of sensors and network organization, and limits the maximum duty cycle of the radio. These devices need to be designed such that their resonance frequencies match those present in the system to which they are attached to maximize efficiency and power output. This project describes an effort to audit and analyze the energy availability for an oscillating type of piezoelectric or electromechanical harvesting devices on in a potential installation on a consumer automobile.

A data logging system using wireless sensor nodes is designed and installed in an automobile to measure the vibrations present in the vehicles structures. The vibration data is taken along side velocity and location parameters, and a correlative analysis is performed to characterize the conditions under which certain vibrations are encountered.

This audit is followed by details of related work in wireless sensor network (WSN) software. The computational capacity of individual WSN nodes is investigated through an implementation of the Fast Fourier Transform algorithm on a sensor network node. Using network nodes for significant computation can affect network and operating system design. Implications of timing, energy use, and software design are discussed.

INTRODUCTION

The development of WSNs is hinged on the supporting technologies, such as novel low-power radio designs, micro-batteries, energy harvesters, and low-power protocols. While the application space is very broad, WSN's are starting to be deployed in industrial settings using standardized communication stacks, such as ISA100 and Wireless HART. Wirelessly enabled "smart meters" for power metering are being installed in residences in California. Wireless instrumentation of structures for HVAC and general building control has also been demonstrated[1].

Harvesting devices based on a number of methods and materials have been proposed and shown to be viable power sources. This space is demands novel techniques for harvester design, for instances where battery packs would be un-maintainable or too bulky. Thermally powered biological implants using dispenser-printed thermoelectrics have been proposed and demonstrated [2]. Mechanical oscillating harvesters using piezoelectric materials [3] and electromagnetic schemes have also been heavily investigated. The individual nodes in a wireless sensor network are often referred to as "motes" [4]; the design of an individual mote can vary significantly, and two classes of complexity are exemplified in the latter half of this project.

ORGANIZATION

This report is divided into two major sections: the first section details the energy audit on an automobile and the analysis of the resulting data. Section 1.1 introduces this component of the report and gives background on the project. Section 1.2 describes the design of the experiment and the system and Section 1.3 describes the installation of the sensors into the target automobile. The resulting dataset and the analysis thereof is described in Sections 1.4 and 1.5, and concluding remarks on the energy audit project are given in Section 1.6.

The second focuses on an investigation of the computational capacity of WSN nodes, wherein the Fast Fourier Transform is implemented and performance tested. Section 2.1 introduces this half of the project and relates it to the preceding Section 1. Section 2.2 gives a background on the algorithm and its application. Section 2.3 compares the hardware architectures on which the code is tested. Section 2.4 describes the code implementation and speed results. Section 2.5 gives concluding remarks for the applied computation section of the report.

SECTION 1: ENERGY AUDIT

1.1 OUTLINE

Several types of energy harvesting power plants for micro devices have been proposed in the past. Emerging technologies are exemplified by micro-scale thermoelectric generators and oscillatory benders, both piezoelectric and electromagnetic. The very limited power generation of micro-scale harvesting devices imposes significant restrictions on the design and activity level of sensor network nodes they are powering.

Use of wireless sensor networks in cars for instrumentation has previously proposed and investigated in a limited context, with attempts to implement a sensor in a harsh environment powered entirely by energy harvesting[5].

MEMS harvesters span a large design space, with many possible materials and geometries, which are chosen to cater to a specific resonant frequency and magnitude of oscillation. Efforts are taken in the design of automobiles by addition of damping materials to reduce mechanical vibrations, which is contrary to the goals of the energy harvesting design. The required power to run a well designed sensor node can be very low, and so even the lowest level of vibration is of interest for a design target.

1.2 SYSTEM DESIGN

Vibrations were measured using miniature accelerometers attached to the vehicle. The data was acquired with off-the-shelf sensor units from Microstrain, Inc. The Microstrain sensors use MEMS type accelerometers: two



Figure 1: Schematic of organization of sensors and coordination computer.

ADXL210 chips are mounted orthogonally in a single chip package inside the node. This overcomes the limitation of single planar chip 3-axis accelerometers, which have poor bandwidth in the Z direction, perpendicular to the plane of the device package. This sensor configuration provides equal sensitivity and bandwidth in all directions. All nodes are equipped with 802.15.4 compatible radios and batteries for wireless installation.

The 802.15.4 radios provide packet-based communication in a close proximity range, usually less than a few meters. The radios are optimized for low power consumption, and are common in WSN applications. The radios can operate on one of 16 channels, arbitrary divisions of the 2.4 Ghz band, as to not interfere with each other. It is not uncommon for a WSN to hop over multiple channels to provide a more reliable connection [6], however the deployed sensor network operated on only a single channel due to limitations in the provided Software Development Kit (SDK) from the sensor manufacturer.

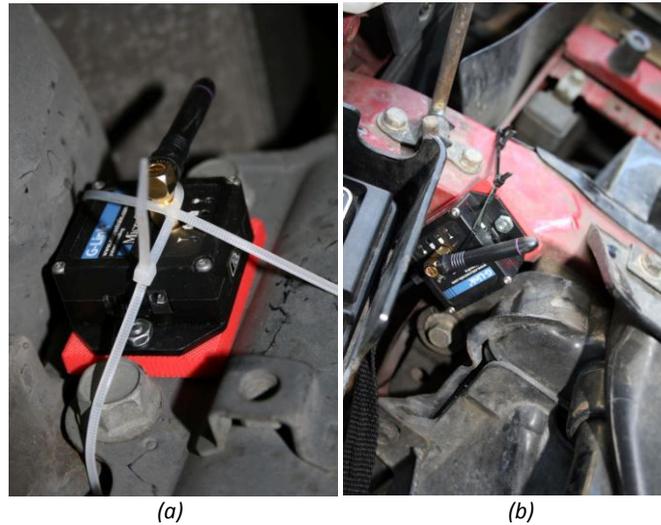
From tables in[7], we estimated that the accelerations we would find would be less than 10 G in any direction. The tables also indicate the most vibrations of significant amplitude will occur below 1 Khz. The selected Microstrain sensor has a quoted 10 mG accuracy and 9 mG resolution[8]. The dynamic range of this sensor was determined to be sufficient sensitivity for the frequency and amplitude regime we are investigating for design points for energy harvesting devices.

The system is coordinated by a notebook computer using a radio base station. All nodes communicate via a single hop to the base station. A consumer GPS unit is connected via Bluetooth to the coordinating computer to record vehicle position and velocity. Software was written in the Python programming language to automate the collection of data: Microstrain provides details of a simple serial protocol between the host computer and the USB base station to send commands to compatible nodes. The coordinator program regularly triggers data collection on all the nodes simultaneously, after which all data is then downloaded from the nodes, and stored with information recorded by the GPS unit. The packet radio communications was not documented in the SDK, so all coordination was done through the base station and the network operates as a star-topology network.

1.3 SENSOR INSTALLATION

The sensors were installed on body sheet metal, on un-sprung suspension components, the interior of the passenger compartment, and directly on the intake manifold of the engine. All experiments are performed with a front engine, rear wheel drive car driven on public streets and highways of varying quality. The sensors were

magnetically mounted to the sheet metal components, or rigidly clamped in place on plastic interior pieces. The



*Figure 2: Photograph of installed accelerometer sensors.
a) On suspension components, b) engine bay sheet metal structure*

magnetic mount was independently tested with a shaker table, and the mount did not show any damping, drift, or loosening. The orientation of the nodes when installed was arbitrary and not aligned with the road surface; the sensors have equal accuracy in all directions, and so there should be no sensitivity to orientation. During data collection, sensor orientation is determined in-situ by sampling accelerations when stationary to determine the direction of gravity.

The locations chosen for sensor installation were either areas of expected high vibration or potential installation points for a useful wireless sensor node powered by an energy harvesting generator. Some of these candidate locations are described below.

A number of sheet metal structures were instrumented: the separator between the engine bay and passenger compartment, or the “firewall”, the passenger foot well, near the Engine Control Unit, the bracing and skin of driver’s side door, and the floor of the cargo space in the rear of the vehicle. These represent applications for low-rate sensors, such as temperature sensing for internal environmental control or for routing and repeating nodes. Other interior components were also instrumented for measurement, such as a plastic panel of the door interior and center of the steering wheel.

Large shocks and vibrations were suspected to be found in the suspension components, particularly those which are suspended by the spring and shock absorber unit. Rigid components, such as the lower A-arm and anterior

side of the wheel hub were also instrumented. This location would be a natural choice for a routing node for an in-tire sensor; the proximity to the wheel would allow the in-tire node to operate with a minimum radio power.

1.4 RESULTS

The resulting dataset is many vibration spectra on the 3 axes of each sensor, each with an associated velocity, vehicle location, and sensor location string. The vehicle location is not directly used in any computation or filtering; public roads are used, not a closed track, so road quality and path repeatability is unreliable.

A presentation interface, shown in Figure 3, was also written for browsing the datasets and for presenting the results of the project in a qualitative way. A web application based on the Google Maps API was used as a viewer for the datasets, where each sample was placed from the GPS data. The markers are color coded to the

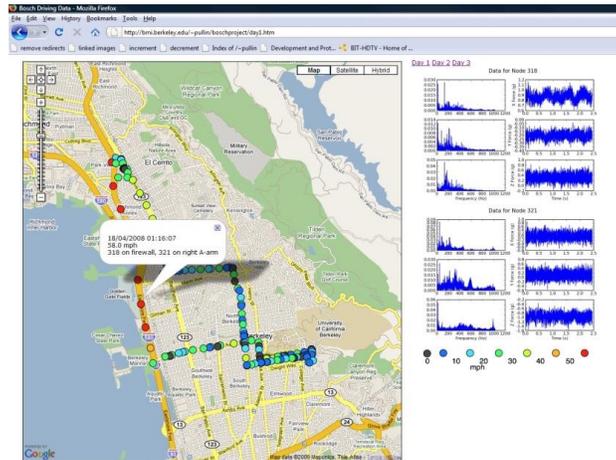


Figure 3: Data browser show samples located on a map, data and spectra on the right for all active sensors.

vehicle's average velocity for that dataset. Clicking on any point shows the recorded acceleration data and the spectra for the three separate axes, as well as a balloon containing the placement of the sensors and the time of the experiment. The map interface can be viewed online at <http://bmi.berkeley.edu/~pullin/boschproject/>

1.5 ANALYSIS

For a given installed sensor location, a spectrogram of the vibration spectrum can be generated, as in Figure 4. The data is organized in the vertical direction in the order in which it was recorded, from the start of the collection run to the end. It is more useful to visualize the data with each row ordered by increasing velocity. Regions of reliable activity can be identified by a vertical band that does not drift significantly in frequency, as can be seen in

Figure 4. This is only one examples of the large dataset that was collected; a summary of some specific peaks, locations, and conditions is given in Table 1.

To quantify how commonly a vibration of a given magnitude and frequency occurs, plots like Figure 5 are generated. These plots show, as a function of frequency, the percentage of records above a threshold value of acceleration amplitude. These figures show that for all locations, the most significant vibrations are found at frequencies less than 300 Hz; this matches the range of resonant frequencies reported in prototype MEMS energy harvesting devices.

1.7 CONCLUSION

The design of a vibration sensing wireless instrumentation system has been described, and the results from it's installation in a consumer automobile are reported an analyzed. Although only one automobile was used for the test due to the limited scope of the project, this same instrumentation could be applied to a variety of vehicles or mechanical structures to produce a similar audit for available vibrations for harvesting. Table 1 gives a summary of conditions and locations of consistently observed peaks in the recorded spectra which fall in an application regime for oscillatory energy harvesting devices, and could be considered as a design guide for targeting a harvesting system.

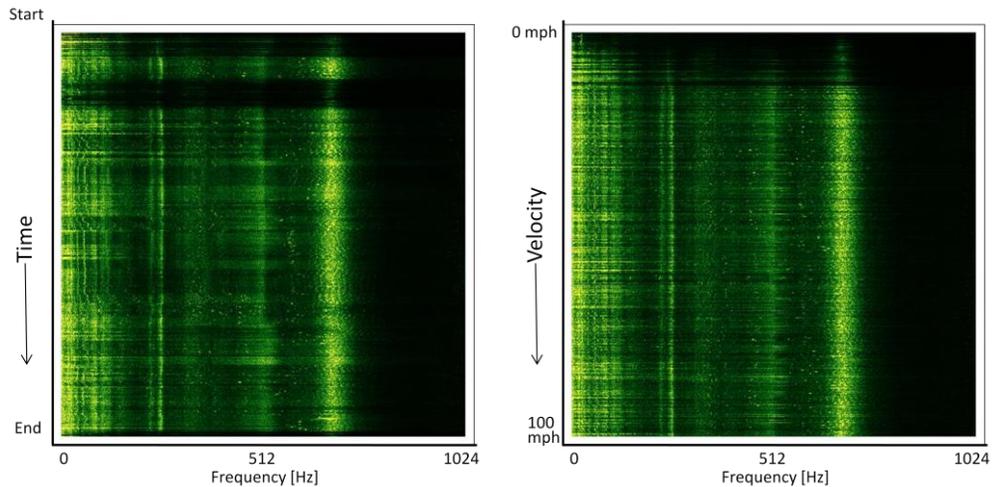


Figure 4: Spectrogram of vibration data. Left: time ordered, right: velocity ordered. Sensor node installed on lower suspension A-arm.

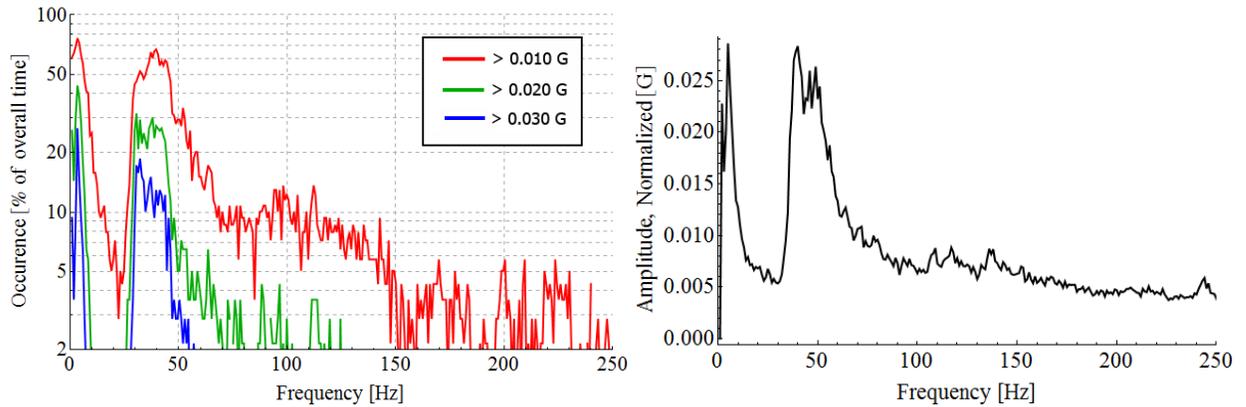


Figure 5: Top: Occurrence of vibrations as a function of frequency for several thresholds. Bottom: A representative root-mean-square value of spectrum amplitude across the same frequency range. Sensor node was installed on engine bay sheet metal.

Center Freq. [Hz]	RMS Amplitude [mG]	Sensor Location	Condition, Speed
200	200	firewall	highway cruise, 65 mph
685	150	firewall	highway cruise, 60 mph
75	200	wheel hub	highway, high load, 70 mph
71	80	rigid suspension	street driving, 25 mph
200	250	sheet metal	street driving, 25 mph
15-20	50	firewall	idling, 0 mph

Table 1: Summary table of significant vibrations peaks from the instrumentation of an automobile, identified as good design points for oscillating mechanical harvesters.

SECTION 2: COMPUTATION ON MOTES

2.1 OUTLINE

The microprocessors commonly found on wireless sensor nodes are fully programmable digital computers with robust storage resources, and thusly are capable of running complex algorithms and calculations and executing code beyond simple coordination of the node's sensors and radio. Running signal analyses like this locally on the node can reduce the amount of data to be transmitted over the radio, reducing the power usage of the node and the traffic

load on the network. An investigation of the feasibility and performance of a typical signal processing algorithm was the focus of the second half of this project: the goal was to implement the Fast Fourier Transform on multiple sensor node platforms and compare the speed, power usage, benefits, and implications of using the algorithm in a WSN context.

2.2 BACKGROUND

The Fourier Transform (FT) is one of the most common tools in signal processing applications. The FT is commonly used to transform a signal in the time domain signal into the frequency domain, which provides the spectral power density as a function of frequency. The FT is defined for both a continuous time function, and discrete-time data:

$$\begin{aligned}
 \text{(a) CTFT : } F(\Omega) &= \int_{-\infty}^{\infty} f(t) e^{-i \Omega t} \\
 \text{(b) DFT : } X[k] &= \sum_{n=0}^{\infty} x[n] e^{-i \frac{2\pi}{N} k n} \\
 \text{(c) DTFT : } X(\omega) &= \sum_{n=-\infty}^{\infty} x[n] e^{-i \omega n}
 \end{aligned}$$

Figure 6: Equations for (a) Continuous Time Fourier Transform, (b) Discrete Fourier Transform, (c) Discrete Time Fourier Transform

The datasets on the motes come from sampling the Analog to Digital Converter (ADC) at regular time intervals. This gives a sequence of data values, so the Discrete Fourier Transform (DFT) is used to calculate the spectrum. Note that the DFT itself does not carry any time or frequency information; this is inferred from knowing the sampling frequency and the number of samples. The transformed dataset, of length N , is paired with matching frequency components, $f = \left\{ \frac{2048}{N}, 2 \frac{2048}{N}, \dots, 2048 \right\} Hz$.

Computationally, the time complexity of the standard DFT would be $O(N^2)$, as a sum over the length N dataset is done for each of N points in the dataset. As originally described by Cooley and Tukey[9], there exists an algorithm for computation of the DFT of a set of data that requires substantially fewer operations than the full summation in the DFT of the set by taking advantage of recursion. The Cooley-Tukey algorithm, known commonly as a Fast Fourier Transform (FFT), requires only $O(N \log[N])$ operations to achieve the same result, and is therefore the standard method implemented for FFT calculations on microprocessor systems.

There are many details that are elided here, as FFTs are a significant area of study in signal processing. The standard Cooley-Tukey decomposition relies on a divide-and-conquer style algorithm that recursively looks at subset of the data that are $\frac{1}{2}$ of the length of the previous data, also known as a radix-2 algorithm. This requires that the length of any input dataset is exactly equal to a power of 2. Variations on this algorithm can achieve shorter compute times, generally at the cost of code complexity and size, memory footprint, and dataset size limitations.[10]

2.3 HARDWARE ARCHITECTURES

Two hardware platforms are used to implement and test the FFT algorithm. The platforms were chosen to represent two distinct design targets and levels of computational ability in an embedded device: the T-mote sky has a Texas Instruments MSP430 microcontroller optimized for cost and low power operation, and has a very limited clock speed, whereas the SUN Microsystems SunSPOT uses a much faster Atmel ARM9 microprocessor designed for high throughput. Details of both platforms of given in table 2.

Modern microcomputers have support for floating-point math operations in hardware, as a Floating Point Unit (FPU) designed into the CPU. The FPU allows the chips to perform math operations on IEEE standard single or double precision floating point numbers as single instructions which execute with comparable speed to integer

	T-Mote Sky	SunSPOT
Architecture	16-bit MSP430	32-bit ARM9
Max CPU freq.	8 Mhz	180 Mhz
RAM	10 KB SRAM, on-chip	512 KB SRAM, off-chip
ROM	48 KB flash, on-chip 1 Mbit flash, off-chip	4 MB flash, off-chip
Operating System	Contiki OS	Squawk JVM

Table 2 : Comparison of the hardware platforms examined

arithmetic. Most embedded processors, such as the ARM9 of the SunSPOT and the MSP430 chip on the T-mote Sky do not include an FPU, as they are design for low cost, transistor count, and complexity. To perform the same math operations on IEEE format floating point numbers, software libraries are needed that can decode the mantissa and exponent, and carry out the math correctly. The algorithm can alternatively implemented using fixed-point math operations. The FFT is implemented with both methods on both platforms, and comparisons are made in the following sections.

2.4 IMPLEMENTATION

The floating-point implementations use special functions to handle the exponent-and-mantissa storage of IEEE floating point numbers; all mathematical operators, including special functions like exponentiation, sin, and cos are all linked in from software floating-point libraries. Since none of the embedded CPU's on the motes have FPU's, these codes execute slowly, as even simple arithmetic requires a function call with many instructions.

The common alternative to floating point operations is to use a fixed-point arithmetic scheme, where integer data types are specially treated to represent fractional numbers. Instead of storing a mantissa and exponent, a single machine-sized word (16 or 32 bits) is considered to be separated into sections representing positive and negative powers-of-2. In this way, the two sections represent integer components and fractional components, and



Figure 7: Layout of IEEE 754 floating-point representation and arbitrary fixed point representation.

machine instructions for integer math will give correct results, as long as overflows and underflows are considered and managed by the implementer.

Using fixed-point math also elides the need for functions from the software math library to be linked into the final compiled code. The SunSPOT platform is not affected by this, as the entire runtime libraries are included

	f_{cpu}	Floating Point	Fixed Point
Tmote Sky	4.1 Mhz	11.9 sec	222.3 ms
SunSpot	180 Mhz	264 ms	N/A

Table 4: Time for 512-point, 16-bit FFT on various processors, for floating and fixed point implementations

N	Floating Point [ms]	Fixed Point [ms]
16	217.8	3.682
32	527.7	8.636
64	1188	19.93
128	2597	45.03
256	5680	100.7
512	11850	222.3

Table 5: Expanded timing information for the FFT implementations on the Tmote Sky. N is number of samples in the dataset being processed.

on the SunSPOT . The T-mote sky has very limited flash storage, so inclusion of just a few math functions takes a significant amount of on-chip flash.

Table 4 gives the result of testing the speed of implementation for a 512-point dataset. The compiled size of the various implementations is given in Table 3; this is the amount of ROM space that the function would occupy on the target sensor node. For platforms with limited RAM, the code will be executed from flash, which has longer read times than RAM, and overall executing is slowed. Systems with sufficient RAM space can copy the functions from flash into RAM to increase execution speed, either at boot time or during run-time. In this case, the Tmote Sky is executing the code from flash ROM.

	Floating Point	Fixed Point
Tmote Sky	11K	3K
SunSpot	256K	256K

Table 3: Compiled size of FFT algorithm for various processors, for floating and fixed point implementations.

2.5 APPLICATIONS

The investigation of the Fourier Transform naturally follows the preceding work in vibration sensing, as vibrations are often analyzed in the frequency domain rather than the time domain. There are several benefits to be realized in moving this calculation onto the mote.

An interesting and useful result of the Fourier Transform is that for a real-valued signal, the output spectrum will be exactly symmetric about the y-axis, Nyquist-Shannon Sampling Theorem[11]. Knowing a priori that our system will only record real valued data from the analog-to-digital converter onboard, we know that the output of the transform will be symmetric. If our sensing application is concerned with the frequency make of the signal, as would be the case for many condition-based monitoring installations of WSN in an industrial context, then the FFT can be performed on the mote, at the source of the data, thereby reducing it to a symmetric power spectrum, and the output dataset is half the length of the full set of samples recorded by the ADC, as shown in figure 8.

This immediately halves the number of data bytes to be transferred from the radio for sense-and-send scenarios, thusly reducing the overall energy use in the radio. Further reductions can be done, to summarize the data as prominent peaks, maximal values, and average noise. With these extensions, a node in a machinery monitoring application could greatly reduce its energy requirements by doing a full suite of frequency domain analysis onboard, and only transmitting summary statistics, or alerts and events upon condition changes.

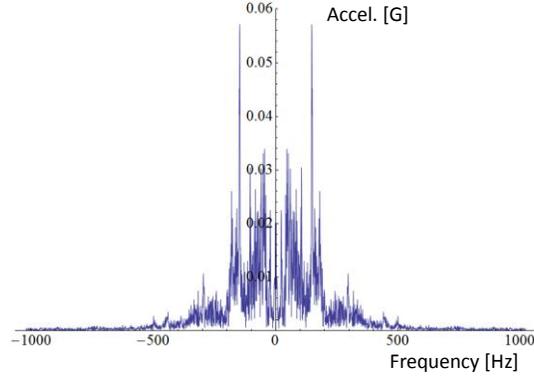


Figure 8: Example of a symmetric spectrum from vibration dataset

Using the datasheet values for the chips on the Tmote Sky platform, the TI MSP420F1611 microcontroller[12] and the Chipcon CC2420 radio[13], the energy used to compute the FFT and send the data can be estimated and compared to the energy use for transmitting the complete dataset without any signal processing. Figure 10 shows the time and energy comparisons for these two scenarios as a function of the dataset length. Note that in practice, the dataset length is restricted to certain values, depending on the implementation of the FFT; in this case, it must be a power of 2 due to the radix-2 implementation. The steps that are visible in the curves are due to the overhead of a packet header, due to the maximum packet size of 128 Bytes. The derivation of the energy and time expenditures is given in figure 9. Operating values taking from datasheets are given in table 6.

<p>To send B_{data} bytes of raw data:</p> $N_{pkts} = \text{Ceiling}[B_{data}/MDPU]$ $B_{total} = B_{data} + B_{head} * N_{pkts}$ $E_{total} = E_{byte} * B_{total} + E_{start}$ $T_{total} = (B_{total}/BPS) + t_{start}$	<p>To FFT, send $B_{data}/2$ bytes of raw data:</p> $N_{pkts} = \text{Ceiling}[(B_{data}/2)/MDPU]$ $B_{total} = (B_{data}/2) + B_{head} * N_{pkts}$ $E_{total} = E_{byte} * B_{total} + E_{start} + t_{FFT} * P_{cpu}$ $T_{total} = B_{total}/BPS + t_{start} + t_{FFT}$
--	---

Figure 9: Comparison of time and energy usages for the two presented schemes.

2.6 CONCLUSIONS

This implementation of the FFT demonstrated that on-mote analysis is possible with existing hardware, and uses less time and energy in total for the task of spectral analysis, as shown in figure 10. It is also shown that consideration must be given to underlying hardware and software features of the platform when designing a sensor node to achieve maximum performance.

Parameter	Value
MDPU	112 Bytes
B_{head}	16 Bytes
B_{data}	$2 * (\# \text{ samples})$
E_{start}	4.33 μJ
E_{byte}	0.573 μJ
t_{start}	1.49 ms
P_{cpu}	6 mW
t_{FFT}	<i>see Table 5</i>

Table 6: Values for hardware used in this comparison

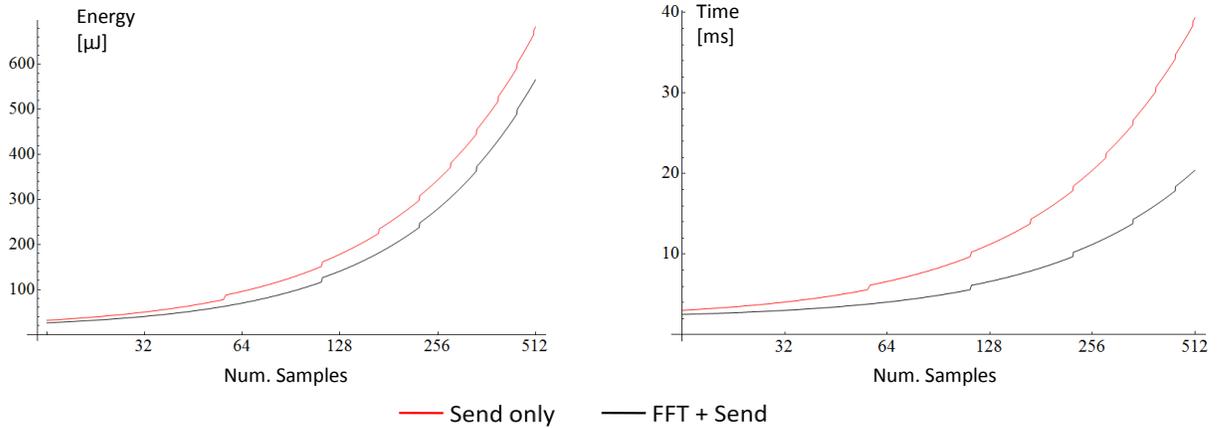


Figure 10: Comparison of time and energy usages for the two presented schemes.

As sensor node platforms are miniature and increasingly micro-integrated, custom tailoring for specific applications will take place. With the microcontroller, radio, sensors, and harvesters all manufactured together using wafer-processing technology, highly specialized CPUs or accessory cores can be added to the die, providing hardware acceleration of this type of algorithm. Such design considerations could drive the total CPU on-time to an absolute minimum while also reaping the benefits of local analysis for very little energy expended.

ACKNOWLEDGMENTS

Thanks are due to the following individuals and organizations for support of these projects: Paul Wright, Kris Pister, Dan Steingart, Nathan Ota, Mike Koplou, The Robert Bosch Corporation, and the California Energy Commission.

References

1. Ota, N.K., *The application of wireless sensor networks to residential energy efficiency and demand response*. 2007.
2. Koplw, M., et al. *Thick film thermoelectric energy harvesting systems for biomedical applications*. in *2008 5th International Summer School and Symposium on Medical Devices and Biosensors*. 2008. Hong Kong, China: Ieee.
3. Dutoit, N.E., B.L. Wardle, and S.-G. Kim, *DESIGN CONSIDERATIONS FOR MEMS-SCALE PIEZOELECTRIC MECHANICAL VIBRATION ENERGY HARVESTERS*. *Integrated Ferroelectrics: An International Journal*, 2005. **71**(1): p. 121 - 160.
4. Brett, W., *Smart Dust: Communicating with a Cubic-Millimeter Computer*, L. Matt, L. Brian, and S.J.P. Kristofer, Editors. 2001. p. 44-51.
5. Yuen Hui, C., et al. *PicoCube: a 1 cm³ sensor node powered by harvested energy*. in *2008 45th ACM/IEEE Design Automation Conference*. 2008. Anaheim, CA,: Ieee.
6. Dust Networks, *Technical Overview of Time Synchronized Mesh Protocol* Available from: http://www.dustnetworks.com/cms/sites/default/files/TSMP_Whitepaper.pdf.
7. Roundy, S.J., *Energy Scavenging for Wireless Sensor Nodes with a Focus on Vibration to Electricity Conversion*. 2003.
8. *Microstrain G-Link Datasheet*. Available from: http://microstrain.com/product_datasheets/G-LINK_2.4_GHz_Datasheet_Rev_10.00.pdf.
9. Cooley, J.W. and J.W. Tukey, *AN ALGORITHM FOR MACHINE CALCULATION OF COMPLEX FOURIER SERIES*. *Mathematics of Computation*, 1965. **19**(90): p. 297-&.
10. Gold, L.R.R.a.B., *Theory and Applications of Digital Signal Processing*. 1975: Prentice-Hall, New York.
11. Shannon, C.E., *COMMUNICATION IN THE PRESENCE OF NOISE*. *Proceedings of the Institute of Radio Engineers*, 1949. **37**(1): p. 10-21.
12. *Texas Instruments MSP430F1611 Datasheet*. Available from: <http://focus.ti.com/lit/ds/symlink/msp430f1611.pdf>.
13. *Chipcon/Texas Instruments CC2420 Datasheet*. Available from: <http://focus.ti.com/lit/ds/symlink/cc2420.pdf>.

