Mobility in Wireless Sensor Networks

by

Ankur Mehta

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kristofer S. J. Pister, Chair
Professor Michel Maharbiz
Professor Alexandre Bayen

Fall 2012

The dissertation of Ankur Mehta, titled Mobility in Wireless Sensor Networks, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

University of California, Berkeley

Mobility in Wireless Sensor Networks

Abstract

Mobility in Wireless Sensor Networks

by

Ankur Mehta

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kristofer S. J. Pister, Chair

Abstract

For you

And everyone else

Contents

List of Figures

List of Tables

Acknowledgments

I want to thank my advisor for advising me.

Chapter 1

Introduction

Everyday life is getting ever more intertwined with the Internet; the densely connected set of nodes on the global network, along with the data contained therein, is often referred to as "the cloud". With the extensive proliferation of laptops, cell phones, tablets, and similar electronic devices, humans can now easily access the cloud at a moment's notice. But that interaction is still largely directed and one-dimensional – the only behavior is for digital data to be transmitted at the command of a user.

The next major revolution will be to augment the cloud with more immersive agents – pervasive sensors will autonomously gather and process data from their enviromnent, while distributed actuators can directly interface with the physical world. These devices will form the "swarm at the edge of the cloud" [32]. Many science fiction stories have been built on the foundation of swarms independently acting cyber-physical systems [38, 11]; the goal of this research is to pave the path forward to that very technology.

1.1 Wireless sensor networks

One of the great visions of wireless sensor network (WSN) research is the idea of a ubiquitous and seamless interface between the physical and online worlds. Described by such phrases as "Smart Dust" and "the Internet of Things," WSN research aims to saturate our environment with small, cheap, multi-functional nodes that can sense, process, and communicate. WSNs are already employed in several commercial applications; custom networks are used to monitor industrial machinery, household power consumption, and urban congestion. However, WSN research is still an emerging field – part of the design process still includes determining the appropriate application space that WSNs can potentially fill.

At its most basic, a WSN system need not include sensors at all – a wireless network can merely provide a communications infrastructure to existing sensors or standalone electronics. Allowing a number of disparate devices and appliances to communicate with each other or a centralized controller already enhances the nature of interaction with electronics. Adding dedicated sensing to the nodes in a wireless network then significantly expands on the abilities delivered by a WSN. Sensors such as temperature, occupancy, or power monitors enable for example new home automation or energy efficiency paradigms.

1.2 Mobility

The utility and functionality of a sensor network can be greatly enhanced by introducing mobility to some or all of the nodes. There are many ways such actuation can benefit the

system.

At a system level, mobile elements can be used to simply deploy a conventional stationary network, potentially reducing installation costs and thus increasing the application space. Then, if some of the nodes themselves can move, they could rearrange themselves after deployment to increase wireless connectivity, forming a robust network that can respond dynamically to changing environmental conditions.

Mobile sensor nodes can also be used to optimze utilization in resource-constrained systems. A small number of expensive or otherwise scare sensors can be placed on mobile robots, moving throughout the environment to maintain full coverage in a time-multipliexed manner. Alternately, a homogeneous system can use mobile nodes to concentrate sensing around areas of interest, gaining dynamic range where necessary while still maintaining optimal usage of individual nodes.

1.3 Networked robots

Sometimes, mobility itself is a component of the application goal, for instance in exploration, deployment, or retrieval missions. In these cases, there are still synergies between sensor networks and mobile robots. Distributed networks can be used as a communications infrastructure, allowing for robust two-way data transfer over ranges far longer than a single-hop line-of-sight link. Additionally, a network of distributed nodes can gather necessary environmental data for use by the mobile robots, including for example data about the

Figure 1.1: A possible home of the future contains an distributed wireless network with pervasive sensing, immersive actuation, and integrated mobile elements.

robots themselves to assist in guidance and navigation control.

1.4 Potential applications

Such systems of networked mobile swarms have uses throughout the consumer, industrial, and military sectors...

Chapter 2

WARPWING and the GINA Mote

From sensors and devices through autonomous robots to swarm behaviors, there are a number of infrastructure requirements necessary to test and develop various robotic and network system designs. In particular, processing and wireless communication are necessary for any such endeavors. Typically, an inertial measurement unit (IMU) is also necessary for localization of mobile nodes.

Many research groups independently design their own hardware solutions, or adapt a handful of fixed commercial subsystems to their custom application. However, these approaches are often time-consuming and inefficient, causing valuable research effort to be spent re-creating solutions, often by those inexperienced in hardware design.

The Wireless Autonomous Robot Platform with Inertial Navigation and Guidance, WARPWING, was designed to streamline robot development by collecting all the electronics requirements for robot control in one open source project, from hardware selection to board

design to firmware and software programming. By providing a full solution in an open source format, researchers can focus solely on their areas of research, while still maintaining a cohesive environment in which to develop and test a complete system. As users adapt parts of WARPWING to their varied custom applications, their changes are also shared with the community, obviating the need for redundant independent development.

WARPWING is centered around the Guidance and Inertial Navigation Assistant, GINA, shown in figure 2.1. This is a low mass wireless inertial measurement unit (IMU) designed for use as a general purpose micro air vehicle (MAV) or robot controller. It comprises inertial sensors for angular rate and linear acceleration along with a general purpose feature-laden microprocessor. It can interface to additional sensors or drive a number of actuators (some of which are described below) via an expansion header, and provides built-in communication over a 2.4GHz wireless link.

The WARPWING project consists of hardware components – a device database, schematic design, and board layout – and software components – firmware for the onboard microprocessor and applications and modules for a basestation computer – along with relevant documentation. This entire project is made available on Sourceforge, a popular open source community site, at `http://warpwing.sourceforge.net`.

Section 2.1 describes some of the considerations driving the WARPWING project decisions and GINA design. The specific designs are presented in section 2.2, detailing the current GINA hardware along with the firmware and software needed to use the GINA board. Section 2.3 expounds on the open source nature of the WARPWING project, giv-

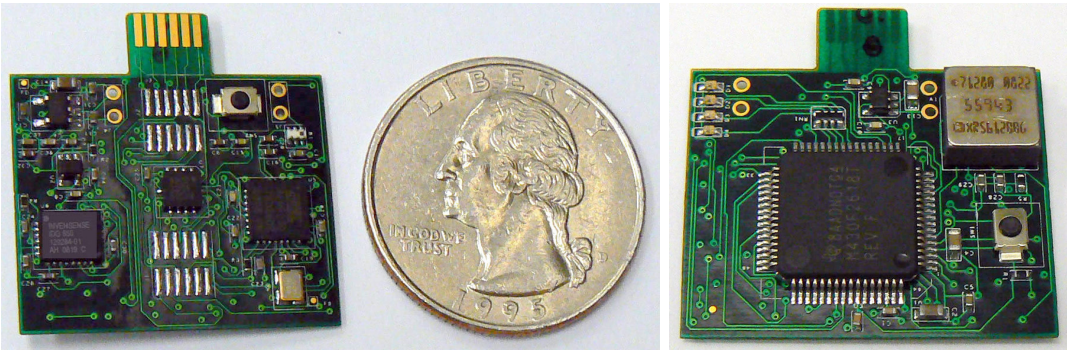

Figure 2.1: GINA provides inertial sensing, processing, control, and wireless communication on a board smaller and lighter than a US quarter.

ing examples of modifications made to the default project by other research groups in their robotic applications. Finally, section 2.4 examines the case studies from the previous section to draw some conclusions about open source hardware for robot development, and presents a vision for how this project can evolve in the future.

2.1 GINA history

Initially created as a flight control system for a small inertially guided rocket (see section 3.1), GINA was designed to be the smallest, lightest, and lowest power stability and guidance control solution with off-the-shelf components. It was then also used on a miniature coaxial helicopter for use as an autonomous MAV, as in section 3.2. As development on both those projects continued in parallel, it became clear that the requirements for both, and indeed a large portion of autonomous air vehicle and small robot research in general, were highly similar.

In practice, the overall size of a robotic system is often dictated by the weight it needs to carry, so minimizing overall size requires minimizing the total mass of the system. Lowering power draw is also beneficial, as it reduces the size of the battery necessary. The specifics of the GINA hardware are described in the following section; the primary design consideration in the development of the GINA electronics was to minimize weight while still providing the required functionality for a general autonomous robot.

The 6 cm^2 GINA board weighs 1.6 grams and runs off of a single 3.7V lithium polymer cell. It consumes a maximum of 30mA at 3V when fully powered; intelligent duty cycling can make use of low power modes to lower average power and increase lifetime. The system uses entirely commercial off-the-shelf components aside from a custom printed circuit board (PCB), and can be made in its entirety in one-off quantities for less than $200.

2.2 WARPWING architecture

2.2.1 Hardware design

Inertial sensors

GINA has full 6-axis inertial sensing, composed of 3 MEMS sensors.

Angular yaw rate is measured by an Analog Devices ADXRS610 single axis analog gyro. The pitch and roll angular rates are measured by an Invensense IDG650 dual axis analog gyro. These gyros measure rates up to $\pm 300^\circ/s$, and are digitized by an analog to digital

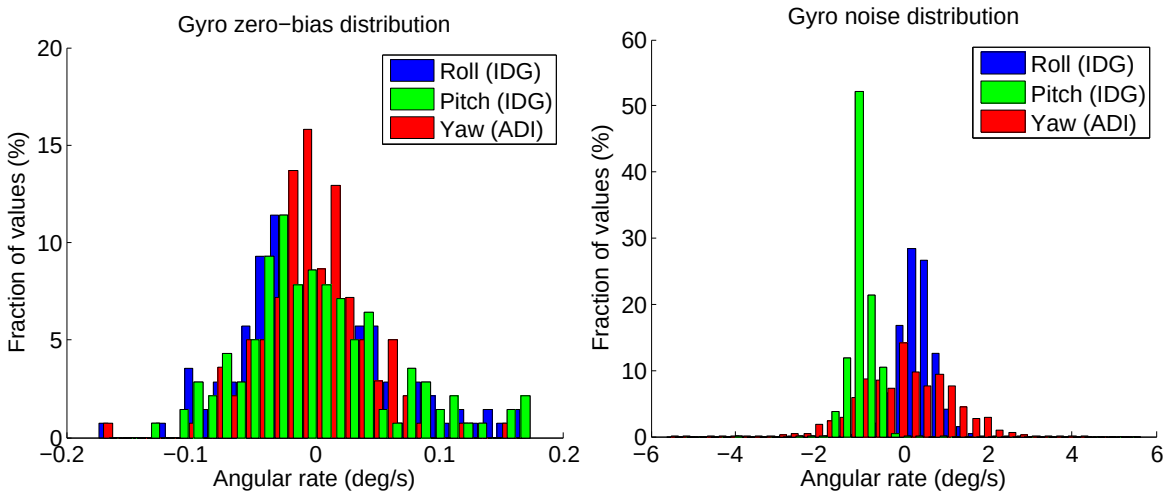

Figure 2.2: The variation of the gyro zero-bias for a single board across 150 runs over 5 days is shown in the graph on the left for each of the three axes. The graph on the right shows the zero-rate gyro outputs over 2000 measurements during a single run. The bias offsets are clearly visible in the right graph, as well as a notable difference in performance between the two gyros.

converter (ADC) on the processor. Both of these devices have on-chip temperature sensors for use in calibrating out temperature coefficients.

There are two components to the noise in the gyro outputs. First, after subtracting a linear temperature-dependent offset, the zero-bias of the gyros varies across runs. Second, in a single run, the output of the gyro displays additive white gaussian noise in each measurement.

These noise terms are shown in figure 2.2 and summarized in table 2.1.

Linear motion is detected using a Kionix KXSD9-1026 3 axis digital accelerometer, measuring a configurable full-scale range of ±2g, ±4g, ±6g, or ±8g with a 12 bit ADC at 400 Hz. It has a much lower temperature sensitivity and zero-bias noise, so can be assumed to have

Table 2.1: Noise in the inertial sense axes

Axis	Bandwidth	Measurement noise σ	Zero-bias error σ_{bias}
a_x	500 Hz	$.57m/s^2$	N/A
a_y	500 Hz	$.46m/s^2$	N/A
a_z	500 Hz	$.61m/s^2$	N/A
ω_p	140 Hz	$0.42°/s$	$0.07°/s$
ω_q	140 Hz	$0.26°/s$	$0.06°/s$
ω_r	140 Hz	$1.04°/s$	$0.04°/s$

a single zero-mean additive white gaussian noise term added to each measurement across all runs. These values are summarized in table 2.1.

It is important to note that though this sensor suite is comparable to or better than commercially available IMU options, these are merely the ones currently used on this board; there are now more advanced (e.g. lighter, lower noise, more tightly integrated) sensors on the market. Because the schematics are open source, it is a fairly straightforward task to replace the sensors with their newest counterparts; once done, it becomes available to anyone else who would like to use it. In fact, as will be described below, one such revision includes a more sensitive accelerometer to be used for vibration sensing instead of motion detection.

Wireless communication

Communication to the board is handled by an Atmel AT86RF231 wireless radio. This is an IEEE 802.15.4 compliant 2.4GHz RF transceiver capable of half-duplex communication at 250 kbps. It is fully interoperable with off the shelf 802.15.4 devices, so no special hardware is necessary to communicate with a GINA mote. It also has enhanced non-standard modes

capable of communicating at higher data rates up to 2 Mbps.

GINA has an both an on-board chip antenna as well as a micro-coax connector for an external antenna. The appropriate antenna can either be manually selected, or automatically chosen by the radio based on received signal strength. This allows for generally reliable communication within a room-sized area, although care may need to be taken to mitigate interference and multipath effects (see [46] for a possible solution). For long range communication, directional high-gain antennas can be used at the basestation and/or on the GINA mote itself; a +14dBi antenna at the basestation and a +8dBi antenna on the mote allowed reliable communication over a 1km long line-of-sight link. A collection of motes can be connected in a mesh network.

Processing

The on-board processing is handled by a Texas Instruments MSP430F2618. This is a highly capable 16 MHz 16 bit microprocessor. It has onboard analog to digital converters (ADCs) and an I2C serial interface to read the sensors, and an SPI module to interface to the radio. A hardware multiplier allows for 16 bit integer or fixed point single cycle multiplies. Additional on-chip functionality includes direct memory access, timers, and a well developed interrupt stack.

The processor also has a number of peripherals, accessible via expansion headers on the GINA board. In particular, in addition to general purpose input-output (GPIO) pins, there are two additional serial ports to read from and write to digital sensors and actuators, as

well as a handful of pulse-width modulation (PWM) outputs to drive conventional robot actuators such as servos and motors.

Daughter boards

The expansion headers are intended to connect to an application-specific daughter board. A generic break-out board, as shown in figure 2.3, was designed to allow easy breadboard prototyping; a USB-serial connector was also included to allow for enhanced debug communication to and from the GINA. The expansion headers were passed through the board to allow for debugging of other daughter boards as well.

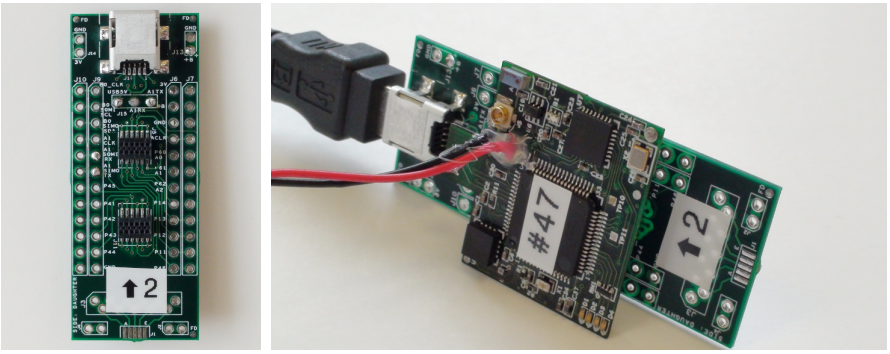

Figure 2.3: A break-out daughter board allows the GINA board to be connected to a standard breadboard for quick prototyping, with

A robotic controller daughter board was also designed to route commonly used signals to appropriate connectors, as shown in figure 2.4. Two independent sets of PWM outputs from the processor can be used to control up to 3 servos (or brushless motor controllers) and 2 motor drivers consisting of a mosfet and flyback diode.

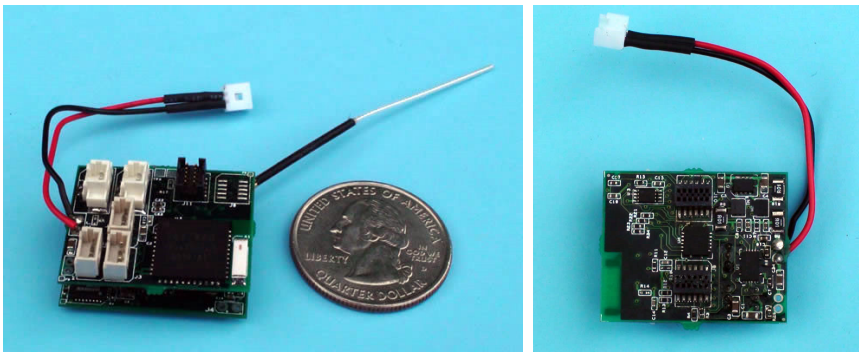

Figure 2.4: A daughter board for the GINA mote brings PWMs out to connectors for servos, uses other PWMs for brushed motor controllers, and includes additional sensors such as a battery fuel gauge, GPS receiver, and vision processors.

Additional sensors were also included for evaluation purposes on this daughter board. A GPS receiver with an on-board chip antenna is connected to one of the serial ports, while a battery monitor / fuel gauge is connected to another. Connectors were also included to interface to Wiimote cameras, which provide IR blob detection for an off-the-shelf vision-based localization system.

Recent updates

The WARPWING platform has been constantly evolving to take advantage of newer and better hardware. In particular, as the MEMS sensor suppliers release ever more powerful sensors, sensing fidelity can continue to improve. Work has begun on GINA 4.0, a smaller, lighter, more functional mote that comprises a single 9 axis IMU sensor along with an integrated processor / radio that can communicate directly over Bluetooth 4.0 to a number of mobile devices.

2.2.2 GINA microprocessor firmware

The MSP430 microprocessor on the GINA board was programmed using embedded C and compiled in the IAR integrated design environment (IDE). The basic firmware was designed to be broadly applicable for most requirements. The processor starts up in a holding loop, listening for commands both over the 802.15.4 radio (for wireless operation) as well as the serial port (for use with a USB-serial connection to a computer). A number of commands and operating modes are available, and described in the API documentation available with the source code on Sourceforge.

The most relevant mode is the control loop. A hardware timer is used to run this loop every 3 ms. Every iteration, the inertial sensors are polled and the data is placed into a packet sent out over the 15.4 radio. Then the processor waits for commands, in particular setting the PWM duty cycles for actuators. This enables an off-board feedback controller operating on a basestation computer to receive and process data to generate actuator control signals, and pass that back to be implemented on the GINA with a short latency.

This loop provides a sensor update rate of 333 Hz, with about 48k instruction cycles available per loop. With the processor only handling interfacing with the sensors and radio, most of these cycles are spent idle; an autonomous system can use these to implement an application specific feedback control directly on-board.

2.2.3 Basestation software

The default behavior of the GINA firmware offloads the responsibility of actual robot control to a basestation computer, allowing for much easier design and development of robot control. A hardware device is required to allow the computer to speak to the GINA using the 802.15.4 wireless standard; any such transceiver will work, but slightly specialized firmware has been written for the Atmel RZ USBStick to simplify the basestation software.

The software must then interface to the RZ USBStick (appearing as a serial port to the computer); again, the full API is documented with the source code available on Sourceforge. A collection of utilities has been written in Python, forming an easy to read and modify cross platform skeleton upon which to build application-specific software. Default programs that are included in the WARPWING package allow real-time graphing of IMU outputs, data logging, and manned control using a 4 axis joystick to control two motor and two servo outputs on the GINA mote. More advanced routines, such as an extended Kalman filter (EKF) to generate an attitude estimate from the inertial rates (adapted from [16]) have also been developed.

As is common in the open source world, users sometimes write their own software and make it available to the community. Matlab and Labview routines have been implemented by groups using WARPWING, and can also used to interface to GINA motes.

Figure 2.5: The Microstrain 3DM-GX2, a commonly used commercial IMU, can be used to calibrate and validate the performance of the GINA board

2.2.4 Performance

The Microstrain 3DM-GX2, shown in figure 2.5, was one of the smallest commercially available IMUs at the time. It weighs 16g without its enclosure, and consumes 90mA at 4.5V. It is very widely used in autopilots for unmanned aerial vehicles, and has an optional wireless-enabled model. The 3DM-GX2 costs $1695, or $3095 for the wireless model. From its website, the "3DM-GX2 is a high-performance gyro enhanced orientation sensor which utilizes miniature MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm." [29]

A GINA board was mounted to the 3DM-GX2 unit, and data was collected simultaneously from both devices. The scaled data is shown in figures 2.6 and 2.7 for a sample angular rate and linear acceleration trace. It is seen that GINA provides sensor outputs that are slightly noisier than the commercial device, but otherwise matches the output very closely. A 35 ms latency in the 3DM-GX2 is also visible from this data. Taking into account this lag, the sensor data from GINA has a $> 99.5\%$ correlation to the commercial IMU output.

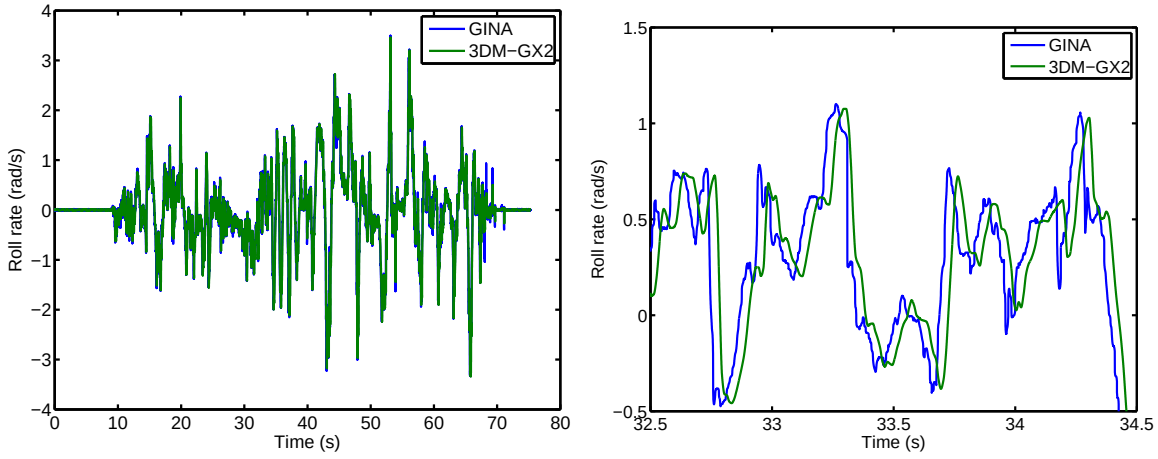

Figure 2.6: Comparison of gyro output between GINA and 3DM-GX2

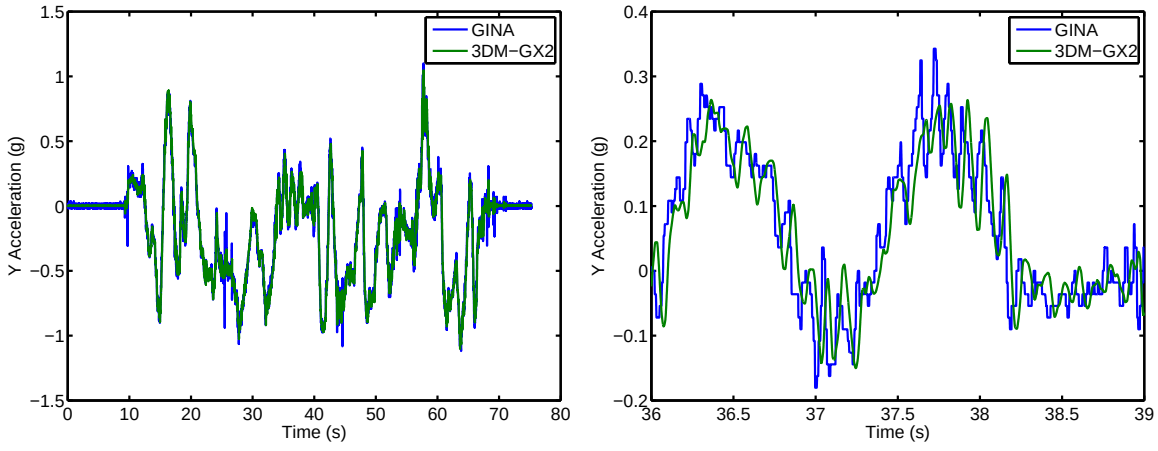

Figure 2.7: Comparison of acceleration output between GINA and 3DM-GX2

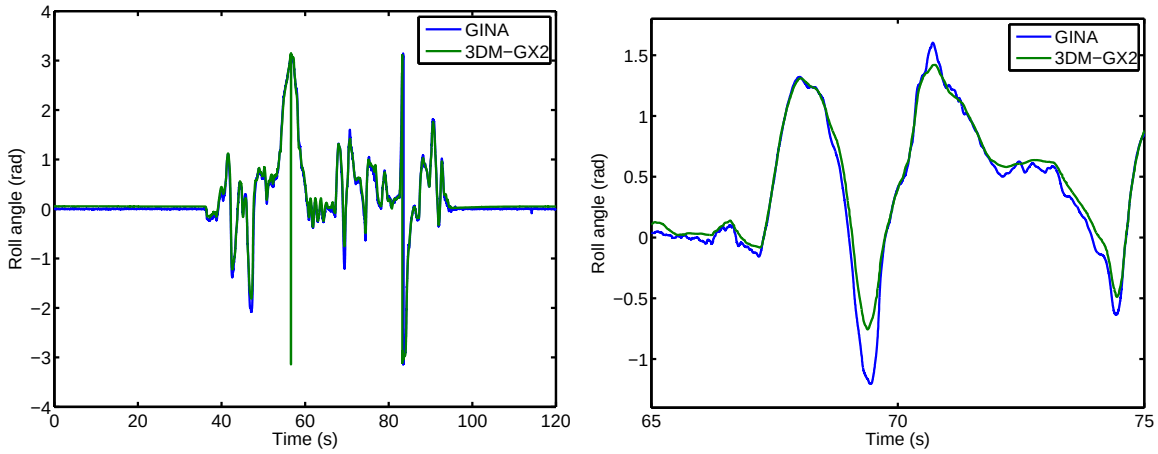

Figure 2.8: Roll angle estimate comparison. GINA and 3DM-GX2 agree closely in their state estimates.

The 3DM-GX2 also provides an attitude estimate which can be compared with the output of an EKF running in real time on GINA data. The two attitude angle estimates are shown in figures 2.8 and 2.9. It is again seen that the GINA estimates are a bit noisier, but this comes with a quicker response time to motion. This time constant is most visible at the end of the pitch trace. The attitude estimates between GINA and 3DM-GX2 have a $> 98\%$ correlation.

2.3 Modification and Sharing

The goal of the WARPWING project is not just to present an ultra-small full-featured wireless IMU and robot controller, but to use that to advance other robotic research as well. Any aspect of WARPWING can be modified to fit the specific needs of a design. Thus, even with little to no electronics ability, researchers can have a platform on which to base

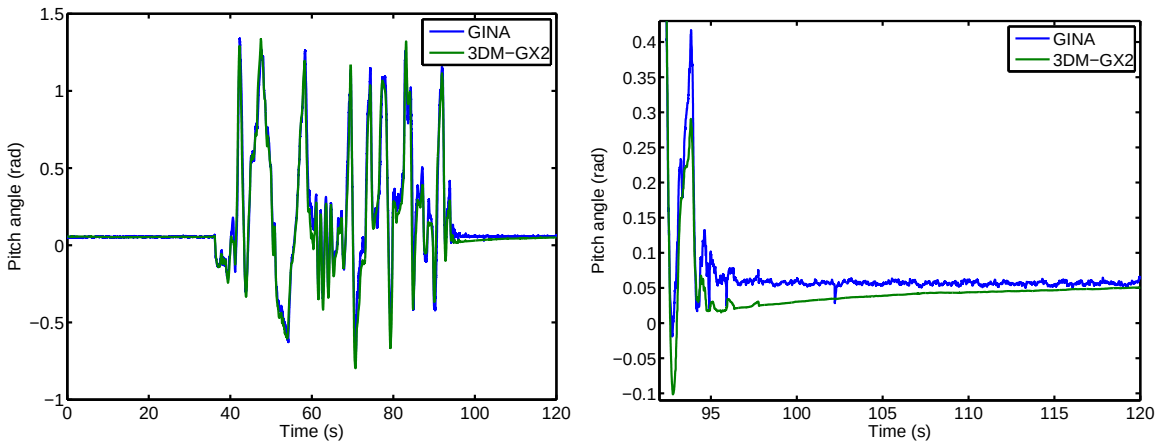

Figure 2.9: Pitch angle estimate comparison. The noisier GINA estimate also allows a much shorter settling time than the 3DM-GX2.

an inertial robot controller.

The default WARPWING distribution as described above allows researchers to mount the wireless GINA mote to a dynamic system and log its inertial rates on a basestation computer, all out of the box. The system was distributed to over a dozen research groups around the world, and all were able to collect data within days. Some examples are shown in figures 2.10, 2.11.

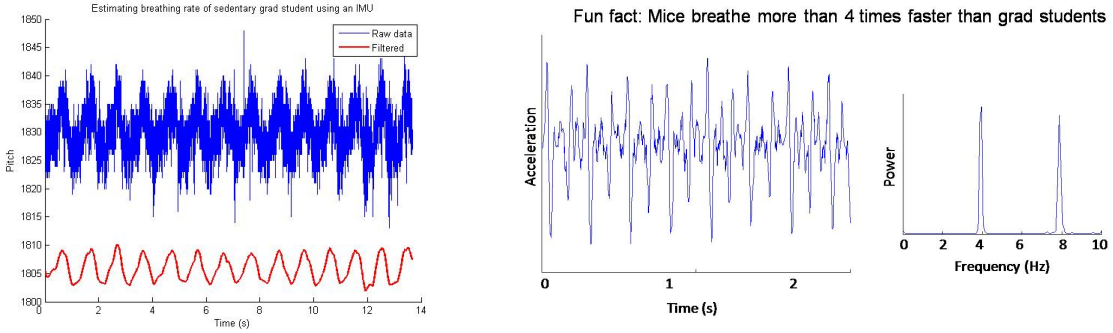

Figure 2.10: Inertial data collected from GINA motes mounted on mammals is used to calculate sedentary breathing rates. Data courtesy Subramaniam Venkatraman.

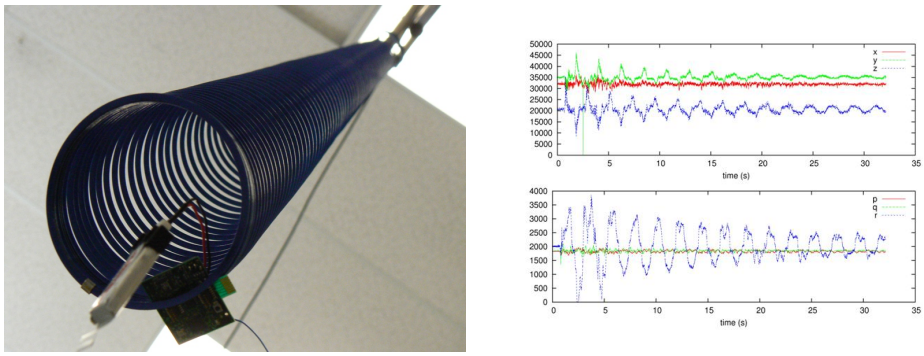

Figure 2.11: Inertial data collected from a GINA mote mounted on a coiled spring, demonstrating both oscillatory rotation and translation. Data courtesy Thomas Watteyne.

When used for more than simply measuring inertial rates, however, the default WARPWING configuration is likely insufficient for controlling an arbitrary robot. But because the system is very general, typically only a few specifics need to be modified to build the desired controller. Because of the nature of the open source community, these changes are shared, simplifying similar changes in the future.

There are already projects using WARPWING that have modified some or all of the components of the entire system. A select few are summarized below as case studies for the ways WARPWING can be adapted to simplify robot development.

2.3.1 Software

In order to use WARPWING to control a specific robot, the software is almost certainly going to need to be rewritten. For a large portion of robots, however, that will be the only component that needs to be modified.

The Costello Research Group at the Georgia Institute of Technology have developed a

hybrid air/ground vehicle called the hopping rotochute that uses rotors to generate the lift required to cause the robot to hop over obstacles and along trajectories [5, 6].

The default hardware and firmware configuration for the GINA mote is sufficient to control the actuators, and so with the control hardware taken care of by WARPWING, research effort can be dedicated to mechanical vehicle design and control algorithms for newer versions of the hopping rotochute, as shown in figure 2.12. The software only needed to be modified insofar as to interface the existing controller written in MATLAB to the WARPWING basestation transceiver.

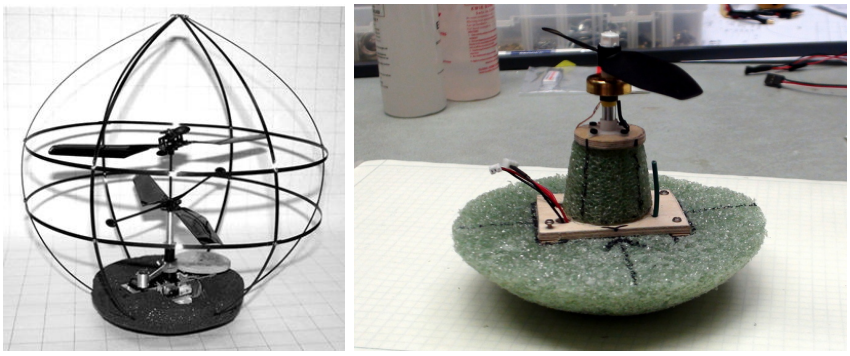

Figure 2.12: The hopping rotochute developed at GATech uses WARPWING as its flight controller. The software interface was rewritten to fit in to the existing MATLAB framework, while the rest of the WARPWING system remained unchanged. This allowed for quick development of the mini rotochute, shown at right.

The WARPWING setup can also be used for non-robotic applications. The small size and weight of the GINA board makes it particularly suitable for body sensor network applications. In particular, inertial sensing can be used to estimate the pose of body joints and extremities. Figure 2.13 shows a plastic ring holding a GINA board – the default code can be used to gather data on the behavior of the wearer, as shown in figure 2.14. Custom software can

be written to interpret this data, for instance by combining estimated pose and measured dynamics to estimate letters typed on a virtual keyboard.

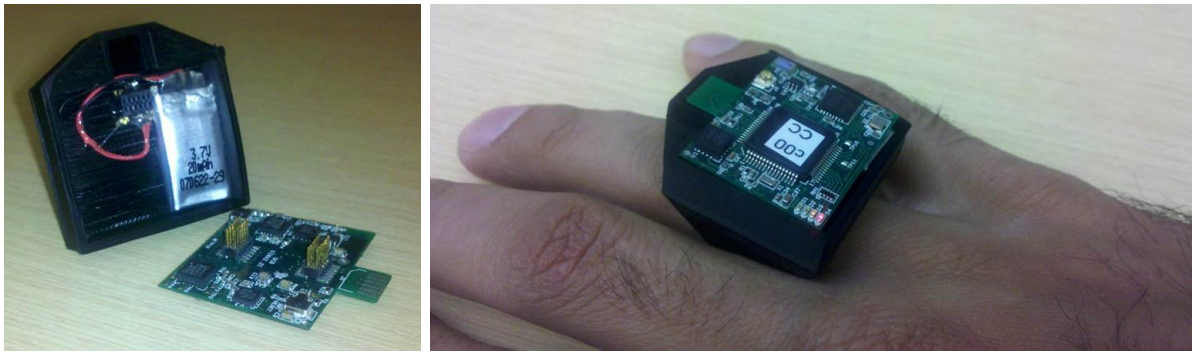

Figure 2.13: The RinGINA device contains a GINA mote on a custom manufactured ring. The WARPWING platform was used to quickly develop an unobtrusive wireless body-mounted sensor.

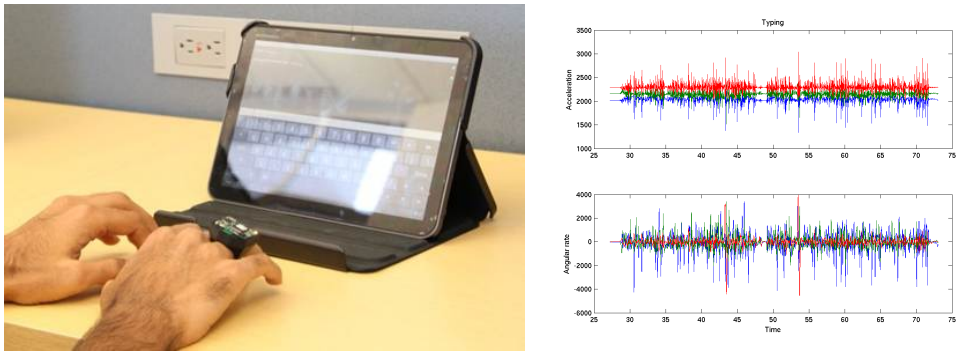

Figure 2.14: The WARPWING code can gather inertial data from the RinGINA, measuring 6DOF state and dynamic motion. Custom software can be written to translate the data into virtual keypresses to eliminate a physical keyboard.

2.3.2 Firmware

For more complicated robots, with more advanced onboard behavior, the firmware on the GINA mote may also need to be modified. A mini quadrotor, shown in figure 2.15 has been

Figure 2.15: The mini quadrotor developed by Daedalus Flight Systems and UMD uses the GINA hardware, but modifies the firmware to include an onboard attitude stability loop to control its four brushless DC motors.

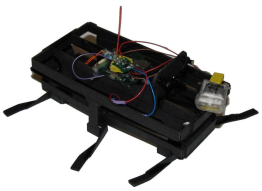

Figure 2.16: DASH, a small, lightweight crawling robot developed at UC Berkeley, uses the WARPWING system to measure inertial rates and motor back EMF for dynamic analysis and control.

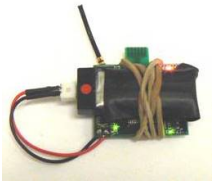

Figure 2.17: This small lightweight footstep detector is designed from the schematic level on up using the WARPWING system as a base, but replacing the accelerometer with one far more sensitive.

developed by Daedalus Flight Systems and the University of Maryland. It uses four brushless DC motors for its actuation and requires an onboard attitude stability loop.

The WARPWING platform took care of the majority of the flight control system requirements, and the only elements that needed to be added in firmware were specific to the feedback control loop. Routines were written to derive setpoint values for the inertial rates, mix them with joystick inputs sent via radio, and wrap a loop around the measured rates to drive the PWM outputs controlling the motors. For this project, the software was also ported to Labview for convenience of the researchers.

Because the hardware and the bulk of the firmware was already implemented in WARPWING, the group was able to focus on mechanical system design and development and control system tuning.

The firmware on the GINA was also completely rewritten for the OpenWSN project [45],

a non-robotic application of the hardware. In that project, the board was used as a node in a distributed wireless sensor network; completely new firmware was written for a standards compliant implementation of the IEEE 802.15.4e network stack.

2.3.3 Layout

The Biomimetic Millisystems Lab at the University of California, Berkeley have developed a hexapedal crawling robot called DASH [7]. In order to study the dynamics of DASH towards a future adaptive controller, the researchers needed to incorporate an IMU onto the robot. The WARPWING platform provided a lightweight, low power, and easy to use single board solution that interfaced to the existing motor driver to control the robot.

One additional requirement for the system was to measure and report an analog signal from the robot: the back EMF of the motor. In the default GINA layout, there was an unused ADC input to the processor, but it wasn't exposed in the expansion header. It is possible then to adjust the layout of the GINA board to rewire an ADC input to an expansion pin rather than a GPIO, allowing it to be used for the additional analog sensor. In this case, the firmware was also modified slightly to read that ADC channel and send it back over the radio, and the software was amended to handle that data appropriately.

2.3.4 Schematic

Another project at UC Berkeley considered the use of networked sensors (shown in figure 2.17) distributed across a floor in a room to determine the location of footsteps using the time difference of arrival of the vibrations at each of the sensors. These sensors could potentially be dispersed by slightly larger autonomous robots, and so minimizing size and weight is an important consideration.

The WARPWING system can provide most of the needed functionality from measuring sensor readings to forming a time-synchronized mesh network, but the accelerometer is designed for robotic systems requiring higher dynamic range at the expense of sensitivity. The system was modified to replace the accelerometer with a much more sensitive analog sensor. This required the change to propagate through the layout, firmware and software, but still much of the system remained unchanged, greatly speeding development time.

Ultimately, this new sensor was incorporated as a permanent addition onto the next revision of the board.

2.4 Summary

The open source model for software development has begun to make inroads into the hardware community, and the WARPWING project fully embraces that approach. Starting from GINA, an extremely small and lightweight but powerful wireless IMU and processor, the WARPWING project provides a very broad base from which to design robot controllers and

more generally sensor motes for which size and weight are at a premium. The examples presented above show that any of the system can be changed, whether it be a sensor composing GINA, the layout of the board, the firmware running on the GINA processor, or the software running at the base station. The rest of the system can still be used as is, thus obviating the need for redundant development effort, and greatly increasing the efficiency of robotic research and design.

As is the nature with open source projects, developments made by users in the community are fed back into the project to be spread throughout the system. As more users contribute to WARPWING, it becomes more versatile and less new development is needed for new applications. In much the same way as standard fabrication processes for IC and MEMS devices have pushed research farther into device design, the current WARPWING platform similarly simplifies robot controller development, and drives research to more efficiently consider higher level designs.

Chapter 3

Micro Air Vehicles

The GINA board of the previous chapter was borne out of the need for a robot controller; it was then employed in a number of research robots.

A robot is defined by its mobility; there are a number of actuation options for mobile robots. Ground based systems can employ wheels or legs, while aerial vehicles can use fixed or rotary wings. Though specific applications usually admit a particular optimal mobility subsystem, general or multipurpose robots often must consider tradeoffs between the various options. Flight is typically the most versatile, allowing maneuverability in all three dimensions; however, it is correspondingly the most difficult to implement.

Vital to autonomous behaviors, and arguably the most difficult component of such operations, is localization. Without external infrastructure, the robot must be able to calculate for itself an estimate of its position within its environment. In practice, localization turns out to be the biggest challenge in implementing micro air vehicles (MAVs).

3.1 Small scale rocketry

Current trends in miniaturization have resulted in the general reduction in size of electronic systems; the same principles can potentially be applied to satellites and their launch vehicles. Drawing from progress in wireless sensor network (WSN) technology, microelectronic systems can potentially be used in such space communication networks.

WSNs are of increasing interest in both academia and industry. Distributed sensing, processing, and communication have far reaching applications; indeed WSNs have found their way to all varieties of settings around the world, from oceans to plains to mountains to urban environments [34]. There is considerable ongoing research employing WSNs for their distributed nature, as well as research into WSNs themselves in areas such as reliability and robustness, communication theory, and distributed algorithms.

An area not yet infiltrated by WSNs though is space – orbits above Earth's atmosphere and beyond are still dominated by large one-off spacecraft. Only recently has there been analysis of potential deployments in space (such as in [20], [43]). Such satellite systems would be able to address fundamental wireless communications research in the absence of notable interference from ground based sources and physical obstacles, as well as conduct atmospheric and astronomical research. As the availablity and functionality of electronics go up and the cost goes down, the required hardware becomes smaller, cheaper, and more accessible. While previous small satellite research has focused on systems on the order of kilograms, sensor nodes have shrunk to where a 10 gram system is sufficiently powerful for

many purposes.

The deployment of small-scale satellites for these orbital systems requires a launch vehicle (LV) of comparable scale and cost to the payload mote being deployed. Designing such a vehicle requires analysis of the limits on miniaturizing a ground-to-orbit system, and in particular, conditions on and designs for the orbital insertion of sub-kilogram payloads. There are two main requirements on a LV to put a mote into low earth orbit (LEO): the system must deliver enough energy to attain an orbital trajectory, and it must be controlled to follow the insertion path.

3.1.1 Energy to LEO

Energy considerations drive the mechanical rocket design, constrained by physical properties of available materials.

General rocket equations

The effort needed to execute any orbital maneuver is commonly described by the scalar Δv or delta-v, referencing a change in velocity. In order for a LV to reach LEO from rest on the surface of the Earth, the required Δv_{leo} is composed as follows [36]:

$$\Delta v_{leo} = v_o + \Delta v_d + \Delta v_g + \Delta v_c + \Delta v_{atm} - v_{rot}. \tag{3.1}$$

The first term of equation (3.1), v_o, is the speed needed to keep the payload in orbit around the Earth. The remaining terms are mainly dependent on the trajectory of choice to

reach this particular orbit. For a circular orbit,

$$v_o = \sqrt{\frac{G \cdot M_e}{(R_e + h)}}, \tag{3.2}$$

where G is the gravitational constant, M_e is the mass and R_e is the radius of the Earth, and h is the orbit altitude. Equation (3.2) indicates that v_o for a satellite in a LEO at $200km$ altitude is approximately $7.8km/s$. The additional Δv terms are Δv_d due to air drag, Δv_g needed to climb against Earth's gravitational potential, Δv_c in losses when effecting a desired trajectory, and Δv_{atm} for lower engine performance during the ascent in the atmosphere. These losses sum up to an additional $1.5 - 2km/s$, despite the direct Δv boost caused by the Earth rotation $v_{rot} = 463m/s$ at the equator [36]. All told, a Δv_{leo} of 9.5 - 10 km/s is required to reach LEO for a ground launched LV.

Calculating the Δv generated by a launch vehicle's (LV) propulsion system requires a time history of its instantaneous thrust ($|F|$) and mass (m):

$$\Delta v = \int \frac{|F|}{m} dt, \tag{3.3}$$

where the integral is carried out over the duration of the maneuver in question. This can be evaluated out for a specific rocket design to yield the ideal rocket equation

$$\Delta v = I_{sp} \cdot g \cdot \ln\left(\frac{m_i}{m_f}\right). \tag{3.4}$$

The specific impulse I_{sp} is an intrinsic property of the fuel and $g = 9.81m/s^2$ is the acceleration due to gravity. Clearly, the Δv of the LV depends strongly on the rocket's mass ratio between the initial take-off mass m_i including the mass of the fuel and the final mass m_f

after burnout consisting of only the payload and structural mass of the rocket. Multistage rockets are evaluated by summing the Δv's of each burn calculated independently.

Adaptation for minirockets

A rocket experiences a drag force

$$F_d = \frac{1}{2}\rho C_d v^2 A, \tag{3.5}$$

where ρ is the density of the surrounding air, C_d is the drag coefficient, v is the velocity of the rocket relative to the air, and A is the rocket's cross sectional area. This results in a penalty

$$\Delta v_d = \int \frac{F_d}{m} dt, \tag{3.6}$$

that scales up with decreasing length.

With this increased Δv requirement, the rocket equation (3.4) indicates that the structural mass of a small scale LV must be kept as low as possible. Miniaturization of electronics and advancements in structural materials have enabled the miniaturization of satellite payloads and associated LV, and the primary design consideration for such a system is keeping a low mass.

The goal of course is to be able to launch a 10 gram satellite into orbit from the surface of the Earth. However, the payload delivery rocket can itself piggyback on other vehicles to reduce the Δv requirement of the rocket system. The minirocket can be used for orbital trajectory insertion, with an ultra high altitude balloon (UHAB) or large scale rocket carrying

the rocket system to the upper atmosphere.

The concept of launching rockets from a balloon is not a new idea and has been performed extensively during the 1950s by J.A. Van Allen for upper atmospheric research [42]. Deployment of a 690kg payload to a peak altitude of 49.4km by a UHAB has been demonstrated [33]. Similarly, conventional rocket LVs often have spare payload capacity which can be used to deliver much small systems [41].

There are a number of energy advantages for air-based launches:

- Starting powered flight above 98.5% of the atmosphere, the LV experiences less than 3% of the drag force compared to a ground launched vehicle, tremendously decreasing Δv_d [18].

- Δv_g and Δv_c are also lowered since the LV has to fight Earth's gravity and follow a given trajectory for a shorter amount of time. Additionally, the LV doesn't need to compensate for wind gusts, a concern at lower altitudes.

- The engine operates at peak performance when exhausting into near vacuum, directly increasing thrust F due to an increased ratio between combustion chamber and ambient pressures [31].

Minirocket design

The mechanical design of a complete miniature rocket LEO LV requires a complicated interplay of materials science, aerodynamics, and mechanical engineering – it is a significant

undertaking, and beyond the scope of this work. Nonetheless, first order analysis of such a system is necessary to establish the feasiblity of such a system.

To keep costs, complexity and structural mass at a minimum, a solid propellant seems to be favorable for a small-scale LV. The need for pipes, valves, tanks, and insulation in liquid propellant engines would contribute to a high overall structural mass. The main disadvantages of solid-fuel propellant are the lower specific impulse I_{sp} and the lack of active throttling, though the latter could potentially be overcome by a combination of intelligent propellant grain design and control system tuning, allowing specific thrust-time characteristics.

A solid fuel rocket is little more than a open-ended cylinder packed with fuel forming the combustion chamber with a nozzle for exhaust emission, upon which the control system and actuators are mounted. To minimize structural weight, high tensile strength carbon fiber can be used to form the combustion chamber. A standard solid rocket fuel consists of an ammonium perchlorate oxidizer in a hydroxy-terminated poly-butadiene (HTPB) binder. The physical characteristics of such a system are summarized in table 3.1.

With these parameters, and some optimistic but not unreasonable design assumptions on structural integrity (e.g. ignoring the possibility of chamber wall ablation or buckling), a design optimization can be carried out on the geometry of such a rocket. The minimum (total system) mass cylindrical rocket can be calculated by imposing constraints of structural stability (the carbon fiber rocket body must be able to withstand the pressure of combustion) and mass fraction (derived from the necessary total Δv from equation 3.1 along with the

Table 3.1: Physical parameters of a proposed solid fuel minirocket system

Parameter	Subsystem	Value	Units
Fuel exhaust velocity	Fuel	2.6	km/s
Fuel density	Fuel	1.763	g/cc
Chamber pressure	Rocket body	100	atm
Chamber pressure	Rocket body	3	GPa
Chamber pressure	Rocket body	1.75	g/cc
Payload mass	Payload	10	g
Control system mass	Controls	10	g

Table 3.2: Single stage to orbit (SSTO) rocket designs

Δv (km/s)	Diameter (cm)	Length (cm)	Wall thickness (μm)	Mass (kg)
10	3.1	101	51	1.3
12.5	6.6	218	110	13.3
15	n/a	n/a	n/a	n/a

rocket equation in equation 3.4).

For a single stage to orbit (SSTO) rocket, then, the resulting geometry is shown in table 3.2. Though $\Delta v_{leo} = 10$ km/s is easily attainable, adding a 50% safety margin proves impossible. Instead, a multi-stage rocket would be necessary. The generated geometries for a two stage rocket are shown in table 3.3. A 50% margin of error is still quite achievable, yielding a rocket system under 10 kg in mass and 2 m long. A 100% safety margin, though unwieldy, is also achievable. For such higher total Δv, even more stages would be recommended. Nonetheless, this shows the feasibility of a backyard-scale minirocket that can deliver a 10 g payload into LEO.

Table 3.3: Two stage rocket designs for LEO insertion

Δv (km/s)	Diameter (cm)	First stage length (cm)	Second stage length (cm)	Wall thickness (μm)	Mass (kg)
10	2.0	156	39	33	1.1
12.5	4.0	112	13	66	3.0
15	5.7	178	14	95	8.8
20	9.2	900	13	154	108

3.1.2 Guidance control system

System design

Active control is necessary to guide a payload along a trajectory to reach LEO. A typical solid rocket motor is often dynamically unstable, as shown in figure 3.1. Furthermore, there is no ballistic path to LEO ("what goes up must come down"). As described above, the lower stages of a multi-stage rocket can be replaced by an alternate carrier such as a balloon or larger scale rocket for an air launch. In either case, though, the uppermost stage of the minirocket will perform the final orbital trajectory insertion.

To guide the LV along the specified trajectory, a controller must determine the position and attitude of the rocket to subsequently counteract any deviation from the preset flight path. Ideally, the full six degree-of-freedom (6DOF) position would be known, identifying both its location and orientation in space. Typically the controller employs an inertial measurement unit (IMU) augmented with additional sensors to generate an accurate estimate of the 6DOF state of the LV from inertial rate measurements and external observations.

From the state estimate, corrections need to be applied to maintain a desired flight profile.

Figure 3.1: In the absence of aerodynamic forces, a bare solid rocket engine spins out of control. A stabilizing controller is therefore necessary on a minirocket LV.

Given that most of the flight occurs in extremely thin atmosphere, aerodynamic surfaces such as fins would be useless for the LV's attitude control. Therefore, more sophisticated systems such as a gimbaled nozzle for thrust vectoring or controlled offset mass for torque generation need to be considered. The overall system is diagrammed in figure 3.2.

State estimation and control

Implementation of the rocket controller began with an evaluation of the sensing subsystem. A board containing the GINA gyro sensors was attached to a CC2430 evaluation board for wireless data offloading; a receiver was attached to a laptop to log the sensor data over the duration of a rocket flight. This sensing package was mounted in the nose cone of a rocket alongside a video recorder aimed at the ground. This system is displayed in figure 3.3.

The measured gyroscope data during a flight of this test rocket was collected by a laptop basestation, as annotated in figure 3.4. An integration routine was run on the data to

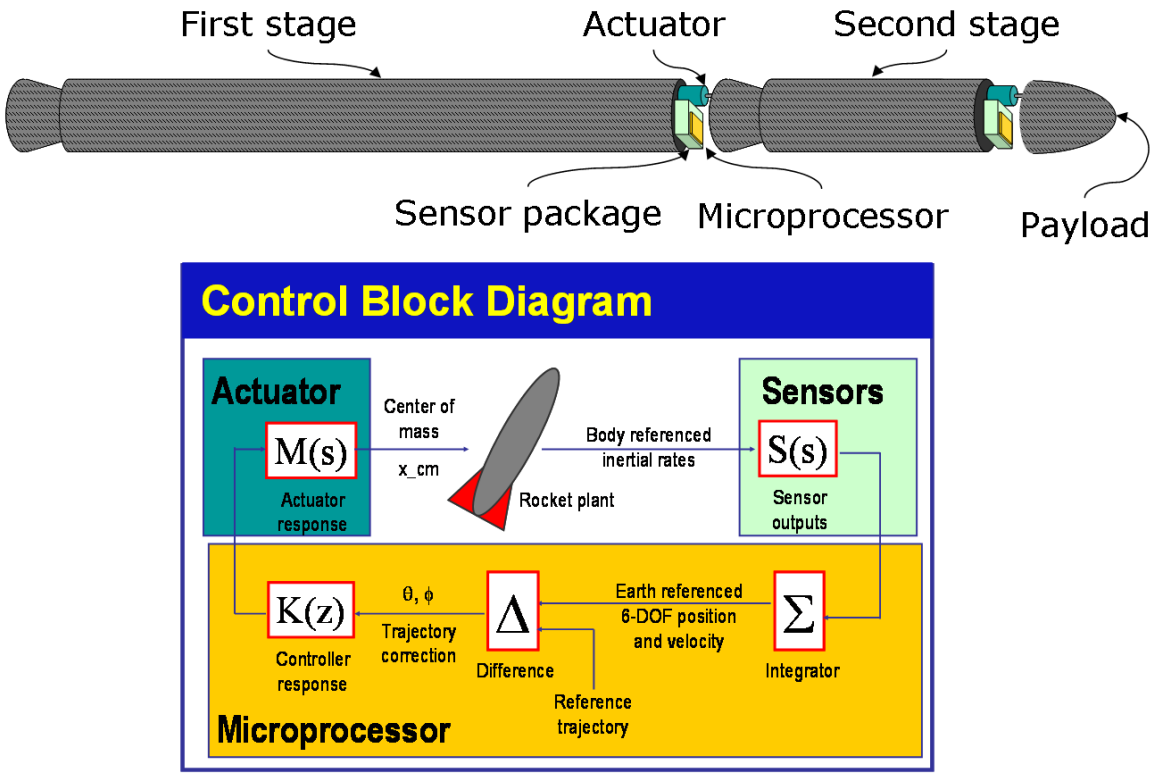

Figure 3.2: A feedback control system can be used to stabilize flight and follow a trajectory. The 6DOF state is sensed by a 6 axis IMU and input to a microprocessor. The current heading and trajectory error is calculated, and a control signal is output to an actuator. The actuator positions a mass or aims the nozzle to induce the appropriate torque on the rocket.

Figure 3.3: The original rocket containing GINA sensors and a video camera

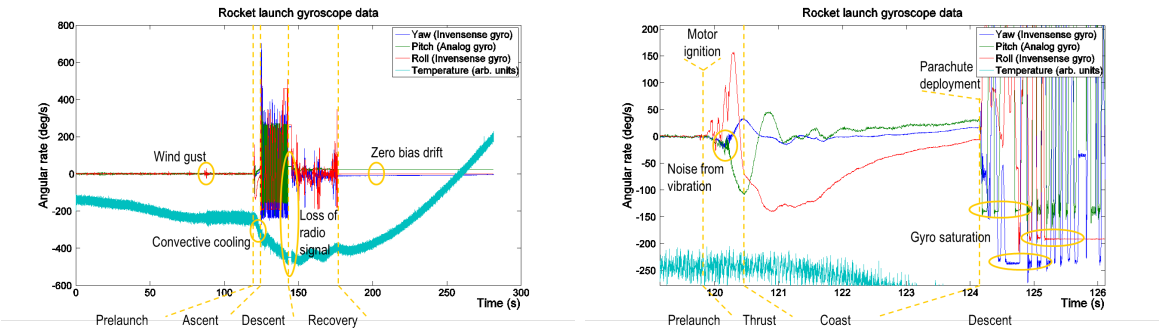

Figure 3.4: Gyroscope data from a flight shows many characteristics of rocket operation. This data can be integrated to yield an attitude state estimate for flight control.

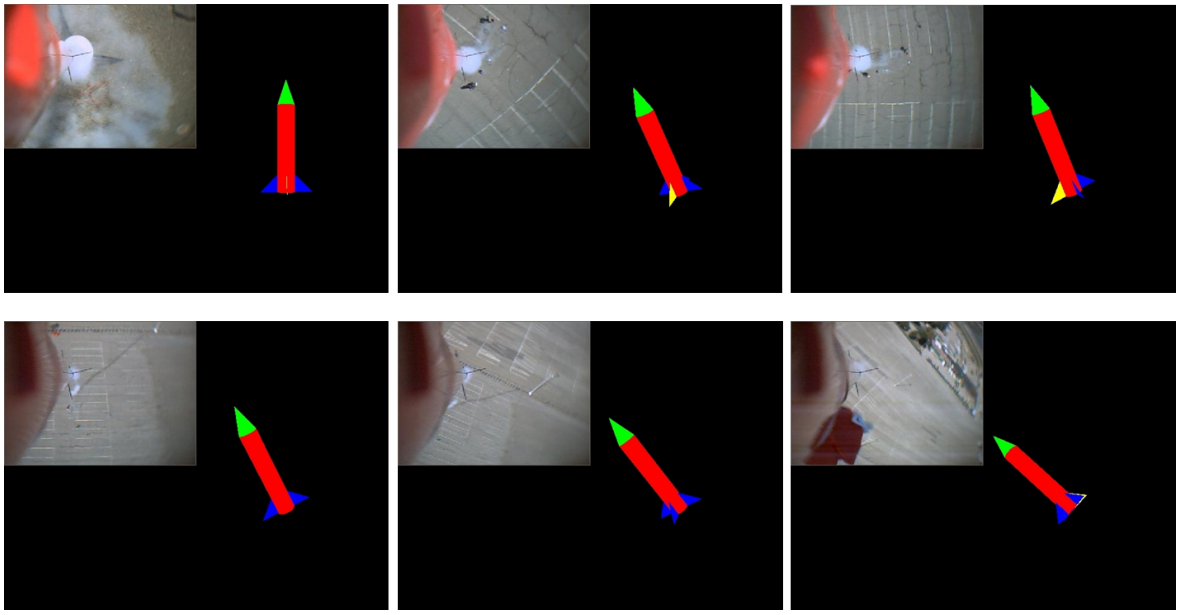

Figure 3.5: The original rocket containing GINA sensors (above) has evolved to the rocket shown on the bottom, using the WARPWING platform for 3 axis inertial attitude feedback control.

generate a 3DOF rocket attitude estimate. A computer generated model displaying this attitude estimate was superimposed on the video recording of the flight to give a indicator of the estimate fidelity. Screenshots of this comparison are displayed in figure 3.5; visual inspection determined that the estimate matched the actual state to a relatively high degree.

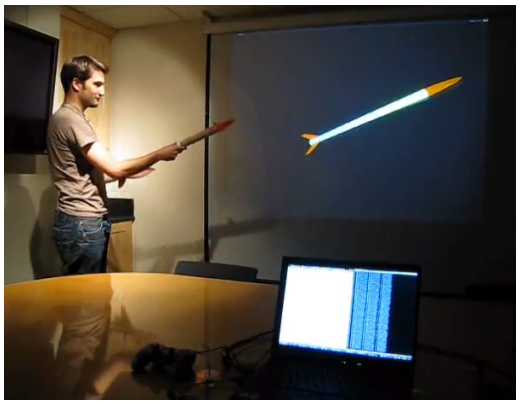

Figure 3.6: GINA sensor data transmitted wirelessly to a laptop basestation is integrated in realtime to give a 6DOF state estimate. The estimated attitude is projected alongside the physical system demonstrating excellent agreement.

Figure 3.7: An updated rocket contained a full GINA mote for 6 axis sensing, along with 3 axis attitude control with 4 channel flywheel and gimbaled nozzle actuation.

With this validation of the rocket control sensing subsystem, updated rockets were designed with a full GINA mote controller as presented in [24]. With new hardware giving higher quality 6DOF sensing, the state estimation was far more accurate, as shown in figure 3.6. Ultimately a feedback controlled rocket was built with flywheels and a gimbaled motor mount for 3 axis attitude control as seen in figure 3.7.

3.2 Autonomous helicopters

A rocket must control its body attitude in order to effect its navigation profile; a similar correspondance holds for helicopters. As such, the GINA controller and WARPWING platform were also adapted for use on a number of helicopter airframes.

3.2.1 Design goals

Autonomous helicopters are of particular use in indoor environments – in practice, MAVs must be controllable and agile. When operating in general environments, the MAV must be able to maneuver despite unknown air gusts or in tightly confined spaces. Furthermore, by definition the system is required to be small. In practice, the size of the system is dictated by the weight it needs to carry, so this constraint requires minimizing the total mass of the system. Lowering power draw is also beneficial, as it reduces the size of the battery necessary.

Ultimately, the end goal was to use these MAVs in clusters or swarms. To this end, the system should be cheap and reproducible. In particular, the system was designed with as many low cost off-the-shelf components as possible, with minimal custom design or complex assembly.

3.2.2 Helicopter platforms

There is a large market for commercial helicopter hardware in the toy industry – a number of companies design and sell remote controlled helicopters. Many of these devices use

standardized actuators with a simple wireless interface: a handheld remote generates control signals which are then mixed to drive the motors and servos responsible for helicopter motion. These form a very robust platform on which to conduct MAV research; by abstracting mechanical and aerodynamic system design to companies, effort can be focused on core autonomy research.

One vehicle selected as the platform on which to design the MAV system was the Walkera 4#3 remote controlled single main rotor - tail rotor (SMRTR) helicopter. It is a four channel electric helicopter with main and tail rotor throttle and elevator and aileron cyclic controls. It was selected based on the considerations above: it was the smallest available off the shelf fully controllable helicopter. It has a 180 mm diameter main rotor, and is driven by a 400 mAh lithium polymer (LiPo) battery. In its default configuration, it weighs a total of 45 g including the 10 g battery. A newer product by the same company is the Walkera 5g6 coaxial helicopter. It is slightly smaller with a 175 mm rotor diameter, but it too weighs 45 g and runs off the same battery. These two helicopters are shown in figure 3.8.

In order to investigate autonomy on such systems, the electronics of the helicopters were removed and replaced with the GINA board. The WARPWING code could then be configured in a variety of operating modes, allowing measurement of the inertial rates and control of the actuators. The helicopters outfitted with the custom electronics are displayed in figure 3.9.

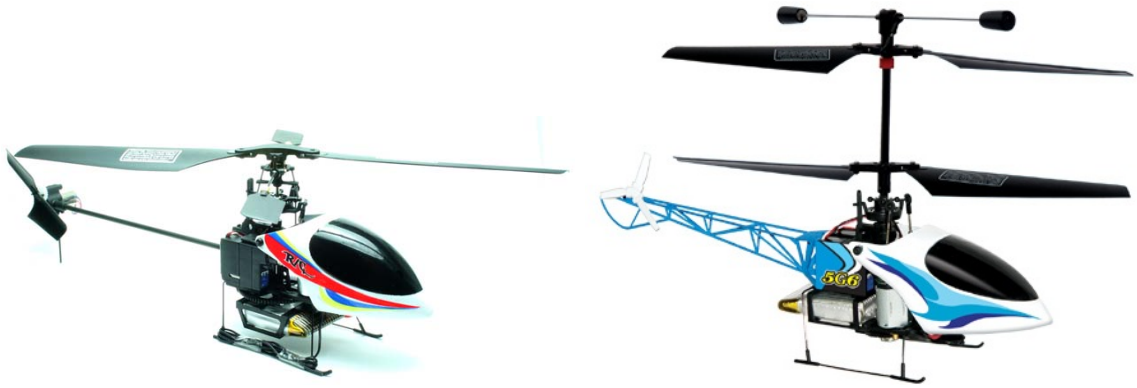

Figure 3.8: The Walkera helicopters selected as the platforms for this project are small, cheap, and commercially available 4 channel electric helicopters.

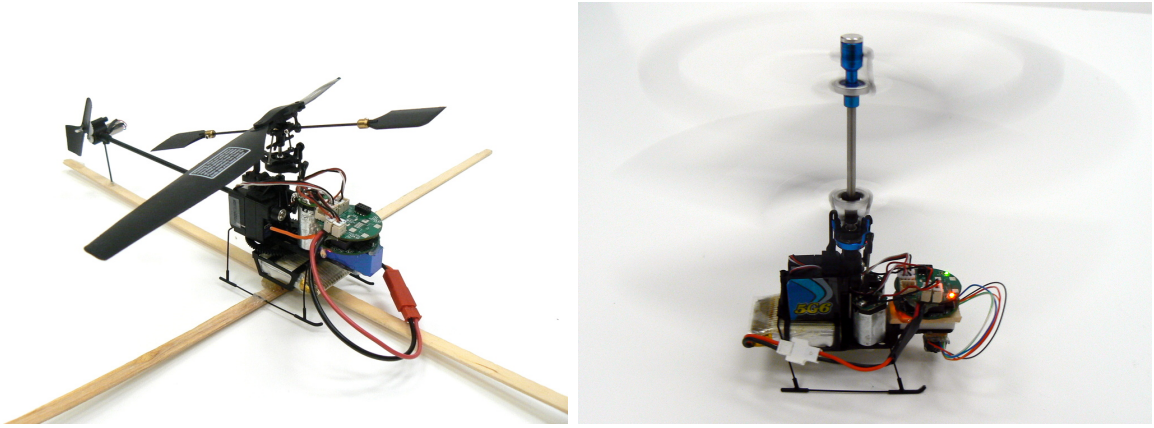

Figure 3.9: The off-the-shelf Walkera airframes are controlled by a GINA board running WARPWING code to investigate autonomous operation.

3.2.3 Extended Kalman Filter attitude state estimator

Stability and guidance control on MAVs requires state feedback; in practice the state of interest is the 6 DOF linear and angular position. As such, the first element in a flight controller is a state estimator that can generate 6 DOF pose information from a suite of sensors.

The state evolution of the helicopter attitude, that is, its pitch and roll angles, is a

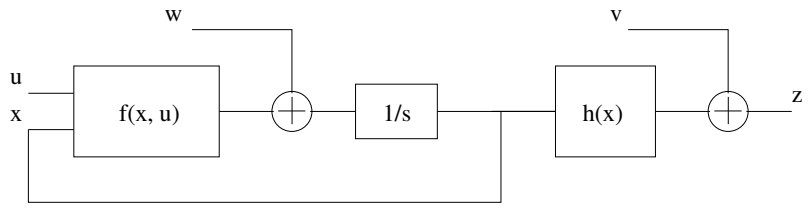

Figure 3.10: An extended Kalman filter can be used to estimate the state of a general noisy nonlinear system.

nonlinear function of the angles and angular rates. The gravity vector, measured by the accelerometer, gives a direct measurement of attitude state. Thus, an extended Kalman filter (EKF) as adapted from [16] was used to integrate the rates and use the state knowledge from the accelerometer in the presence of sensor noise to return the helicopter attitude.

A block diagram for a nonlinear system is shown in figure 3.10. A general system has variables:

- State $\mathbf{x}$
- Input $\mathbf{u}$
- Observation $\mathbf{z}$
- Gaussian state feedback noise $\mathbf{w} \sim \mathbf{N}(\mathbf{0}, \mathbf{W})$
- Gaussian measurement noise $\mathbf{v} \sim \mathbf{N}(\mathbf{0}, \mathbf{V})$,

with governing system equations:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{z} &= h(\mathbf{x}) + \mathbf{v}. \end{aligned}$$

In particular, for the state evolution of the attitude roll angle ϕ and pitch angle θ of a MAV:

$$\begin{aligned}
\mathbf{x} &= <\phi, \theta> \\
\mathbf{u} &= <p, q, r> \\
\mathbf{z} &= <a_x, a_y, a_z>
\end{aligned}$$

$$\begin{aligned}
f(\mathbf{x}, \mathbf{u}) &= \begin{bmatrix} p + q\sin\phi\tan\theta & + & r\cos\phi\tan\theta \\ q\cos\phi & + & r\cos\phi \end{bmatrix} \\
h(\mathbf{x}) &= \begin{bmatrix} \sin\theta \\ -\cos\theta\sin\phi \\ -\cos\theta\cos\phi \end{bmatrix}
\end{aligned}$$

The noise covariance matrices W, V are estimated from sensor data taken while the helicopter is stationary.

The state update is a continuous time process, so a continuous form of the EKF is used to update the a priori state estimate and state covariance matrix:

$$\begin{aligned}
\dot{\hat{\mathbf{x}}} &= f(\hat{\mathbf{x}}, \mathbf{u}) \\
\mathbf{A} &= \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \\
\dot{\mathbf{P}} &= \mathbf{AP} + \mathbf{PA^T} + \mathbf{W}
\end{aligned}$$

This method is also robust to dropped packets introducing variable time intervals between samples. The a posteriori update is then performed using

$$\begin{aligned}
\hat{\mathbf{z}} &= h(\hat{\mathbf{x}}) \\
\mathbf{C} &= \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \\
\mathbf{L} &= \mathbf{P}\mathbf{C}^{\mathbf{T}}(\mathbf{C}\mathbf{P}\mathbf{C}^{\mathbf{T}} + \mathbf{V})^{-\mathbf{1}} \\
\mathbf{P} &= (\mathbf{I} - \mathbf{L}\mathbf{C})\mathbf{P} \\
\hat{\mathbf{x}} &= \hat{\mathbf{x}} + \mathbf{L}(\mathbf{z} - \hat{\mathbf{z}})
\end{aligned}$$

3.2.4 Flight control attempts

To begin analyzing the dynamical system, the WARPWING software was configured to fly the Walkera SMRTR helicopter under manual control, and an expert pilot flew the system in a hover for over a minute. The sensor readings and control signals were logged over the course of the flight. Also recorded were the state estimate from the EKF running in real time on the sensor data received at the base station and the output of various low-pass filters (LPFs). Sensor data for the roll / sideways motion is shown in figures 3.11 and 3.12, and an EKF estimate of the roll angle is shown in figure 3.13. The motion of the main rotor feeding through to the helicopter body is clearly visible as a high amplitude oscillation in both plots, as well as beats in that oscillation due to mechanical resonances. Nonetheless, the output of the EKF was considerably less noisy, especially after application of a LPF.

With the yaw rate shown in figure 3.11, a single pole low-pass filter decreased the ampli-

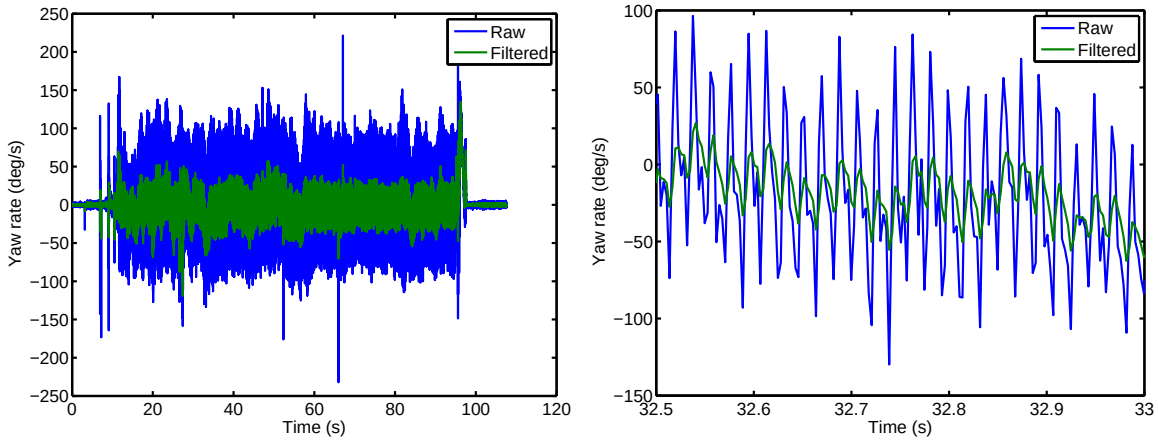

Figure 3.11: The yaw rate sensor output of GINA mounted on a helicopter driven to hover, before and after application of a low pass filter. The large scale yaw motion is recoverable from the filtered output, and so can be used to stabilize helicopter heading

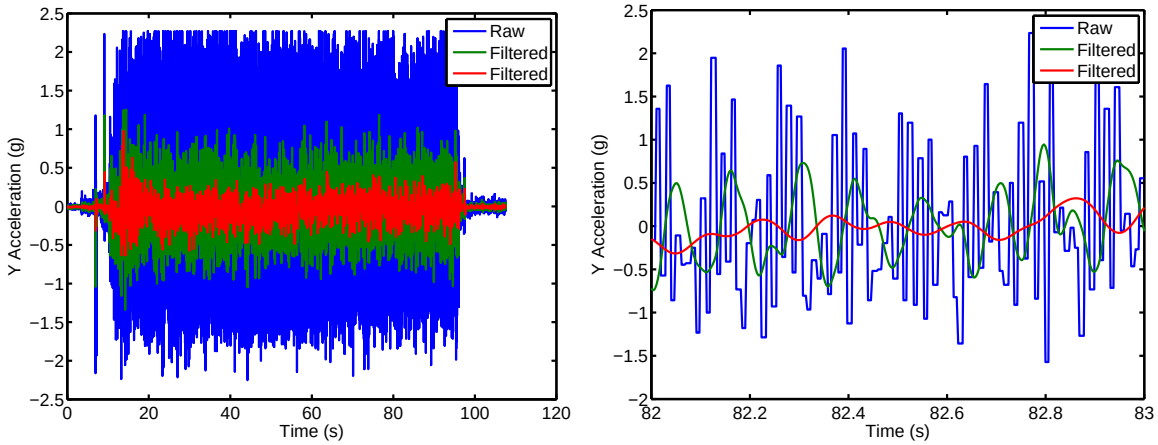

Figure 3.12: The linear acceleration measurement with application of a weak and strong low pass filter. With a weak LPF, the overall body motion is still swamped by the rotor signal, while a strong LPF introduces so much lag as to be useless for control.

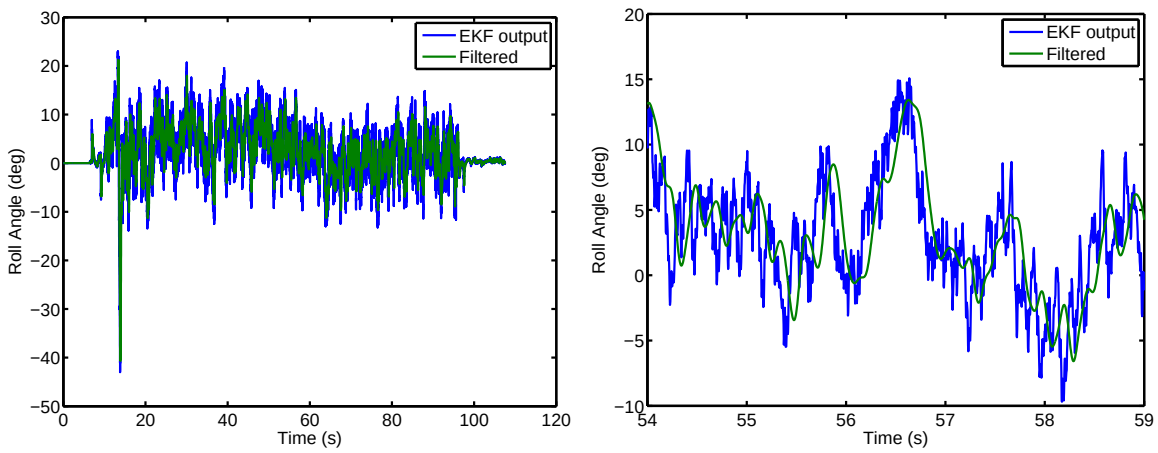

Figure 3.13: The roll angle output of the Kalman filter shows considerably lower impact of rotor motion, and can be filtered to provide a smooth signal.

tude of the rotor feedthrough such that the true yaw rate of the body was apparent without significant lag, despite the presence of the main rotor signal. A simple PID controller on this signal was applied to drive the tail rotor, zeroing the yaw rate to maintain heading. Despite the absence of a direct heading reference, the yaw rate PID controller was able to effectively eliminate large scale heading changes. This reduced the degree of the helicopter system by one, and eliminated one control input. What remained was to control the the linear motion of the helicopter.

At first the pitch angle / elevator cyclic and roll angle / aileron cyclic were considered complete independent subsystems. Applying PID loops on these systems did successfully generate restoring torques to hold the body attitude constant. However, regardless of the attitude setpoint, the helicopter failed to hold a position. In fact, looking at the trace of the roll angle in figure 3.14 during a sample controlled flight, the attitude setpoint was not

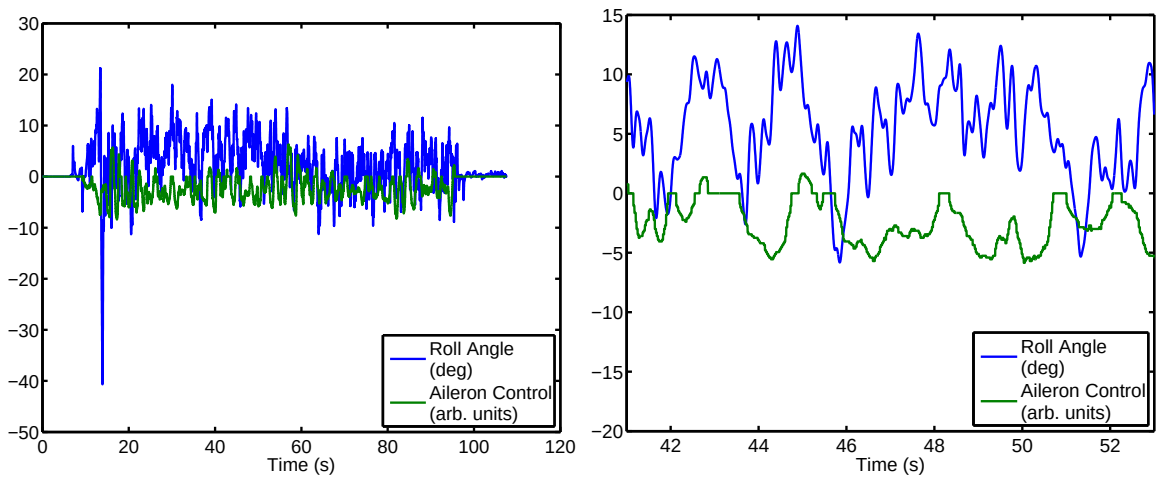

Figure 3.14: A trace of the roll angle during a controlled flight displays a time-varying attitude setpoint over the course of a hover. Comparison to the controller input reveals a complicated dependence on additional states.

constant over time. Furthermore, a comparison of the expert pilot's aileron control input to the body roll angle reveals that while often a negative control input increased the roll angle, the correlation was not that simple, and there were likely additional relationships to other states.

It turns out that the cyclic controls change the plane of rotation of the helicopter blades, but this does not get immediately translated to a body attitude. It does, however, generate the body acceleration. Thus, the cyclic controls would need to be a function of attitude, angular rate, and linear motion combined. The expert pilot confirmed that his visual feedback of the helicopter state included mainly the plane of the rotation of the blades and the overall body position and velocity.

What became apparent was that the overall linear motion of the helicopter was hard

to detect from the sensor data. Without an absolute position reference, the only way to get position and velocity information was by integrating the accelerometer measurements, and small offsets in zero bias quickly accumulated huge errors. The unfiltered accelerometer signals, however, also didn't yield much insight into the linear motion, with the vibrations of the body saturating any overall motion seen by the body. Low pass filters applied to the accelerometer output were only able to reduce the vibration signals slightly before the lag became significant, as seen in figure 3.12.

As the dynamics on the SMRTR Walkera 4#3 proved too difficult to stabilize, attention shifted to the inherently more stable coaxial design of the Walkera 5g6. While the SMRTR helicopter was unstable on the order of seconds, a properly trimmed coaxial helicopter could hold hover for several tens of seconds without feedback control. This stability necessarily came at the expense of controllability – as the helicopter maintained its state relative to the reference frame of its surrounding air, a wind gust or current would carry the helicopter along, requiring greater control effort to remain stationary with respect to ground.

A further benefit of the coaxial rotor configuration was greater independence of states. A SMRTR helicopter using its tail rotor to counter the main rotor torque generates a sideways force, which must be then countered by a permanent non-zero roll angle to generate the required opposite force from the main rotors. By removing the tail rotor, a coaxial helicopter ensures that to first order, each actuator controls only single axis of rotation.

These factors enabled a feedback controller to more robustly maintain hover in the MAV without human control. WARPWING platform code was written on a base station laptop to

download the inertial rates from the helicopter-mounted GINA board. The laptop computed the state estimate using the EKF described above, then evaluated the attitude error from the zero setpoint for hover. These values were fed into independent PID controllers, driving the actuators on each rotation axis. The PID coefficients were determined heuristically.

3.2.5 Additional position sensors

Though this controller was able to maintain the flight stability of the coaxial helicopter, it could still not be used for guidance or navigation – the helicopter would float along the slightest air currents, and the doubly integrated offset of the accelerometer caused the system to lose track of its absolute position in a very short time. In a typical room, this translated to on the order of a minute flight time before drift carried the helicopter into a wall.

In order for a MAV to maintain a true hover with respect to its environment, additional sensing to measure absolute position was necessary to reset sensor drift.

VICON

Many other robot research projects bypass the state estimation step in an on-board controller by using external infrastructure to inform the controller of its 6DOF state. A popular such option is the VICON motion capture system, which uses a collection of video cameras to view reflective dots on the objects of interest. A central system collects and processes all the images from the cameras to build a rigid body model with full state information.

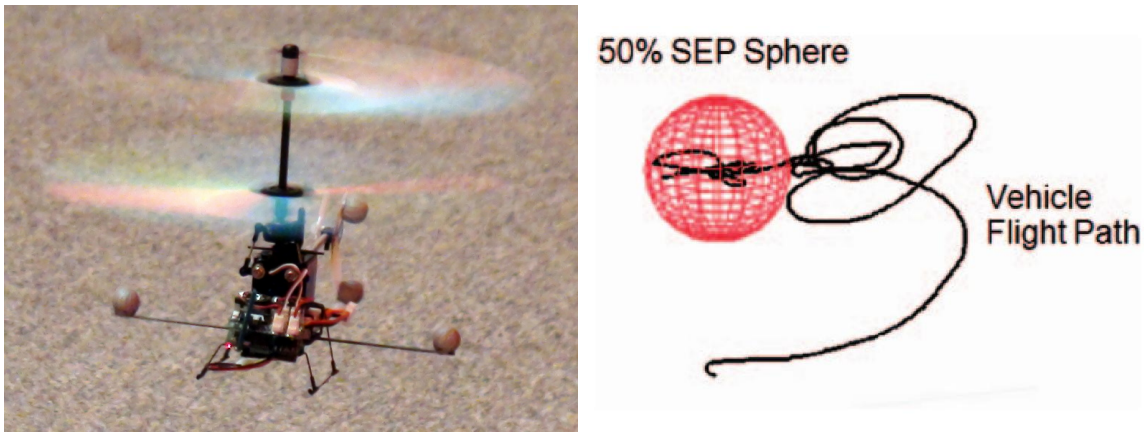

Figure 3.15: The Walkera 5g6 coaxial helicopter is shown using the WARPWING system with VICON position feedback to autonomously hover. The silver balls are the reflectors used by the vision based motion capture system. On the right, a sample flight path from [49] is plotted in space using the VICON data.

To validate the performance of the GINA based controller, VICON input was incorporated into the helicopter feedback control loop. The inner control loop remained unchanged – an error between the state estimate and state setpoint generated a control signal on the appropriate actuator, where again the axes were treated as independent. However, the state estimate was now determined by analyzing the VICON results.

The augmented control system was easily able to maintain hover (shown in figure 3.15), and therefore could effect navigation commands by following a moving setpoint. A quantification of the flight performance can be found in [49], where the helicopter was found to have a spherical error probability (SEP) of 0.142 m. That is, the helicopter would on average stay within a 15 cm radius of the desired setpoint, on the order of the volume of the helicopter itself.

Figure 3.16: The commercially available WiiMote contains a 300 mg "smart camera," which contains image processing hardware to return the coordinates of IR blobs in its field of view.

Beacon

The VICON system is large and expensive, and ultimately unreasonable to expect in real-world applications. To successfully transition the technology out of the lab, an alternate position sensor is necessary. Instead of using a collection of externally mounted cameras to view the robot system, a camera can be mounted on the system to view the environment. With a structured environment, then, processing the viewed image can yield a pose estimate.

Vision processing is typically computationally intensive. However, the WiiMote, the commercial console game controller seen in figure 3.16, contains an infrared (IR) camera with a built in image processing ASIC. The camera does threshholding and feature detection and returns the 2D coordinates of IR emitters in it's field of view. The device contains a simple serial interface that can be driven by the GINA mote.

This camera can be mounted on the helicopter and used to image an IR emitter in the environment. Such a beacon can be seen in figure 3.17, along with a visualization of the

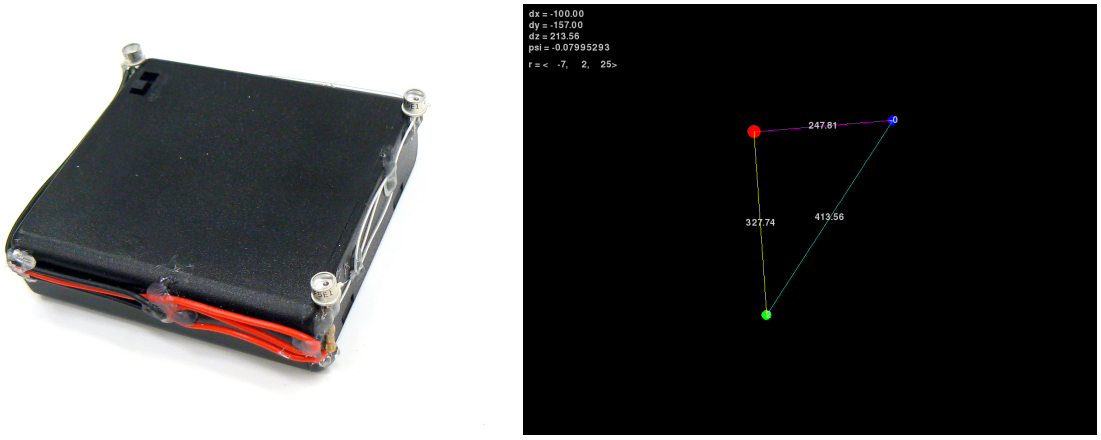

Figure 3.17: An infrared beacon consisting of three emitters with a known geometry can be used as a localization target. At right, the output of the WiiMote camera can be used to extract the camera pose.

returned data from the sensor. Given a known emitter configuration, the position of the camera relative to the beacon can be geometrically calculated [19]. This position estimate can then be fed into the feedback loop in much the same way as the VICON input, allowing for extended stable hover of the helicopter above the beacon. A helicopter platform with the GINA controller modified to accept the WiiMote input is shown in figure 3.18, and sample calculated geometry data is shown in figure 3.19.

3.3 Summary

The GINA robot controller as described in the previous chapter was developed and enhanced via the design of several MAV robot systems. Two modes of aerial actuation, chemical rockets and rotary wings, were employed to effect 3D motion. Though the design space and flight

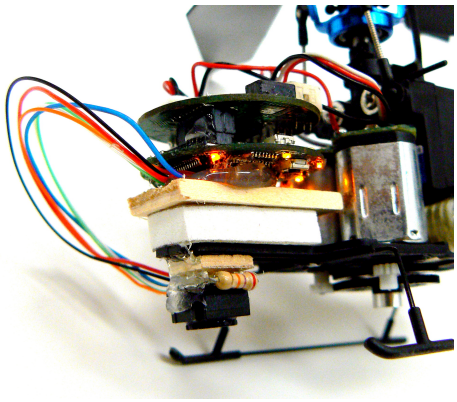

Figure 3.18: A downward pointing WiiMote camera is wired into a serial port on the GINA controller to enable the controller to hover over the IR beacon.

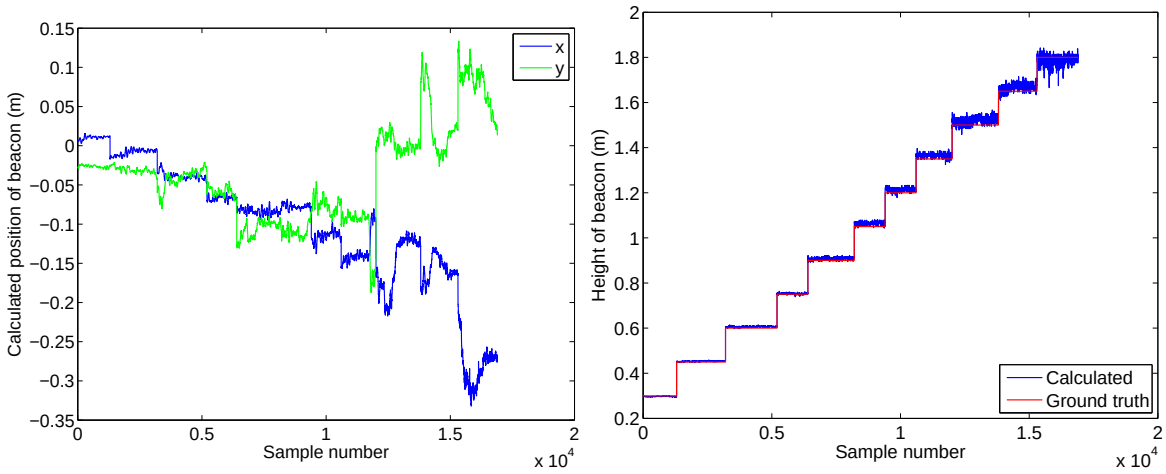

Figure 3.19: Calculated geometry from the WiiMote camera can be used as a position sensor to estimate the helicopter pose relative to the IR beacon.

envelopes of the two MAV modes were drastically different, they shared a number of common requirements, addressed by the versatile WARPWING platform.

Implementation and analysis of these robot systems clearly demonstrated the need for accurate self-contained localization, a concern which is still an open research problem. With the overarching design constraint of minimizing size and weight, the localization question often translates to a search for more accurate microscale sensing, clever applications of alternative sensors, and intelligent sensor fusion to build robust state estimates at small size and mass scales.

Chapter 4

Wireless Protocol Modifications

Networked robots necessarily need an underlying network infrastructure for communication; a full system cannot be evaluated without considering its impacts as well. As a swarm of robots is essentially a wireless sensor network (WSN) with added mobility, the analysis of the underlying communications can be built upon the existing WSN infrastructure. Furthermore, enhancements to conventional stationary networks correlate to improved functionality of mobile networks as well.

4.1 Background

4.1.1 IEEE 802.15.4

A full communications stack includes many levels of abstraction, allowing for modular reuse and interoperability between potentially disparate applications. In practice, it is the lowest

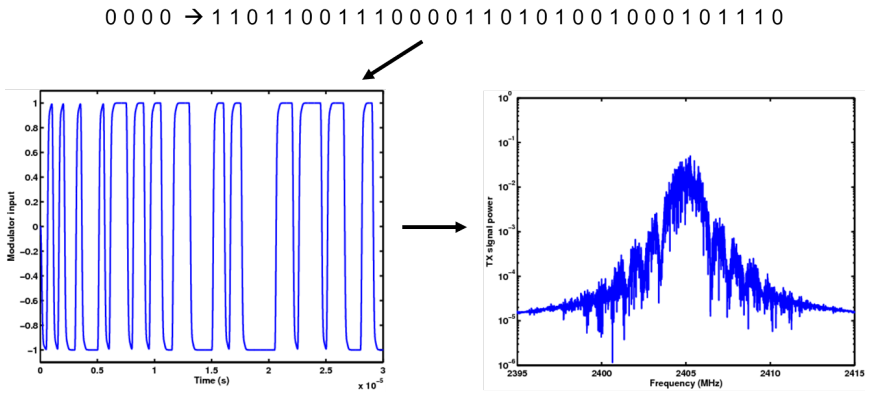

Figure 4.1: In the IEEE 802.15.4 standard, 4 data bits get encoded to a 32 chip long codeword, which is modulated using OQPSK-HSS onto an ISM band carrier.

levels of the stack, that is the physical (PHY) and media access control (MAC) layers, that define a network. The IEEE 802.15.4 specification [22] is the de facto standard PHY and MAC protocol for WSNs.

The IEEE 802.15.4 allows a number of frequency bands; typically the 2.4 GHz industrial, scientific and medical (ISM) band is used for operation. Data to be communicated gets encoded using a set of 16 balanced pseudo-orthogonal codewords, such that 4 data bits gets transmitted as 32 signal chips. These chips are modulated at 2 Mchip/s using offset quadrature phase shift keying with half sine pulse shaping (OQPSK-HSS) onto the carrier wave. The 1/8th rate coding therefore implies a data throughput of 250 kb/s. The carrier wave can be at one of 16 frequencies (channels) in the ISM band. This process is diagrammed in figure 4.1.

4.1.2 Time synchronized channel hopping

The IEEE 802.15.4 standard does not specify when or on what channel to communicate; this left up to the network designer to implement. The simplest solution is to operate on a single channel, with nodes free to transmit at will, and set to receive otherwise. However, this method is extremely power hungry, and is neither scalable nor robust.

By precisely scheduling communication between nodes throughout the network as described in [39], mote hardware can be powered down while not scheduled to communicate, thus saving considerable energy throughout the network. In addition, the schedule that allocates time slots for communications can also allocate frequency channels, and ensure that successive packets between the same pair of nodes happen on different channels. These channel hopping schemes that exploit frequency diversity prove robust to sources of communication failure, such as narrowband interfence and multipath fading [44]. More elaborate schemes of frequency hopping can gain even greater benefit; regardless, all methods of channel hopping improve nearly every metric of WSN performance [46].

Together, these MAC methods are known as a time synchronized channel hopping (TSCH) protocol; this was standardized in IEEE 802.15.4e in Spring 2012 [23].

4.2 Variable data rate adaptation and control

The IEEE 802.15.4 standard provides robust error correction by encoding groups of 4 bits into symbols 32 chips long, allowing for relatively high chip errors in the presence of low signal-to-

Table 4.1: Energy savings comparison

Data Rate	Additional SNR Required	PA Scaling Savings	Variable Rate Savings
250kb/s	0 dB	0%	0%
500kb/s	2.6 dB	16%	48%
1000kb/s	5.6 dB	31%	72%
2000kb/s	14.1 dB	55%	84%

noise ratio (SNR). In typical 802.15.4 networks, however, most links have much higher SNR than required for the default coding scheme. This implies excessive redundancy, resulting in inefficient communication. By optimizing the coding rate, a more efficient network could reduce power consumption and increase lifetime.

An alternate way of exploiting excess SNR to reduce power consumption could be to lower the output power of the power amplifier (PA) of the radio transmitter. However, the efficiency of a PA decreases quickly with decreasing output power; for example, an 8x reduction in transmitted power results only in a 40% savings in consumed power. A look at table 4.1 indicates that when power savings for a single packet are calculated based on PA efficiencies, varying the data rate has much greater energy saving potential.

Dynamic Rate Adaptation and Control for Energy Reduction (DRACER) [26] is an addition to the IEEE 802.15.4 PHY specification adding 500kb/s, 1000kb/s and 2000kb/s data rates to the existing 250kb/s along with a MAC extension to select the appropriate data rate. A biorthogonal coding scheme preserves the hardware simplicity captured in the original standard while providing robust error performance. Using link level acknowledgments

with the MAC extension, the data rate of the next packet is based on the SNR of the most recent packet. When a packet failure occurs, a simple backoff scheme is used to restore communication. This allows for data rate to be highly adaptable and managed on a per-link basis without the input of a centralized manager.

4.2.1 Related Work

The prior work in rate adaptation is helpful in understanding the problems faced by variable rate systems, choosing a coding scheme, and in optimizing network performance.

Wireless networks can optimize throughput across channel conditions by communicating with a variable coding rate, reducing latency, congestion, and network energy consumption. Studies in ad hoc wireless sensor networks have shown that single rate communication reduces network efficiency [13], yet the IEEE 802.15.4 protocol provides for only a single data rate. The ultra-wideband extension IEEE 802.15.4a as well as other standards including IEEE 802.11 and Bluetooth include many data rate options, and 802.15.4 radios exist that provide non-standard higher data rate options, but none provide a mechanism for choosing the optimal rate [22, 37, 4].

The selection of data rate in IEEE 802.11, Bluetooth and other networks has seen a significant amount of academic attention, and the focus has been on reduced latency or increased throughput [10, 50, 25, 47]. Reducing the network energy consumption is the most important metric for wireless sensor networks even though latency and throughput can be

important in some cases [15]. Many of the studies use minimal to no empirical data. Energy has been considered in point to point links, but data from physical networks is not considered and energy is not the dominant metric [27, 3]. Variable rate wireless sensor networks have been considered for improving routing algorithms with throughput and congestion as the motivation [13, 21].

4.2.2 Experimental Wireless Network Setup and Measurements

Setup

Data collected from a Dust Networks test network deployed in an industrial environment was used to analyze the performance of a typical 802.15.4 wireless network [12]. The nodes consisted of a microcontroller, CC2420 radio [9], and +15dBm power amplifier. The network was deployed in a three story building housing the offices and production equipment for a printing factory. A single network manager (data sink node) was placed on the 3rd floor, and the 44 nodes were connected to the manager using a time synchronized, frequency hopping, multi-hop, mesh network[39]. Each node recorded data regarding the conditions on each link to its neighbors at the 16 carrier frequencies specified by the 802.15.4 standard. This information was collected every 15 minutes from each node and included packet error rates (PER), received signal strength (RSS), and a link quality indicator (LQI). Each packet was 133B long including the preamble, and about 9 million successful packet transmissions and receptions occurred in the network during the 26 day study period. For this discussion

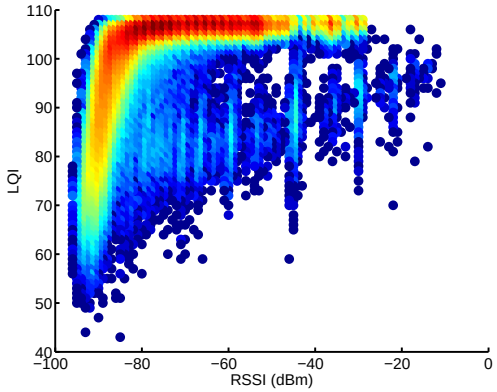

Figure 4.2: The number of links at a given RSSI and LQI is shown in this density plot. Red points indicate numbers on the order of 10^4 while dark blue points indicate numbers on the order of 10^0.

a path-channel will refer to a data connection between two nodes at a particular carrier frequency, and a link represents a path-channel with conditions averaged over a particular 15 minute interval.

Measurements

Every successful packet received by the CC2420 returns two measures of path conditions. The RSS indicator (RSSI) measures the signal power P_s and is given in dBm. The link quality indicator (LQI) is a less specifically defined number, but is often thought of as a measure of chip error rate (CER). For the CC2420, which uses soft decisions, the LQI is more precisely a measure of SNR (P_s/P_n). A density plot showing the number of links returning specific RSSI and LQI values is shown in figure 4.2.

This plot shows that the vast majority of links lie on a single curve, monotonic in LQI with respect to RSSI. The width of this curve shows the variability in noise floor across time

and channels and differing noise performance across devices. Off of this main curve, there are additional curves containing a number of links of high RSSI with notably lower LQI. These curves indicate the presence of in-band interferers artificially increasing the noise power P_n, lowering the SNR. While the signal power itself remains high, there is greater probability of error in these regions. It is useful to note though that there are far fewer links in these high-noise curves; the deployed system contained only very sporadic interference.

The 802.15.4 standard does not specify how the LQI is calculated, but the CC2420 datasheet provides information to map LQI to SNR. The datasheet specifies that soft decisions are used in the correlator, a low-IF receiver architecture is used, and that the average correlator output for 8 symbols from the preamble are averaged to calculate LQI. It is known that 3 bit quantization for soft decisions is standard design practice, and the maximum LQI output in the CC2420 case is $LQI_{max} = 8 \cdot ((2^3) - 1)/16 = 112$. This value agrees with the datasheet and experimental data. Once the method for the calculation of LQI is known, a simulation can be run using a noncoherent FSK detector and many preambles while varying SNR. The resulting data is a function that maps SNR to LQI, and inverting this function provides the desired map of LQI to SNR shown in figure 4.3. Due to dynamic range and bandwidth limits in the receiver, higher LQIs represent significantly more SNR. Due to the integer granularity on the LQI, the SNR change from one high LQI value to the next can be quite large.

An important performance metric in wireless sensor networks is the packet error rate (PER), which is directly related to SNR [26]. Thus, it makes sense when considering link

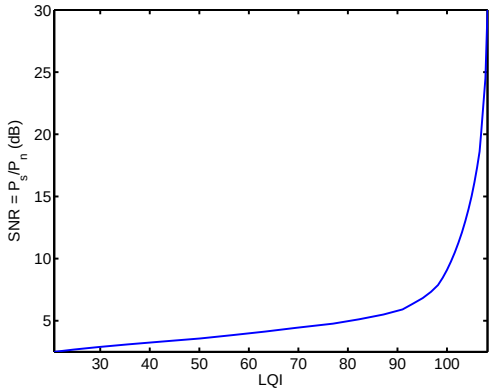

Figure 4.3: Simulation results returning LQI across SNR can be inverted to predict SNR given LQI, as shown in this plot. At high SNR, the integer-level granularity of LQI results in lower SNR resolution.

performance to examine the LQI as opposed to RSSI. In fact, looking at a slice across the link density plot in figure 4.2 at a constant RSSI, the link SNR is observed to be bimodal. Figure 4.4 shows the presence of a high-noise interferer resulting in low SNR despite high signal power. Estimating performance from RSSI for links in the lower hump would then result in higher than expected error rates. Again, there are considerably fewer links exhibiting this interference in the network considered here, however, other networks may be more heavily interference dominated.

Network Analysis

An aggregate indication of link quality in the network can be made by looking at the complimentary cumulative distribution function (CDF) of the SNR across all links in the network, as shown in figure 4.5. It is clear that most of the links contain significant excess SNR, which can be exploited to send data at higher bit rates using the DRACER protocol described in

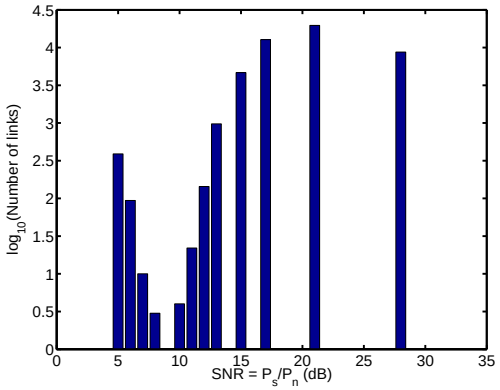

Figure 4.4: A histogram of links across SNR at constant RSSI = -62 dBm shows bimodality derived from interference. The spacing towards high SNR is an artifact of the reduced resolution at high LQI.

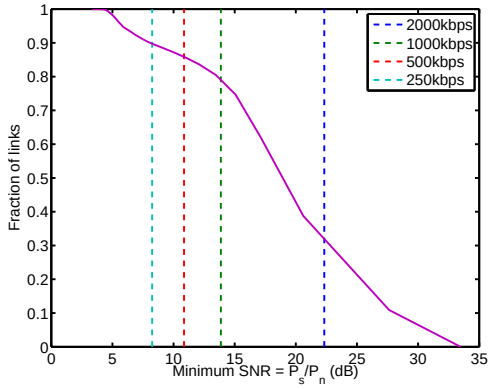

Figure 4.5: The complimentary CDF shows the fraction of links in the network having SNR greater than indicated. The vertical dashed lines show the minimum SNR threshold given legacy 250kb/s 802.15.4 coding, 500 and 1000kb/s biorthogonal coding, and 2000kb/s raw schemes.

the next section. The minimum SNR required to meet the 802.15.4 PER specification for the various data rates and the fraction of links exceeding that SNR are shown in table 4.2.

The link quality can also be correlated across time. The packet error rate is measured for each link in the network. A CDF of the average PER in the 15 minute time interval following an error free time interval is shown in figure 4.6. This indicates that channels are mostly

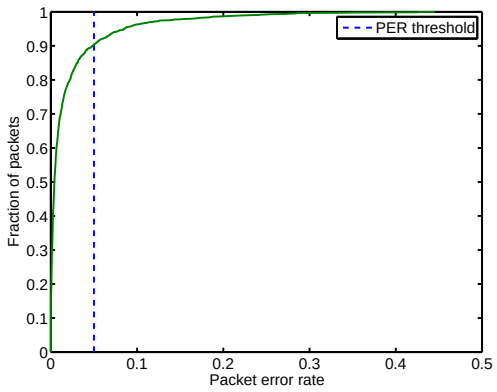

Figure 4.6: The CDF of average PER across path-channels in the network following an error free time interval. In 90% of the path-channels observed, an error free interval was followed by another interval in which the PER was below that required by the standard.

stable over time; 90% of the time, an error free interval is followed by another successful interval. The data collected in the network only resolves down to 15 minute granularity; with packets sent at faster rates, it stands to expect that a packet transmitted following a successful transmission will also be successful.

4.2.3 Implementation of variable data rate coding

The implementation of the DRACER protocol can be accommodated in the PHY and MAC layers. The PHY changes include the inclusion of additional code sets and automatic detection of the incoming data rate at the receiver. The MAC can automatically select data rate based on RSSI or LQI information.

The PHY must recognize the intended data rate of the incoming packet so that the appropriate detection scheme can be used. The legacy standard specifies a synchronization header to start each packet consisting of a preamble followed by a two symbol start frame

delimiter (SFD). The same legacy preamble remains unchanged enabling the receiver to always start in legacy mode. Three additional SFDs are selected to denote the three additional data rates. When the receiver detects a particular SFD, it changes the reference code set and code length for the relevant data rate. Because the signaling rate is slow compared to the speed of electronics, changing the code set between symbols is not a concern.

Automatic rate selection reduces network overhead. Acknowledgments (ACKs) must be enabled so that both radios in a link are able to measure channel conditions for each packet sent. Initial communication occurs at the legacy rate, and either the RSSI or LQI is used as an estimate of channel capacity. LQI provides better performance in the presence of interference, but the conversion from LQI to available capacity is not always known. It was shown that over time scales of minutes, the typical channel characteristics remain similar with high probability. Therefore the best predictor of the current channel is simply the conditions seen by the last packet. Regardless of current data rate, the rate for the next packet is selected based on the most recent RSSI/LQI at that frequency and for that point to point link. The thresholds for selecting the appropriate data rate can be derived from the network model, as in the following section.

Failure to receive an ACK for the packet results in its retransmission, and there are a variety of options for retransmitting a packet sent at a higher data rate. The transmitter could continue to send at the higher data rate until it is successful, with no reduction in rate (backoff scheme A in table 4.4 below). Alternately, upon a packet failure the transmitter can drop down one data rate and transmit at that rate until receiving an ACK (backoff scheme

Table 4.2: Feasibility of higher data rate communication in an empirical WSN installation

Bit rate	Coding scheme	Minimum SNR required	Percent of links meeting SNR_{min}
250 kb/s	802.15.4	8.25	90%
500 kb/s	Biorthogonal	10.86	86%
1000 kb/s	Biorthogonal	13.87	79%
2000 kb/s	Raw	22.33	30%

B). The transmitter could also drop to the legacy 802.15.4 coding upon a single packet failure (backoff scheme C). Finally, the transmitter could reduce the transmit rate one step for each packet failure until it hits the legacy 802.15.4 rate (backoff scheme D). Different network conditions may require different strategies: in networks primarily dominated by multipath fading and noise, a gradual backoff will provide the best performance, while in networks primarily dominated by interference from other RF devices, maintaining higher data rates (and therefore shorter packets) can reduce the number of collisions and improve performance. In the following performance evaluation, different back off schemes are shown to provide similar performance.

4.2.4 Performance

In the network examined here, packets are generated at a low enough rate that network congestion is not an issue. As such, impacts on latency and throughput cannot be investigated. However, the overall power consumption of the network can be extracted from simulation and will be used to demonstrate the effectiveness of the variable data rate scheme.

Table 4.3: Component power and time requirements

	Microprocessor	Radio RX	Radio TX
Power consumption	9 mW	66 mW	150 mW

	Mote on-time in ms			
Event	250kb/s	500kb/s	1000kb/s	2000kb/s
Packet TX	4.26	2.19	1.18	0.67
Successful Packet RX	5.26	3.19	2.18	1.67
Failed Packet RX	2.00	2.00	2.00	2.00
ACK TX	0.83	0.48	0.32	0.24
Successful ACK RX	0.96	0.61	0.45	0.37
Failed ACK RX	0.25	0.25	0.25	0.25

To calculate the network energy, some network parameters must be defined. The packet transmissions are globally scheduled in the network protocol used here, and as such, the receiver will be listening every time the transmitter attempts to send a packet. If the receiver successfully receives the packet, it will respond with an ACK. The transmitter listens for an ACK after every TX attempt, and it is assumed it will hear one if it is generated. Since the packet success rate was found to be correlated over much longer time scales, it is highly unlikely that an ACK will not be heard. There are a fixed number of packets that need to be sent over each link, and each will be retransmitted until they have all gone through. Finally, the microprocessor will be running during any TX or RX, with an additional 10% overhead for required synchronization and logic. The time each component needs to be on for all network events is shown in table 4.3 which can then be applied to derive energies.

Legacy performance

The data returned from the network gives us the number of TX attempts and the number of ACKs received. Using the assumptions listed above, the amount of time spent in TX and RX for each node can be calculated, and the total energy of all nodes other than the base station can be extracted. This value is 15.0 kJ which is an average power of 135μW per node.

Model setup

To evaluate the variable data rate schemes proposed above, the network needs to be simulated. As mentioned above, a fixed number of packets must be sent over each link, and will be retransmitted upon failure. This number is given by the number of successful ACKs received over each link from the network data. The total packet error rate of each link pe_{tot} is also taken from the network data. This error rate can be divided into two components, errors due to interference and errors due to insufficient SNR. That is, Pr{Packet success} = Pr{No interference} * Pr{Sufficient SNR}, or

$$1 - pe_{tot} = (1 - pe_i)(1 - pe_{SNR}). \tag{4.1}$$

The PER due to SNR pe_{SNR} can be calculated using the LQI taken from the network data [26], and thus the interference based PER pe_i can be inferred. This can be combined with the SNR-based PER calculated for the alternate coding schemes to yield the total PER at each of the four bit rates. This value can then be applied to packet transmission as a

Table 4.4: Percent energy savings

Backoff scheme	LQI thresholding	RSSI thresholding
A	33.8%	23.8%
B	38.7%	24.4%
C	40.1%	38.9%
D	41.0%	39.6%

Bernoulli process, and the expected number of TX failures can be estimated. This model for the network using only 802.15.4 coding necessarily yields the same packet behavior as the true network.

Performance of proposed scheme

We can now use this model to evaluate the schemes described above. The intended data rate for each link is computed from either the LQI or RSSI in the previous link, thresholded to predefined limits for each of 250, 500, 1000, and 2000 kb/s. Considering the packet success behavior as a Bernoulli trial with success probability calculated from the LQI and interference based packet success rate of the current link, the number of packet success and failures at each of the data rates can be computed for the different back off schemes. The thresholds were set to the minimum energy configuration in each case. The overall network energy savings of the data rate schemes with various thresholding and backoff options are presented in table 4.4.

In particular, scheme D with LQI thresholding was able to achieve a total 41% energy savings against the legacy system across the network:

- If the previous time period's LQI was 99 or less, only attempt to send packets at the legacy 802.15.4 rate.
- If the previous time period's LQI was between 100 and 103, attempt to send one packet at 500kb/s. If it fails, send the remaining packets at the legacy 802.15.4 rate.
- If the previous time period's LQI was 104 or above, attempt to send one packet at 1000kb/s. If it fails, try once more at 500kb/s. If both fail, then send the remaining packets at the legacy 802.15.4 rate.

The 2000 kb/s raw rate was never used in this simulation due to its high SNR requirement. Though there are links with an LQI of 108, sufficiently ample for raw data transmission, the decision is based on the previous time interval, and links with an LQI of 108 rarely stayed as such during the following time interval. It is probable that with a finer time resolution, packets transmitted at 2000 kb/s would be more successful for additional savings.

The results of the same simulation can be used to determine the energy savings per node, and this result is plotted against that node's packet throughput in figure 4.7. It is clear to see that in general, nodes that handle more traffic save more energy and therefore contribute greater savings to the overall network. Thus, we would expect that as a wireless network scaled up in size and data traffic, DRACER would contribute greater energy savings, allowing for networks robust to continued growth.

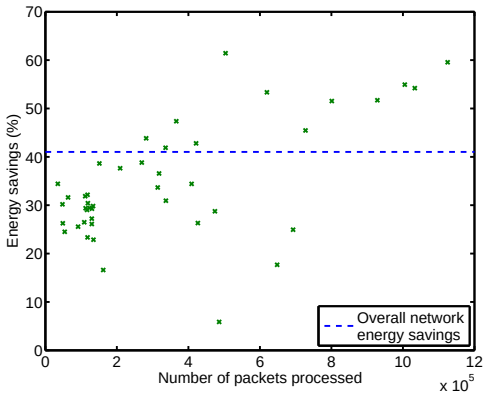

Figure 4.7: A plot of each node's usage versus its energy savings show that in general, the more a node is used, the greater potential energy it can save from variable bit rate.

Results

By using adaptive variable rate signaling in the physical layer of a wireless sensor network protocol, significant power can be saved during network operations. The proposed variable rate scheme DRACER showed an estimated 40% savings in average network power based on total energy consumption from a model built from reference network data of a physical 44 node network. DRACER also reduced contention in the network between competing users by minimizing the occupied channel time, and could be used to lower latency and increase the network data capacity.

4.3 Crystal-free communication

Though wireless sensors networks (WSNs) are becoming increasingly widespread, they are yet far from being ubiquitous. One factor in this is that wireless sensor nodes are not cheap,

requiring a full hardware solution to be effective: current radios all require a quartz crystal to provide an accurate frequency reference, increasing costs due to component and PCB requirements.

There has been some research that has produced crystal-free RF communication (e.g. [14]), however, such communication is typically broadband and not robust to interference. The IEEE 802.15.4 standard is meant to be scalable, low power, and robust. This work demostrates the feasibility of crystal-free 802.15.4 compliant communication, the first step in moving towards an integrated single-chip wireless sensor mote.

4.3.1 Perforance specifications

802.15.4 specifications

The 802.15.4 standard specifies stringent performance limits on compliant systems. The hardware must have an absolute clock accuracy of ±40 ppm, ensuring tight bounds on the transmitter carrier frequency and allowing the receiver to use a narrow channel-select filter to minimize out-of-band noise power. The standard defines sensitivity to be measured at 1% packet error rate (PER) with 26 byte packets, corresponding to a symbol error rate (SER) of 1.9×10^{-4} [26]. In practice, this translates to a minimum acceptable signal to noise ratio (SNR) at the receiver to fulfill the 802.15.4 specifications.

However, as seen above, the vast majority of links in real-world WSNs have significantly better than the minimum SNR required to to hit the acceptable PER, and so other re-

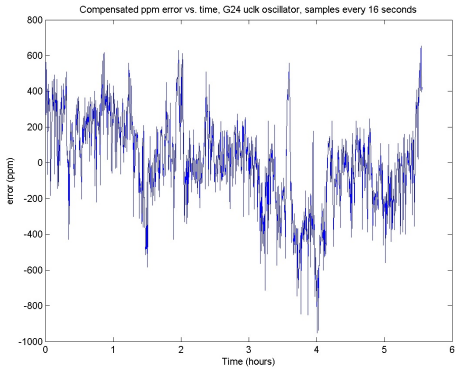

Figure 4.8: The frequency of an on-chip relaxation oscillator drifts mostly between ±600 ppm.

quirements, in particular the clock accuracy, may be relaxed while still maintaining network performance.

Oscillator performance

In order to achieve the 40 ppm frequency accuracy required by the 802.15.4 specification, a high quality quartz crystal (with typical accuracy on the order of ones of ppm) is required as a timing reference. An oscillator fabricated in the silicon would perform far worse: most on-chip oscillators only guarantee accuracy on the order of one percent. It is possible to design oscillators with slightly better performance: a digitally controlled relaxation oscillator was designed in a 0.18 μm CMOS process; the measured frequency accuracy over time of the actual hardware is displayed in figure 4.8. [1] With focused design, then, an on-chip frequency reference can be made to maintain ±1000 ppm accuracy.

[1] Image and data courtesy Mark Lemkin, used with permission.

4.3.2 Channel select filtering

A standard receive chain involves mixing the received RF signal down to a specified intermediate frequency (IF) (which can be at DC for a direct-conversion receiver). The downconverted signal is then passed through a channel-select filter then demodulated. Typically, an accurate frequency reference drives the local oscillator (LO) used in the mixer. This takes a received signal from a given RF carrier frequency down to the expected IF. However, if the LO is not tuned to the same frequency as the transmitter, the downconverted signal will be offset in frequency from the expected IF by the same amount as the difference between the LO and RF carrier frequencies.

The receiver includes the channel-select filter to improve sensitivity by attenuating out-of-band noise power. The filter needs to be wide enough to contain the signal bandwidth with sufficient margin to accomodate clock offset. For an 802.15.4 compliant system, the filter bandwidth (at baseband) needs to be at least (2 MHz)/2 + (40 ppm $\times$ 2.48 GHz) $\approx$ 1.1 MHz. However, to accomodate clock offsets of up to 1000 ppm, the filter bandwidth needs to be expanded considerably lest the signal power itself be filtered out. This serves to increase the noise bandwidth and therefore the out-of-band noise power input to the demodulator, thus increasing the required in-band SNR for a given performance.

A generated 802.15.4 transmission at RF was input in simulation to a receiver consisting of a mixer down to baseband (BB, zero IF), then a low pass channel-select filter, followed by a digital demodulator to recover the data. The LO of the reciever was allowed to vary up to

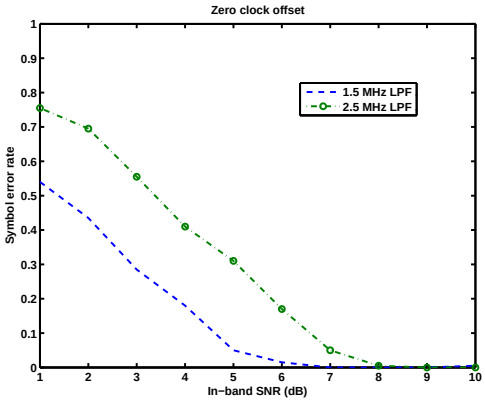

Figure 4.9: The effect of increasing the filter bandwidth is to increase the out-of-band noise power admitted to the receiver, increasing the required in-band SNR to maintain performance.

1000 ppm offset from the actual carrier frequency, and a range of noise power (in-band SNR values) was simulated.

Figure 4.9 compares the effect of varying the channel-select filter bandwidth. As expected, at zero clock offset, the SER drops with increasing SNR, becoming negligible above some threshold. That SNR threshold rises if the filter bandwidth is increased, as more out-of-band noise power is admitted to the demodulator. However, as seen in figure 4.10, at 1000 ppm clock offset between receiver and transmitter, no amount of SNR returns any usable signal with a 1.5 MHz filter: the entire signal has been filtered out and only noise is captured. Expanding the filter to 2.5 MHz bandwidth allows some of the signal through, though there is enough noise power that the SER never really drops to zero.

The full simulation results with a 2.5 MHz filter, shown in figure 4.11, demonstrate the ability of a receiver to trade SNR for frequency offset tolerance, successfully decoding 802.15.4 transmissions under conditions that can be expected from replacing a crystal with

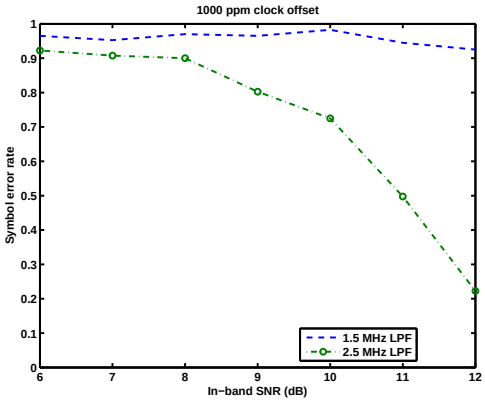

Figure 4.10: Before increasing the filter bandwidth, the input signal is completely filtered out, leading to no output signal regardless of the SNR.

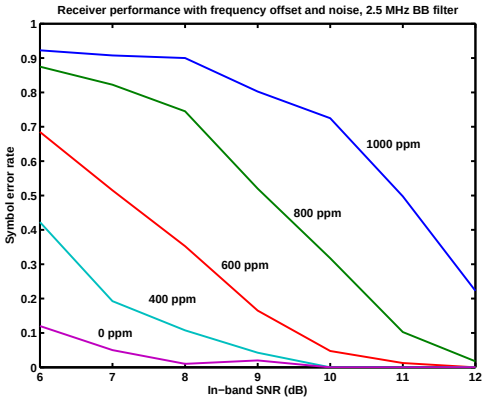

Figure 4.11: The noise performance of an 802.15.4 compatible receiver at varying clock offsets.

an on-chip oscillator. As pointed out above, the additional SNR required to compensate for the clock mismatch is often readily available.

4.3.3 Frequency offset compensation

The aforementioned noise performance can be significantly improved with a digitally controllable oscillator. Many on-chip oscillator designs include controllability, typically through

the use of switched resistors or capacitors. With sufficiently high resolution, such an oscillator can be commanded to get within a few ppm of any frequency, and thus, the LO can be driven to match the RF carrier frequency if it can be estimated.

The downconverted signal from the mixer is offset from the expected IF by the difference between the LO and RF carrier frequency. There have been algorithms devised to estimate this offset directly [30, 28], thereby recovering the clock error. However, in a standard digital OQPSK-HSS demodulator, a frequency shift in the input signal (at IF) translates to a constant DC bias in the demodulated output. Thresholding the demodulator output at this bias point returns the original data chips; the bias value itself can be used to extract the frequency offset between the LO and RF carrier frequencies.

The demodulator used in the above simulations was implemented on a GnuRadio software defined radio platform; the measured DC bias across frequency mismatch was extracted as described and plotted in figure **??**, and its linear relationship to the frequency offset can be clearly seen. By using this measured bias as an error value, a feedback controller can be wrapped around a variable on-chip clock as outlined in figure 4.13. At the extremes of frequency offset, the bias / offset relationship deviates from linear; nonetheless the clear monoticity remains present, and a feedback loop can drive the error signal to zero, aligning the LO with the RF carrier frequency.

The 802.15.4 standard includes a known 5 octet long PHY synchronization header on which this feedback loop can operate. Successive refinements can allow the feedback to be robust to an unknown slope, centering the LO on the RF carrier frequency for the PHY

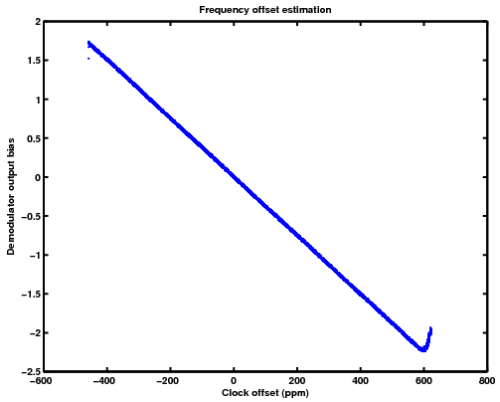

Figure 4.12: The DC bias of the demodulator output (in arbitrary units) is directly proportional to the frequency offset between the receiver LO and the transmitter's carrier frequency. By applying a feedback loop on this bias value to the LO generator, the receiver clock can be driven to match the transmitter clock.

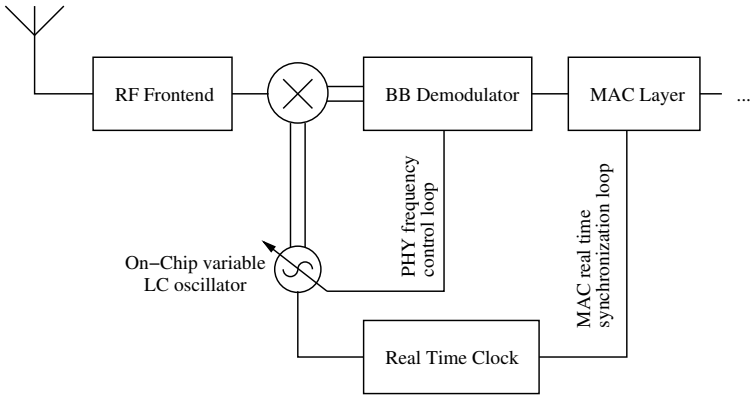

Figure 4.13: A block diagram depicting how the additional frequency offset estimate demodulator output can be used to adjust the LO frequency until it matches the RF carrier frequency

payload. At this point, the output of the mixer is at the expected IF, obviating the need for a wide bandwidth channel-select filter. By allowing the channel-select filter to be of a variable bandwidth, or by including a second stage filter (potentially in the digital domain), the excess noise bandwidth can be filtered out, restoring the original noise performance of a crystal driven 802.15.4 network.

4.3.4 Results

It is often simply assumed that a crystal frequency reference is necessary for RF communication, and as such any wireless sensor network solution will require some board level design. However, using data from actual hardware as a base, this work has presented simulations that demonstrate robust, narrow-band, standards compliant RF communication is indeed possible without an external quartz oscillator. By including the RF oscillator on chip, the path is paved to a single-chip, fully integrated wireless sensor mote.

4.4 Summary

The IEEE 802.15.4 protocol provides the de facto standard for wireless sensor networks, and contains a PHY and MAC layer specification, as well as guidelines for implementation. However, the specification are conservative, resulting in excessive redundancy limiting the performance of such networks. The typical excess SNR can instead be traded off for additional functionality in WSN nodes. By employing a variable data rate coding, energy

consumption can be minimized. By allowing imprecise frequency references, size and component complexity can be reduced.

These are of interest not only to conventional stationary WSN systems, but also those involving mobile elements. Energy savings at the network level results in a lower power draw, allowing robots to operate for longer on the same battery, or reduce size by needing a smaller battery. Similarly, reductions in the size of the network components lowers the payload weight of the robots, also reducing its power draw for the same benefits. Smaller motes also allows more motes to be carried given the same payload capacity, allowing for a larger network deployment.

Chapter 5

Mobile Networking

Though there are many details differentiating specific MAV and robotic applications, at the most basic they all involve moving throughout an environment and passing data to and from the greater world. Only in a very narrow range of specifications can applications be accomplished without any communication at all. Examples include fully autonomous systems, or robots that can store data on board then physically return to a base station. Otherwise, wireless network design becomes an integral consideration in such missions.

The ability for a robot to carry and deploy its own supporting network greatly expands the application space of such a system. That is, by bringing along wireless repeaters, a robot can be free to travel as dictated by the mission specification; deploying a communication network allows the robot to be untethered by limited range wireless links. A motivating example of such a system is sketched out in figure 5.1. In this example, a MAV can travel around an unknown environment with potential RF obstacles, wirelessly communicating

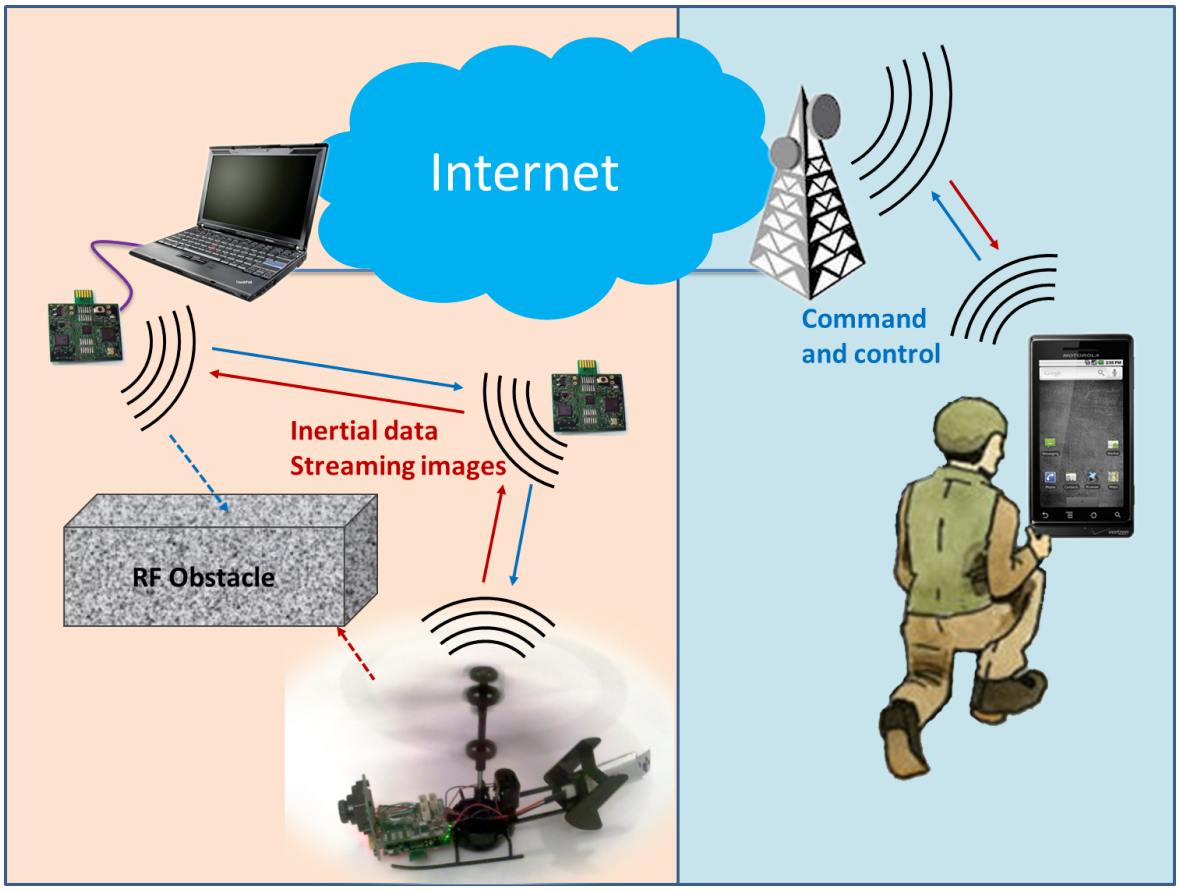

Figure 5.1: This schematic shows a sample MAV application. The laptop represents a base station and acts as a network manager, coordinating communications while serving as a sink for data generated by the MAV and deployed sensor nodes. Data generated by the MAV must be sent through the relays, while the base station can employ existing communications infrastructure to interact with the world.

with a base station or remote operator over its self-deployed infrastructure.

The geometry of the generated network significantly constrains its operation. A star topology on the underlying network would limit communication to only those nodes a single hop from the base station, while a multihop tree topology would be susceptible to loss of connectivity from the failure of a single node. As such, a reliable, robust network must form

a multihop mesh topology, where nodes can have multiple neighbors, and data can pass through several nodes on their way to and from the base station.

Especially in the case of aerial robots, the most significant design constraints of system hardware are weight and power consumption; the two go hand-in-hand. The lower the mass of the MAV with its payload, the less power is needed to fly, thus permitting longer missions. Similarly, the lower the power consumption, the smaller and lighter the battery needed for a given mission. Since the deployed network is initially dead payload weight on the MAV, minimizing overall network power consumption minimizes the battery requirements on the dropped-off nodes (or eliminates in entirely in favor of energy scavenging sources), thus lightening the load and increasing the scope of possible missions.

5.1 MAV-deployed networks

The helicopters of section 3.2 can be modified to enable deployment of wireless nodes with which to build a multi-hop network. In figure 5.2, an additional servo has been mounted to the base of the helicopter; that actuator controls a payload bay containing additional GINA motes. These motes are configured to be wireless repeaters, and can be dropped off by the MAV when connectivity to its previous parent begins to get low. Communication can then be routed through this deployed multi-hop path.

A base station connected to a VICON motion capture system was used to semi-autonomously navigate the helicopter into a mock building through a window in a simulated enviroment.

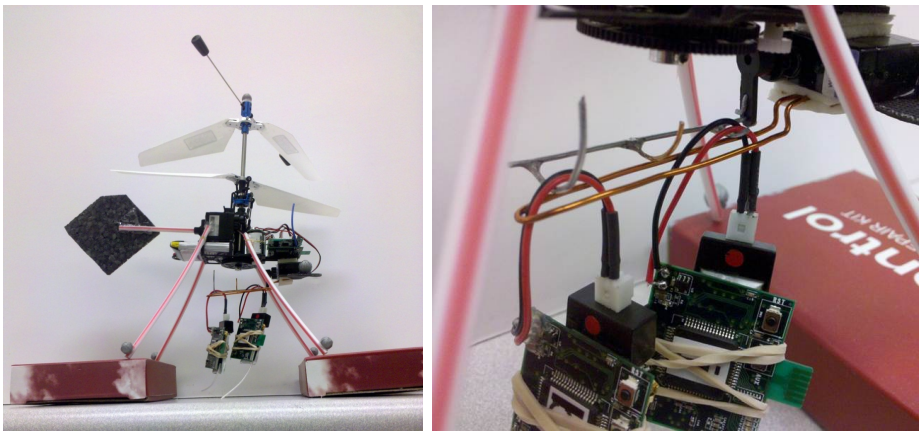

Figure 5.2: A MAV can carry additional wireless repeater nodes as deployable payload. These nodes can be dropped off over the course of a flight to extend communication range.

The wireless signal was significantly attenuated by the building walls – a direct point-to-point link between the helicopter and the base station sporadically dropped packets. Since the base station was responsible for guidanace control, the communication failure resulted in the MAV flying into walls.

The MAV instead was configured to drop off a mote immediately outside the window, then another immediately inside the window. The individual paths – from base station to outside node, outside mote to inside node, and inside node to helicopter – were all high reliability links, and so the resulting network was able to maintain control throughout the mission. Furthermore, the GINA motes contain inertial sensors, and could therefore contribute their own data to the application, potentially reporting on such environmental conditions as nearby footsteps.

The system can be further extended by connecting the base station to the internet as diagrammed above in figure 5.1. In figure 5.3, a helicopter with an onboard video camera

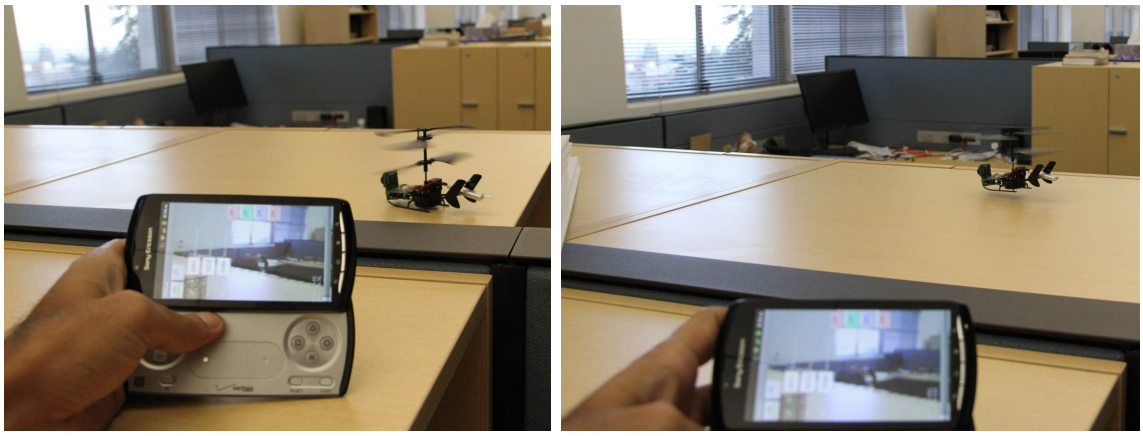

Figure 5.3: An experimental realization of figure 5.1 above passes downstream communication and upstream sensor data along a multi-modal multi-hop network path consisting of 3G, wifi, and 802.15.4 links.

is free to fly about an office space. It communicates over a three hop network of deployed GINA motes to a laptop base station. The laptop is connected via wifi to the internet, which allows a handheld cell phone on a 3G cell network to connect to it. An app on the phone can directly control the actuators on the helicopter, steering its motion, while the imaages captured on the helicopter are transmitted back and displayed on the phone.

5.2 Implementation limitations and design considerations

Though the drop-off motes in figure 5.2 consisting of a GINA board with a battery weigh only 5 g, such a payload is still a significant fraction of the 50 g MAV weight. As such, the addition of that payload significantly reduces the duration of a potential mission. A

helicopter taking off with two motes in its payload bay saw a reduction of flight time from 10 to 2 minutes; a third drop-off mote further shortened that to merely 30 seconds. The critical impact of payload mass is thus distinctly highlighted.

The GINA based drop-off mote divided its mass approximately halfway between hardware and battery. Section 4.3 demonstrated a path towards minimizing system mass by eliminating the need for external components. A single-chip wireless mote could be several orders of magnitude lighter than the multi-component board level motes today. Nonetheless, the battery is still another major contributor to the system weight. Minimizing energy required for communication can reduce the size of the power source for the deployed network. As such, the lessons learned from designing low power stationary WSN deployments must also be applied to these networks deployed by robots.

The generated data in such a system can come from the sensors on the MAV, using the deployed network nodes merely as a communications infrastructure, or it can also be generated by sensors on the deployed nodes themselves. In the case of reconnaissance-type missions, data from MAV sensors is often composed of short, high volume bursts (camera feeds, microphones, etc). The stationary network nodes on the other hand often monitor slowly-changing environmental phenomena (smoke detection, occupancy sensing, etc.). The challenge of designing such a system thus lies in adapting an energy efficient wireless network to both long-term, low data rate reliable sensing missions, while accommodating the burst of short-term, high data rate transmissions of highly mobile MAV agents.

The problem of maintaining reliable communication to mobile nodes through a WSN has

been studied both generally and for specific mission scenarios. In such studies, especially those dealing with MAVs, communication was often assumed to be single-hop line-of-sight from the mobile node to the base station [48, 40]. As previously described, most missions which have data generated in RF-challenged environments are simply infeasible without multihop communication to the outside world. Furthermore, lack of frequency channel diversity can cause a selected transmission channel to perform poorly due to external interference or multipath fading [44]. Even brief drops in connectivity due to such interference can have severe adverse real-time performance effects during a MAV mission.

Alternately, if relaying MAV data over a stationary multihop mesh network, nodes in the network are required to be always on – when not transmitting, each node must keep its radio in receive mode to ensure that data from the mobile node will be received by a node in the mesh [8, 35]. The need to keep nodes listening for incoming packets places a severe burden on the stationary mesh's energy budget; such persistent power consumption requires bigger, heavier batteries on each node. In the scenario where MAVs must deploy the multihop WSN mesh in an unknown environment, fewer nodes can then fit in the payload, which in turn reduces the maximum communication range of the MAV.

Power efficiency has been addressed by medium access control (MAC) layer protocols for stationary wireless networking stacks as discussed in the previous chapter. This set of protocols can then be adapted into multihop mesh networks for MAV operation to ensure low power consumption and communication reliability, while permitting rapid, high throughput data bursts when required.

Table 5.1: Differences in design parameters between stationary and mobile nodes in a hybrid sensor network

Metric	Stationary network	MAV nodes
Duration	10^6 seconds	10^3 seconds
Data rates	10^2 bits / second	10^5 bits / second
Latency	10^2 seconds	10^{-2} seconds
Network routing	stable	dynamic

5.3 Mobile nodes in a TSCH mesh

Table 5.1 shows the significant differences in design parameters which are imposed when comparing stationary nodes to MAV nodes.

Rather than building a single network architecture that compromises on all these metrics, a dual-mode network can be designed instead. The deployed stationary network operates by default as a standard time division multiple access (TDMA) based mesh as described in [39]. As new nodes are dropped off by the MAV, they join the mesh and carry on as described by the relevant TSCH standards. A modification is added to facilitate communications with the MAV; this communication is carried out on an independent set of channels. In an IEEE 802.15.4 network, 16 channels are typically available for transmission. In the proposed architecture, a subset of these channels is reserved for low throughput stationary mesh communications, while the remainder is only used when required by MAV agents.

5.3.1 TSCH modification

The stationary network dropped off by the MAV implements a low power, low duty cycle TSCH mesh. As nodes are dropped off, they join that mesh, with the routing configurations updating accordingly. The frame length must initially be set short enough to incorporate new nodes into the network on a rate comparable to the deployment of the nodes by the MAV. After the MAV mission, this can be slowed down as necessary.

Because the MAV is highly mobile, it will not maintain a constant set of neighbors, and so it is not used as a routing node in the mesh. It still listens for advertisement (ADV) packets to maintain synchronization with the network, but never sends out its own ADVs. Instead, it can send an acknowledgment packet (ACK) to an ADV to initiate communication into the network. After sending an ADV, stationary nodes listen to receive the ACK from a MAV. The advertisements are scheduled inversely proportional to neighbor count, so that on average the MAV would be able to hear one ADV every frame within a neighborhood without risking packet collisions. The frame length must be short enough so that the MAV remains within range of at least one node for the duration of a frame.

An extra slot is added to the frame after the ADV slot for communication from the network to the MAVs. If a node received an ACK to its ADV in the previous slot, it can then confirm with the MAV during the following MAV slot that it will switch to MAV mode for the duration of that frame. The proposed method is very similar to a slotted ALOHA scheme [1]. If two MAVs respond to an ADV at the same time, a packet collision may occur.

In this case, a randomized back-off timer is used to retry communication at a later time. Once a MAV receives a clear-to-send signal during the second slot of each frame (ACK slot) it enters one of two throughput modes.

5.3.2 Low throughput MAV mode

When ACKing an ADV packet, the MAV specifies the type of data it is intending to transmit. If the the data to be transmitted can be fit into a few packets, without stringent latency requirements, the mobile node can send this over the reserved MAV channels to the stationary node in the mesh. The stationary node responds to the MAV with slots in the frame during which it can receive data from the MAV. The MAV then communicates with the mesh as would any other node during its scheduled slots, and the resulting packets can then be routed through the stationary network according to the existing transmission schedule. This mode is outlined in figure 5.4. Multiple MAVs can simultaneously communicate to the mesh in this mode.

5.3.3 MAV burst mode

When the MAV has high volumes of data to transmit, or when latency is a primary concern, sending packets through existing mesh traffic becomes insufficient. In this case, the MAV requests a network mode switch in its ADV ACK. If this is confirmed by the stationary node in the MAV slot (see figure 5.5), then a dedicated multihop route will be opened up between

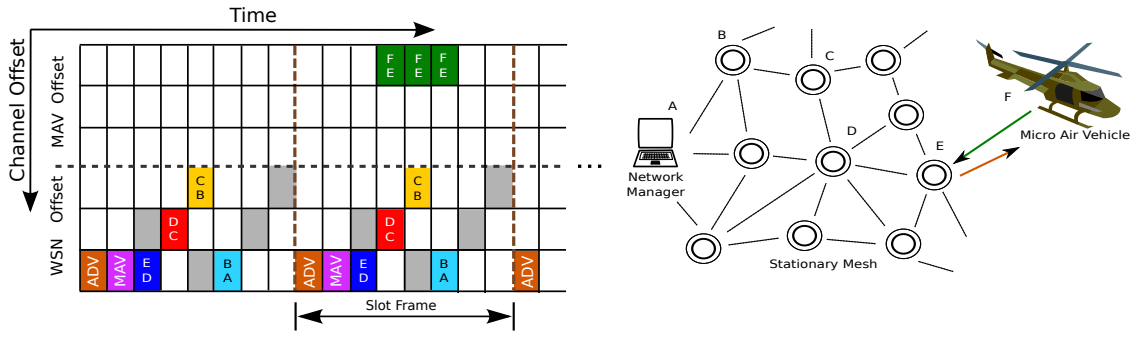

Figure 5.4: If the MAV has a small amount of data to send, it can request a number of slots to communicate to its nearest neighbor in the mesh. These slots can be predetermined by the manager or stationary nodes. That data then gets routed over the mesh as would any other data generated in the mesh.

the MAV and the base station. This route from MAV to base station is determined during standard operation of the mesh, e.g. as described in [2], and is unlikely to change over the duration of a slot-frame. Starting from the node in contact with the MAV, each node in the mesh informs its direct parent of the mode switch, and then shifts to the burst mode.

This upstream notification can happen in two ways. The standard TSCH communication could be used, in which case the notification would reliably propagate upstream over the course of a slot-frame. Burst mode could then be executed in the following slot frame. Alternately and preferably, a separate slot following the MAV slot could be dedicated to this function, in which every node would listen for a mode-switch notification, and immediately switch to burst mode if required.

The burst mode communication happens over an independent set of channels from the main mesh. During the streaming mode slot-frame, nodes not involved in the direct path from the MAV to the base station carry on network operations as usual. Meanwhile, those nodes

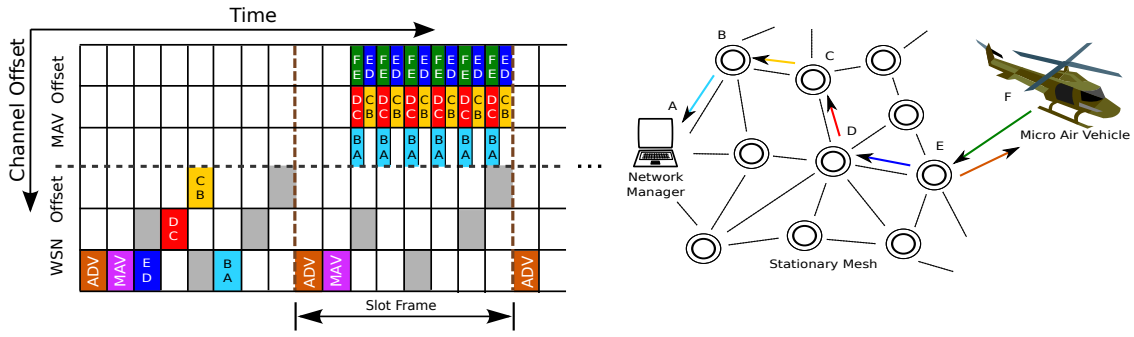

Figure 5.5: In burst mode, the nodes along the direct path from the MAV to the base station remove themselves from the TSCH network and communicate in a separate high throughput, low latency mode.

directly in-route between the MAV and base station establish a separate communications schedule over their channels. While burst communication schedule is similar to the standard TSCH communication, there are a number of important differences.

Since the data is a single, unidirectional, burst mode stream over a predefined path, most of the TSCH headers can be stripped out of the packets to yield a higher data throughput. For similar reasons, the slot length can be narrowed to transmit at maximum radio output (about 5ms per slot in an IEEE 802.15.4E setting) to facilitate higher packet density. For a standard IEEE 802.15.4E network, this can have the effect of doubling the number of available transmission slots. Link level ACKs are also eschewed in favor of higher bandwidth as this sort of data typically does not require very high reliability. This mode is shown in figure 5.5.

This burst mode subnetwork persists for one slot-frame. On the next frame, the network returns to its original state. Data still in transit along the path can either be discarded or

stored for transmission over the default channels. If the MAV has additional data it needs to transmit, it can again request burst mode operations by responding to the next ADV packet it receives.

5.4 System analysis

5.4.1 Overview

Extensive analysis on TDMA networking, in particular TSCH, has demonstrated its importance for low power wireless networking [39, 44, 46]. The low throughput MAV mode is effectively just an extension to a standard IEEE 802.15.4E (TSCH) network, with out-of-band communication reserved for direct point-to-point communication with the mobile MAV. Network maintenance and the remaining traffic is left to the primary unmodified IEEE 802.15.4 mesh, thus ensuring the robustness, reliability, and scalability inherited from the original standard. This time synchronized communication protocol has been implemented on the GINA motes under the OpenWSN stack, validating 10ms slots with an extremely efficient duty cycle for the stationary mesh [45]. This stack supports channel hopping and multihop routing at very low power.

Burst mode communications, on the other hand, augment the underlying TSCH MAC layer with an additional independent set of channels for the MAV burst data. This additional network is effectively a separate, stripped down TSCH network, derived from the original

mesh but operating on a separate set of channels. Given a single unique path from a MAV to the base station, a MAV cannot simply send one packet after the other. Since its parent node must forward the packet, the MAV must pause until the parent node is ready to receive again. This leads to the transmission schedule seen in figure 5.5.

5.4.2 Tuning network parameters

The hybrid network requires partitioning the available network channels between the stationary mesh and MAV communications. With one unique path to the base station, the optimal scenario involves the ability of the MAV to transmit a packet every other slot. If not enough slot offsets are available, the paths to the base station will fill the transmission schedule in a way that does not permit the MAV to transmit a packet every other slot. This is considered sub-optimal.

As a rule of thumb, the number of hops (paths to the base station) can only be twice the size of the reserved channels to ensure that the MAV can transmit a packet every other slot. As a MAV gets more hops away from the base station, more channels should be allocated accordingly to ensure that burst mode is carried out at the fastest possible rate. As such, to achieve high throughput, the number of allocated MAV channels must be at least half the number of hops that the data is expected to traverse (see figure 5.6).

With half-duplex radios, each node along the route must spend half its time receiving and half its time retransmitting the data along the pipe, thus limiting the throughput to half

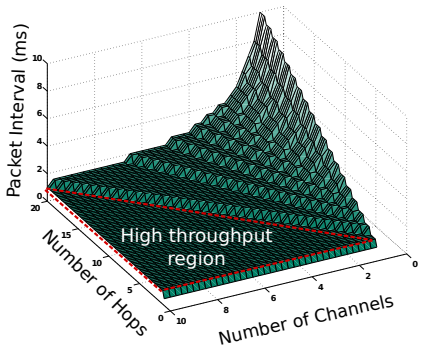

Figure 5.6: Given the expected maximum network distance (number of hops) from the MAV to the base station, a number of channels can be allocated to the MAV burst mode. As long as the number of channels is at least half the number of hops, the MAV will be able to communicate at the maximum supported throughput of the network given one unique path to the base station.

the available data bandwidth. However, with a sufficiently dense network, this limitation could be lifted. If two completely independent paths can be found from the MAV to the base station without overlapping intermediate nodes, data can be alternately sent along both routes, further doubling the throughput to the full available data rate. In this case, the number of reserved MAV channels would need to be at least half the number of hops from MAV to base station.

As there are only 16 channels available in the IEEE 802.15.4 standard, allowing a more extensive network requires more channels to be allocated for MAV use, and fewer for the underlying stationary mesh. However, as shown in [46], the robustness and efficiency benefits of channel hopping become evident with only a small subset of channels; peak performance can be achieved by allocating six channels to the stationary mesh. Ten channels can thus be available for MAV communications, allowing for a reliable network up to 20 hops deep.

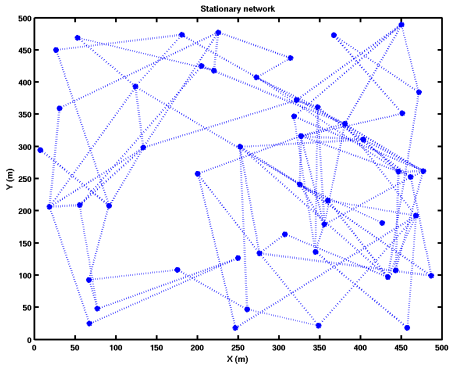

Figure 5.7: A simulated environment has stationary network nodes through which a MAV may travel. Network connectivity is indicated by dotted lines. Note that due to the nature of RF communication, connectivity is not geographical. Similarly, as the MAV travels around the network, the node to which it can communicate may change.

5.4.3 Performance evaluation

To evaluate the feasibility of extending a TSCH network to communicate with a mobile agent, a sample network was simulated using MATLAB. An area was populated with randomly distributed network nodes at varying densities (see e.g. figure 5.7) through which moved the mobile node. Time varying connectivity between nodes was modeled using a stochastic noise penalty atop the Friis equation [17]. This allows for the empirically observable random packet loss despite high average signal strength.

Under the proposed scheme, the MAV communicates into the mesh during a slot frame in which it receives an ADV packet from a nearby stationary node. If the stationary nodes send ADVs too infrequently, the MAV may not receive any packets; on the other hand, if the nodes transmit too often, the MAV may encounter a packet collision, again resulting in a failed ADV. The optimum is reached when each node transmits an ADV with probability

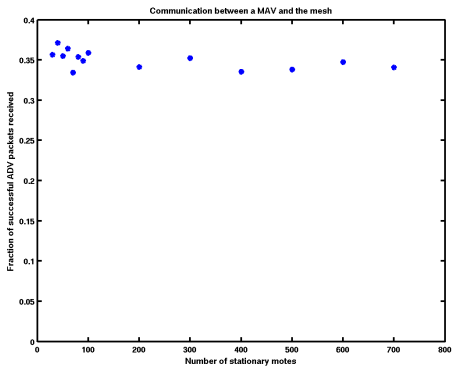

Figure 5.8: Regardless of the density of the underlying mesh network, the MAV successfully hears a fraction of the advertisement packets, and so can communicate during approximately 35% of the available slot frames.

inversely proportional to the number of neighbors it can hear.

In that case, and accounting for stochastic packet loss, the MAV received a successful ADV $\approx 35\%$ of the time, regardless of density (see figure 5.8). In those slot frames, then, data can be offloaded from the MAV into the mesh. Once in the mesh, the data can be transmitted to the base station as governed by the TSCH network parameters [39]. Depending on the desired metric, dropped data packets can be retransmitted to ensure reliability, or ignored to preserve latency.

This is less frequent than can be achieved with a single hop always-on point-to-point link. However, this scheme is inherently robust to scaling. It allows for reliable multi-hop communication even as the position of the MAV and the network topology change. The underlying TSCH network also eliminates the problem of network saturation due to broadcast flooding: since each packet transmission is scheduled and routed, the number of packets does not increase exponentially as the MAV gets farther from the base station. In

that regard then, the reduced throughput is necessary to ensure that the MAV data can arrive at the base station.

5.5 Summary

A low power multi-hop mesh network protocol is often necessary to support the operation of highly mobile MAV nodes. In contrast to earlier research, simulations have shown that it is feasible to extend a TSCH-like WSN stack, which features an IEEE 802.15.4E time synchronized channel hopping MAC layer, to enable rapid, reliable, low power multihop communications with mobile agents. This hybrid protocol permits both low-throughput and high-bandwidth data to be transmitted by the MAVs. This flexibility comes at no cost to network reliability, as effects due to external and intra-network radio interference are mitigated through scheduled communications and channel hopping. Furthermore, battery resources of the stationary mesh are preserved due to a low average radio duty cycle. This system proposes to greatly increase the range and duration of MAV missions in RF denied environments.

Chapter 6

Conclusions and Future Work

Bibliography

[1] N. Abramson. "Development of the ALOHANET". In: *Transactions on Information Theory* 31 (May 1985), pp. 119–123.

[2] N. Accettura et al. "Standardized power-efficient & internet-enabled communication stack for capillary M2M networks". In: *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*. 2012, pp. 226 –231.

[3] L. Alonso and R. Agusti. "Automatic rate adaptation and energy-saving mechanisms based on cross-layer information for packet-switched data networks". In: *IEEE Communications Magazine* 42.3 (Mar. 2004).

[4] *Atmel AT86RF231 Datasheet*. `http://www.atmel.com/dyn/resources/prod_documents/doc8111.pdf`.

[5] E. Beyer and M. Costello. "Measured and Simulated Motion of a Hopping Rotochute". In: *J. of Guidance, Control, and Dynamics* 32.5 (2009).

[6] E. Beyer and M. Costello. "Performance of a Hopping Rotochute". In: *International J. of Micro Air Vehicles* 1.2 (2009), pp. 121–137.

[7] P. Birkmeyer, K. Peterson, and R.S. Fearing. "DASH: A dynamic 16g hexapedal robot". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (Oct. 2009), pp. 2683 –2689.

[8] G. Cardone, A. Corradi, and L. Foschini. "Reliable communication for mobile MANET-WSN scenarios". In: *Computers and Communications (ISCC), 2011 IEEE Symposium on*. 2011, pp. 1085–1091.

[9] *CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Tranceiver*. `http://focus.ti.com/lit/ds/symlink/cc2420.pdf`.

[10] Ling-Jyh Chen et al. "Enhancing Bluetooth TCP throughput via link layer packet adaptation". In: *Proc. ICC 2004*. Vol. 7. June 2004, pp. 4012–4016.

[11] Michael Crichton. *Prey*. New York: HarperCollins Publishers, 2002.

[12] L. Doherty, W. Lindsay, and J. Simon. "Channel-Specific Wireless Sensor Network Path Data". In: *Proc. Intl. Comp. Comm. and Networks (ICCCN)*. Aug. 2007, pp. 89–94.

[13] Mischa Dohler et al. "Kumars, Zipfs and Other Laws: How to Structure a Large-Scale Wireless Network?" In: *Annals of Telecommunications* (2008).

[14] S. Drago et al. "A 2.4GHz 830pJ/bit duty-cycled wake-up receiver with -82dBm sensitivity for crystal-less wireless sensor nodes". In: *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*. Feb. 2010.

[15] A. El Fawal et al. "Trade-off analysis of PHY-Aware MAC in low-rate low-power UWB networks". In: *IEEE Communications Magazine* 43.12 (Dec. 2005), pp. 147–155.

[16] Andrew Mark Eldredge. "Improved state estimation for miniature air vehicles". MA thesis. Brigham Young University, 2006.

[17] H.T. Friis. "A Note on a Simple Transmission Formula". In: *Proceedings of the IRE* 34.5 (1946), pp. 254 –256. ISSN: 0096-8390.

[18] S.J. Gizinski and J.D. Wanagas. "Small Satellite Delivery Using a Balloon-Based Launch System". In: *International Communication Satellite Systems Conference and Exhibit*. Mar. 1992.

[19] R.M. Haralick et al. "Analysis and solutions of the three point perspective pose estimation problem". In: *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*. June 1991, pp. 592 –598.

[20] Darren L Hitt, Charles M Zakrzwski, and Michael A Thomas. "MEMS-based satellite micropropulsion via catalyzed hydrogen peroxide decomposition". In: *Smart Materials and Structures* 10.6 (2001), pp. 1163–1175. URL: `http://stacks.iop.org/0964-1726/10/1163`.

[21] Y. T. Hou et al. "Flow routing for variable bit rate source nodes in energy-constrained wireless sensor networks". In: *Proc. ICC 2005*. Vol. 5. May 2005, pp. 3057–3062.

[22] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical

Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)". In: *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)* (July 2006), pp. 1 –320.

[23] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer". In: *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)* (Apr. 2012), pp. 1 –225.

[24] Florian Kehl. "An attitude controller for small scale rockets". MA thesis. University of Basel, Switzerland, Oct. 2009.

[25] B. S. Kim, Yuguang Fang, and T. F. Wong. "Rate-adaptive MAC protocol in high-rate personal area networks". In: *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC)*. Vol. 3. Mar. 2004, pp. 1394–1399.

[26] S. Lanzisera, A.M. Mehta, and K.S.J. Pister. "Reducing Average Power in Wireless Sensor Networks through Data Rate Adaptation". In: *IEEE International Conference on Communications, ICC '09*. June 2009, pp. 1 –6.

[27] C. E. Luna et al. "Joint source coding and data rate adaptation for energy efficient wireless video streaming". In: *IEEE J. Sel. Areas in Comm.* 21.10 (Dec. 2003), pp. 1710–1720.

[28] R. Mehlan, Yong-En Chen, and H. Meyr. "A fully digital feedforward MSK demodulator with joint frequency offset and symbol timing estimation for burst mode mobile

radio". In: *Vehicular Technology, IEEE Transactions on* 42.4 (Nov. 1993), pp. 434 – 443.

[29] *Microstrain 3DM-GX2*. `http://microstrain.com/3dm-gx2.aspx`.

[30] M. Morelli and U. Mengali. "Joint frequency and timing recovery for MSK-type modulation". In: *Communications, IEEE Transactions on* 47.6 (June 1999), pp. 938 –946.

[31] R.A. Nakka. *Solid Propellant Rocket Motor Design and Testing*. University of Manitoba, 1984.

[32] J.M. Rabaey. "The swarm at the edge of the cloud-A new perspective on wireless". In: *VLSI Circuits (VLSIC), 2011 Symposium on*. IEEE. 2011, pp. 6–8.

[33] E.L. Rainwater and M.S Smith. "Ultra high altitude balloons for medium-to-large payloads". In: *Advances in Space Research*. Vol. 33. 10. 2004, pp. 1648–1652.

[34] K. Romer and F. Mattern. "The design space of wireless sensor networks". In: *IEEE Wireless Communications* 11.6 (Dec. 2004), pp. 54–61.

[35] S. Sajadian et al. "Improving Connectivity of Nodes in Mobile WSN". In: *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*. 2011, pp. 364–371.

[36] Nesrin Sarigul-Klijn, Christopher Noel, and Marti Sarigul-Klijn. "Air Launching Earth-to-Orbit Vehicles: Delta V gains from Launch Conditions and Vehicle Aerodynamics". In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. Jan. 2004.

[37] *Specification of the Bluetooth system: wireless connections made easy*. `http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21__EDR.zip`.

[38] Neal Stephenson. *The Diamond Age*. New York: Bantam Books, 1995.

[39] *Technical overview of time synchronized mesh protocol (TSMP)*. `http://www.dustnetworks.com/docs/TSMP_Whitepaper.pdf`.

[40] S. Todorovic and M.C. Nechyba. "A vision system for intelligent mission profiles of micro air vehicles". In: *Vehicular Technology, IEEE Transactions on* 53.6 (Nov. 2004), pp. 1713–1725. ISSN: 0018-9545.

[41] A. Toorian, K. Diaz, and S. Lee. "The CubeSat Approach to Space Access". In: *Aerospace Conference, 2008 IEEE*. Mar. 2008, pp. 1–14.

[42] J. A. Van Allen. "Balloon-Launched Rockets for High-Altitude Research". In: *Sounding Rockets*. Ed. by H Newell. 1959. Chap. 9.

[43] T. Vladimirova et al. "Characterising Wireless Sensor Motes for Space Applications". In: *Adaptive Hardware and Systems, 2007. AHS 2007. Second NASA/ESA Conference on*. Aug. 2007, pp. 43–50.

[44] T. Watteyne et al. "Mitigating multipath fading through channel hopping in wireless sensor networks". In: *Communications (ICC), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1–5.

[45] T. Watteyne et al. "OpenWSN: a standards-based low-power wireless development environment". In: *Transactions on Emerging Telecommunications Technologies* 23.5 (2012), pp. 480–493.

[46] Thomas Watteyne, Ankur Mehta, and Kris Pister. "Reliability through frequency diversity: why channel hopping makes sense". In: *PE-WASUN '09*. Tenerife, Canary Islands, Spain, 2009, pp. 116–123.

[47] Starsky H. Y. Wong et al. "Robust rate adaptation for 802.11 wireless networks". In: *Proceedings of the conference on Mobile computing and networking*. Los Angeles, CA, USA: ACM, 2006, pp. 146–157.

[48] Huaiyu Wu, Dong Sun, and Zhaoying Zhou. "Micro air vehicle: configuration, analysis, fabrication, and test". In: *Mechatronics, IEEE/ASME Transactions on* 9.1 (2004), pp. 108–117. ISSN: 1083-4435.

[49] S. Zarovy, M. Costello, and A. Mehta. "Experimental method for studying gust effects on micro rotorcraft". In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* (2012).

[50] Jiansong Zhang et al. "A Practical SNR-Guided Rate Adaptation". In: *Proc. INFOCOM 2008*. Apr. 2008, pp. 2083–2091.