DESIGN OF A FOUR-ROTOR AUTONOMOUS HELICOPTER FOR VEHICLE-BASED PHASED ANTENNA ARRAYS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Sean Hugh Breheny

January 2008

ABSTRACT

Autonomous flying vehicles (AFVs) have applications ranging from police surveillance to military synthetic aperture radar mapping. In many such applications, AFVs must transmit large quantities of data to a distant base station. Since this can be difficult without the ability to place large, high-gain antennas on small vehicles, we propose the idea of flying the vehicles in formation while transmitting and forming a phased-array antenna with one element on each vehicle.[1 2 3 4]

To determine the feasibility of this concept, we studied the constraints imposed both by antenna theory and by present AFV technology. While position errors in the array can cause a great reduction in antenna gain, good position sensing can allow phase compensation to recover most of the ideal case antenna gain even with significant position error.

Two electrical team members and one mechanical member worked together to produce a four-rotor helicopter-type AFV which was tested as well as modeled in simulation. The physical vehicle demonstrated autonomous hover and bench tests indicated that it should have 0.8 g excess thrust beyond hover and 10 minute battery endurance. Simulation of a formation of several of these AFVs showed very promising improvements in predicted AFV communication range (in one example, 9.6 dB gain for an array of ten in 10 MPH wind, for a range improvement of a factor of 3, representing 100 kilometer range with a total array power of only 1.5 Watts). Future work may include an actual physical test of formation flight of several AFVs, possibly even with a phased array antenna system, or at least with an antenna simulation based on real position data. This latter simulation was already performed on data from

formation flight simulations, provided by Honeywell, of their Organic Aerial Vehicles (OAVs), showing promising antenna array gain results.

BIOGRAPHICAL SKETCH

Sean Hugh Breheny was born on January 8th, 1980 in New York City. At age 7, he and his family moved to Scranton, Pennsylvania where he eventually attended Scranton Preparatory School. During childhood and adolescence, Sean played the Irish tin whistle, became an amateur radio operator, and was always interested in how things work. He attended Cornell University, where he continued his interests in music and the sciences, as well as serving as president of the student pro-life organization from 1999 until 2003. He was graduated in May 2001 with a B.S. in Electrical and Computer Engineering.

With the encouragement of Professor Raffaello D'Andrea, Sean began a PhD program in Dynamics and Control Systems at Cornell. He ultimately took a leave of absence from the program after finishing his research up to the Master's level to explore the idea of becoming a Catholic priest. After spending two years in the seminary, he left and is currently working as an electrical engineer for Kiva Systems, Inc. in Woburn, Massachusetts.

My research work during this project and this Master of Science Thesis are dedicated to Mary, the Mother of God, Queen of All Hearts, and Patroness of all scientists.

Ego sum totus tuus, Maria!

ACKNOWLEDGMENTS

Professor Raffaello D'Andrea, my committee chairman, was an incredible mentor to me from my sophomore year until graduate school, and even beyond. I will be forever grateful for his guidance and his friendship. I also wish to thank Profs. Donald Farley and David Delchamps for their willingness to serve on my committee and to answer my questions at many points during this project. Profs. Paul Kintner and Mark Psiaki of Cornell also provided very valuable assistance to my research by answering questions about antenna theory, GPS, and estimation using inertial data. My very sincere thanks go to Dr. JinWoo Lee for his help with wireless communications and other technical advice.

Over the four years during which I worked on this project in one capacity or another (as an undergraduate or as a graduate student), I met many utterly amazing students. Chief among them would be Eryk Nice and Jeremy Miller, with whom I spent many a night working on this project. They are both very gifted, talented, and hard-working engineers as well as generous and kind men. Their ideas and advice served me well during this work and in my endeavors since then. Among the others with whom I worked, Leard Huggins, Owen Fong, Matthew Earl, J. Andrew Eichelberger, and Ali Squali stand out as excellent engineering colleagues and friends.

Dr. Andrew Stubbs and Andrea Okerholm helped immensely during the writing phase by proofreading and providing advice on how to finagle Microsoft Word into doing what I wanted. Both of them are excellent engineers in their own right with whom I am honored to share my work.

Lastly, I would very much like to thank my parents, John and Regina Breheny, for their constant support and interest in my education and goals. My father, John Breheny, died during this project of a long-standing illness. Despite his declining health and the struggles this brought to my mother, both of them remained actively interested in my research.

This research was funded by U.S. Air Force Grant F49620-02-0388 and by the National Science Foundation Graduate Research Fellowship awarded to me in 2001.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Because of advances in control theory and electromechanical system miniaturization, interest in autonomous flying vehicles (AFVs) has increased in recent years.[5] Although such vehicles are totally or mostly autonomous in regard to their flight control system and navigation, it is almost always desirable for AFVs to be able to communicate with a base station (either directly or via satellite), especially when they are used to gather data like video or synthetic aperture radar maps.[6] Such a base station may either be a fixed installation, land vehicle, ship, aircraft, or satellite.

Although nearly all currently operational AFVs operate independently of other AFVs, it is logical to investigate whether coordinated groups of vehicles might be able to accomplish their task more efficiently than a single AFV or an uncoordinated group, just as teams of people or packs of animals can often work better together than as individuals. A moderate or large group of AFVs could easily generate very large volumes of data which may need to be transmitted in near real-time to a base station.[7]

Transmitting with a high data rate from a small flying vehicle to a satellite or base station is very challenging, primarily because of antenna size restrictions. Even if an efficient antenna can be accommodated on the aircraft, it is unlikely that it can be large enough to be highly directive. If a means could be found to allow AFVs to use highly directive antennas, their transmission and reception data rate could be increased by a factor of the directivity of the antenna.[7]

One such possible means would be to form a phased array antenna from a cluster of coordinated AFVs. The two-year research project detailed in this paper was an attempt to investigate the feasibility of this method by a combination of simulation and actual prototype AFV construction and testing. First, this paper discusses the project of designing and building a four-rotor helicopter-type AFV, which was capable of semi-autonomous hover in testing (semi-autonomous in the sense that human velocity command input was sometimes necessary to prevent it from drifting away due to inertial navigation errors – it later demonstrated fully autonomous hover). Secondly, it provides an overview of a simulation-based subproject for DARPA (Defense Advanced Research Projects Agency) to take high-fidelity simulated vehicle flight data from AFVs manufactured by Honeywell and predict antenna array performance. Then we conduct a more theoretical analysis of the problem of forming a phased array antenna composed of mobile elements aboard AFVs, including a simple cost-benefit type analysis of several options for base station and AFV antenna type. Finally, the paper concludes with schematics, references, suggestions for future work, and an appendix on basic antenna theory.

The research detailed in this paper suggests that airborne vehicle-based phased arrays would be a feasible solution to the problem of AFV communication. This solution requires decent control of vehicle position and orientation, but relies more heavily on **good position sensing**. This is the benefit of phase compensation, which allows the AFVs to adjust the phase of each transmitter so that their signals add in phase at the receiver location. This works, too, in reverse for the case where the AFVs are the receivers. There are also more involved cases where the elements are close enough to each other to be more than lightly coupled (i.e., they induce significant voltages in

each other due to the signal from adjacent elements). While we assume that each element in the array is driven by a source (i.e., we do not deal with the case of parasitic arrays such as Yagi arrays), we do handle element-element coupling via a coupling matrix method in CHAPTER 3.[2 8]

CHAPTER 1

AFV CONSTRUCTION SUBPROJECT

1.1 Definition of Terms

"a" – $a=K_i/(JR)$, when multiplied by "b", gives the reciprocal of the motor-propeller system time constant. Convenient for simplifying motor equations.

ADC – Analog to Digital Converter

"b" – $b=K_i$, motor torque-current constant.

"C" (in the context of battery capacity) – used to refer to the current which would discharge a battery in one hour of continuous operation, ignoring fast discharge effects. For example, saying that a 2 Amp-hour battery is being discharged at 4 C would indicate that an 8 Amp current is flowing through it.

"d" – indicates the ratio of PWM "on" time to total switching cycle period. For example, if the PWM switch is on for 250 microseconds each cycle with a 1kHz PWM frequency (1000 microsecond period), d would be equal to 0.25

E(s) – Laplace transform of transfer function from commanded angular speed (radians per second) to control effort (Volts)

F(s) – Laplace transform of transfer function from command angular speed to actual angular speed (both in radians per second)

G_i,G_{i_raw},$G_{i\_theoretical}$ – integral gain (G_i,$G_{i\_theoretical}$ are the same quantity, the integral gain in normal physical units of Volts per radian. G_{i_raw} is the integral gain in units of duty fraction per encoder count, which is used by the microcontroller code)

G_p,G_{p_raw},$G_{p\_theoretical}$ – proportional gain (G_p,$G_{p\_theoretical}$ are the same quantity, the proportional gain in normal physical units of Volts per radian per second. G_{p_raw} is the proportional gain in units of duty fraction per encoder count per control loop time, which is used by the microcontroller code)

Half H-bridge – A pair of switching elements (such as MOSFETs) which connect a node either to the positive supply rail or to ground

I, P gains – Integral and Proportional gains

I_{batt} – battery average DC current in Amps

I_{motor} – average DC current in Amps which flows through whichever pair of motor terminals is presently energized

IMU – Inertial Measurement Unit – a device consisting of three angular rate gyroscopes and three linear accelerometers which provides data to be used for vehicle navigation.

IMU drift – the effect which slowly-changing sensor error offsets in an IMU have on navigation estimates. It is relatively easy to remove truly static sensor offsets, but the portion of the offset which changes over time and temperature causes the estimate of orientation, velocity, and position to drift away from the correct values over time.

J – propeller moment of inertia about the propeller shaft in kilogram meters^2

K – linear coordinate in the AFV local coordinate system corresponding to Z in global coordinates. Depending on context, K can also be the feedback control system gain matrix.

K_d – propeller drag torque constant (measured at the propeller shaft) in Newton-meters per (radian per second)^2. This is using a simple quadratic model of propeller drag torque.

K_i – motor torque-current constant (measuring torque after gearing, at the propeller shaft) in Newton-meters per Amp.

K_v – motor "back-emf" constant (measuring rotational speed after gearing, at the propeller shaft) in Volts per radian per second.

LQR – Linear-Quadratic Regulator control – a linear state-space controller, consisting of a simple static gain matrix which is computed to minimize a quadratic cost function[9]

MCU – Motor Control Unit – one of the four local motor controllers on the autonomous flying vehicle.

PI – Proportional Integral control

PID – Proportional Integral Derivative control

PWM – Pulse Width Modulation – a technique for controlling the current in motor windings using only on/off switch devices which are switched at a fixed frequency and whose "on" time is varied between 0 and 100% of each switching cycle.

Pseudostate – AFV state vector expressed in mixed local, global coordinates (local angular rates, global Euler angles, global linear position and velocity) and augmented to include X, Y, and Z velocity error integral and local X, Y, Z (or I,J,K) acceleration. The vector is augmented so that it can simply be multiplied by a static gain matrix to obtain the four propeller angular speed deviation commands.

R – motor driving system resistance (equal to R_w+R_m) in Ohms

R_m – motor winding resistance plus MOSFET switch resistance in Ohms.

RS-422 – A wired serial communication standard specifying differential signaling.

R_w – resistance of the entire motor circuit except for the motor winding and MOSFET switch resistance (this includes battery and wiring resistance) in Ohms.

SPORT – SerialPORT – the synchronous serial port peripherals on the ADSP-21062 DSP IC.

TWI – Two Wire Interface – Atmel's name for a synchronous two-wire bus compatible with Philips I2C standard

USART – Universal Synchronous/Asynchronous Receiver Transmitter – serial port peripherals on the Atmel AVR ATMega 128.

V_b – Battery open-circuit voltage

V_{emf} – motor "back-emf" voltage. This is the open-circuit voltage which is measured at a spinning motor's terminals.

$V_{in}(s)$ – Laplace transform of the effective voltage applied to the motor (this is using the simplification that V_{in}=dV_b)

Θ – Euler angle about the Y axis in radians

τ – time constant of closed-loop local motor control system in seconds

Φ – Euler angle about the X axis in radians

Ψ – Euler angle about the Z axis in radians

$\Omega(s)$ – Laplace transform of rotational velocity of the propeller shaft in radians per second

1.2 System Design Approach

1.2.1 Overview

After two failed attempts at producing an AFV, the team decided that it was necessary to invest considerably more resources in the project in order to ensure that it would have adequate performance at completion. Also, the project goals were now higher than simply achieving hovering flight, but included the ability to do some aerobatic maneuvers and eventually, to hold a tight formation with another flying vehicle. The team therefore selected the best (in the sense of drift rate specifications) inertial measurement unit in the mid-range class, the Systron-Donner Digital Quartz Inertial Measurement Unit (DQI-105).[10] The design also planned eventually to incorporate a differential carrier-phase GPS unit, probably the NovAtel Millennium RT-2. We also set a goal of attaining about a 2:1 thrust to weight ratio with an endurance of at least 10 minutes at hover.

In the design of a small four-rotor helicopter, there are only a few critical design factors: motor/propeller time constant, maximum thrust, ability to deal with translational velocity, endurance, sensor error, and structure rigidity. The time constant of the motor/propeller system interacts with the maximum thrust to form a common control problem: the control of a nonlinear system with limited actuator response speed. The combination of the two place limits on the maximum disturbance amplitude that the system can handle as well as on the maximum overall response speed and disturbance rejection ability. In essence, the better one rejects disturbances, the faster one can respond but the lower the maximum disturbance that one can handle without saturating actuators. It is also true, however, that very light disturbance

rejection can also cause instability by allowing the vehicle to move farther away from the linearized control point, which adds both nonlinear effects and also simple angular coupling if the deviation is angular such as in yaw.

Maximum thrust also affects the ability to do aerobatic maneuvers. Because of the nature of propellers, translational velocity (either vertical or horizontal) of the vehicle can alter the force that each propeller produces. As these are nonlinear effects, it is desirable to be able to neglect them, but one must ensure that the expected translational velocities are low enough to do so. The factors that affect this are propeller pitch and rotational speed.

With any feedback control problem, the limiting factors are actuator limitations, dynamics, or sensor error. The AFV dynamics and actuator limitations were already mentioned. The AFV's sensors should provide linear position and angular orientation data. They also need to provide angular rate data and linear velocity information. Because of time constraints, the team decided that the first version should use only inertial sensors (the IMU and no GPS), perform velocity control, and take velocity commands from a human pilot. The onboard controller would then act similarly to the fly-by-wire system on many modern aircraft, providing stability augmentation but not totally alleviating the need for pilot direction (i.e., the pilot would fill the role of the GPS, providing low-frequency correction of position and velocity).

The main problems encountered with inertial sensors are sensor drift and noise. Drift is a slowly changing offset which causes continually growing error when integrated

(to obtain angular orientation from angular rate gyro outputs, for example). Since drift is such a serious problem when there are no non-inertial sensors available, we decided to obtain a high-quality IMU with the best drift specifications we could obtain.

1.2.2 Batteries

The E-Tec 1200 High Output Lithium Polymer cells have an initial fully-charged voltage of about 4.2 Volts and may be operated down to about 3 Volts. According to somewhat loose hobby-type specs, they can be used at up to 6 C discharge rate continuously and 7 to 9 C on a shorter-term basis.[11] A very simple model for them (which we used in many of our calculations) would be a 3.8 Volt perfect cell with 0.070 Ohms in series. The next level of complexity would be to recognize that the cell's response to a sudden current drain is more like an initial voltage drop followed by a slower settling to a new value. In addition, it appears that the effective resistance is not constant with current, so that larger currents do not cause as much of a voltage drop as one would expect by simply computing the resistance based on the voltage drop at a lower current. An example of this effect would be a cell which shows a 1 Volt drop for 1 Amp and a 1.5 Volt drop for 2 Amps. In addition, there is a decrease in effective capacity with higher currents. Figure 1 below shows several test runs of a two-cell pack of these cells by the distributor or manufacturer.

1.2.3 Motors and Propellers

The propellers used for the AFV can be fairly well modeled as providing a thrust which is proportional to the square of the RPM and a counter-torque drag which is also proportional to the square of the RPM.

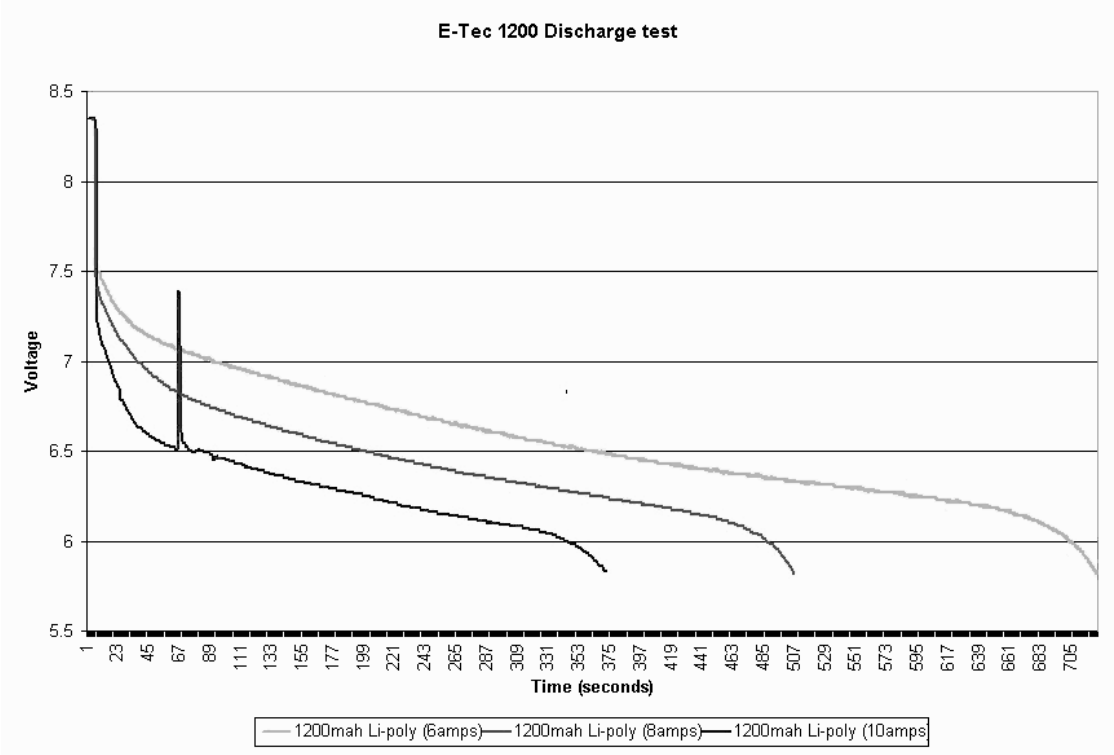


Figure 1 - E-Tec Lithium Polymer 1200mAH Cell Discharge Curves (2 cells in series)

If we express the rotational velocity in radians/second, the thrust in Newtons, and the drag torque in Newton-meters, the thrust constant for the APC 18x6 propeller is about 8.4×10^{-5} Newtons per radian^2/second^2 and the drag constant is 3.14×10^{-6} Newton meters per radian^2/second^2. The propeller was measured to have a moment of inertia about the usual rotation axis of 0.0011 kilogram meters^2. The introductory book by Anderson was useful as a reference for simple aerodynamics.[12]

The MaxCim MaxN32-13D motor, when coupled with the three half-H bridge driver circuit in the MCUs, can be adequately modeled as having a resistance of 0.022 Ohms, an inductance of 11 microHenries, and a voltage constant of 2500 RPM per Volt (which translates to 0.0255 Volts per radian per second after the 100:15 belt ratio).

The current constant therefore is also 0.0255 Newton-meters per Amp after belt drive as required by conservation of energy (Just as a gearbox must change RPM by the same factor by which it changes torque, a motor's voltage constant and current constant must equal each other in these units).[13]

The inductance and resistance together (including the resistance in the wiring and MOSFETs) determine the minimum PWM frequency for low current ripple (high efficiency) operation. The RL current time constant is L/R. In our case, a pair of MOSFETs and wiring contributes about 0.015 Ohms. This yields a time constant of 300 microseconds, considering the motor inductance and resistance given above. The PWM period must be several times smaller than this in order to achieve low ripple. The time constant would correspond to a PWM frequency of about 3.4kHz. Since the MCUs use a frequency of 31.25kHz, they are operating well above the minimum frequency.

When PWM is operated in the low-ripple mode and the battery is sufficiently bypassed by large capacitors, the equation which relates motor current to PWM duty is the following:

$$I_{motor} = \frac{dV_b - V_{emf}}{d^2 R_w + R_m}, I_{batt} = dI_{motor}$$ **Equation 1-1**

where d is the duty (from 0 to 1), V_b is the battery open-circuit voltage, V_{emf} is the motor back emf (voltage constant times angular speed), R_w is the resistance of batteries and wiring going to the batteries, and R_m is the resistance of the motor plus wiring from motor plus MOSFETs. This also shows that the battery current is lower than the motor current by a factor of d.

When the motor current is within the achievable range ($\frac{-V_{emf}}{R_m} \geq I_m \geq \frac{V_b - V_{emf}}{R_w + R_m}$), this equation can also be solved in reverse to yield:

$$d = \frac{V_b - \sqrt{V_b^2 - 4I_{motor}R_w(I_{motor}R_m + V_{emf})}}{2I_{motor}R_w}$$ **Equation 1-2**

It is also easy to solve for the maximum angular speed (radian/second) for a given motor, gearing, prop combination (note that this includes the gearing in the motor constants):

$$\omega_{\max} = \frac{\sqrt{K_v{}^2 + 4V_b\frac{K_d}{K_i}(R_w + R_m)} - K_v}{2(R_w + R_m)\frac{K_d}{K_i}}$$ **Equation 1-3**

where K_d is the propeller drag torque constant, K_i is the motor current constant (same as K_v).

To determine the optimum motor, gear ratio, and battery configuration (number of cells in series by number in parallel) we conducted a large optimization in MATLAB. The code for this is included on Sean Breheny's MS Thesis CD (see X:\2003 Electronics CD\matlab code\genl_opt2.m). The optimization checked all combinations of several different motors, several gear ratios, and various different battery configurations. It eliminated those which would exceed the motor limitations and also those which had less than 5 minutes endurance at hover. It then looked for the one which yielded the greatest excess thrust beyond hover. This optimization was originally constructed when we were planning on using NiMH cells and Astroflight motors. It turned out to be very useful also when we switched to MaxCim motors and LiPoly cells. The seven cells in series by two in parallel configuration was originally

planned in order to lighten the load on the Astroflight 05 motors but it is also a reasonable compromise for the MaxCim motors, although they could also handle the 8x4 configuration which would increase endurance and also, to a lesser extent, excess thrust. However, the excess cost outweighs these advantages. The expected endurance for the 7x2 configuration is about 10 minutes at hover with an excess thrust of about 1 g. An actual single motor test run confirmed that the packs could indeed sustain the hover point for 11 minutes. The 8x4 or a 7x4 configuration would move the endurance to about 20 minutes.

We did notice, however, in bench testing that one pack (out of four tested) of E-Tec cells had one cell fail during an 11 minute hover-speed test. This probably means that the specifications given by the distributor are not very conservative and these cells are really being pushed to their limits by our operating current level.

1.3 Electronic Systems Design

1.3.1 Overview

The AFV's onboard electronic systems are designed simply to control the vehicle, accept commands, and send telemetry back to the base station. These tasks are accomplished by the following units: four motor control units, Systron-Donner DQI-105 inertial measurement unit, main electronics assembly, and Airtronics radio control (RC) command receiver. Each motor control unit consists of a motor control board, 14 cell (7x2) Lithium Polymer battery pack, MaxCim MaxN32-13D motor, and a U.S. Digital E5S-512-375-IHA incremental encoder. The main electronics assembly consists of two boards (*main1* and *main2*) and the four cell Lithium Polymer main

electronics battery. The schematics for these two PCBs can be found in APPENDIX B (Figure 16 and Figure 17). Throughout the design, at various times, the excellent reference *The Art of Electronics* by Horowitz and Hill was used. [14]

Each motor control unit (MCU) is a self-contained system which accepts motor RPM commands and operates a local feedback control system to drive the motor and cause the propeller to reach the commanded speed. It also reports RPM and battery voltage. The MCU schematic is Figure 15 in APPENDIX B.

The main electronics assembly is the "brain" of the system. The *main1* board contains the power supply (which generates all of the required voltages from the main electronics battery) and also the Analog Devices ADSP-21062 DSP which conducts all of the navigation and system control calculations. The DSP directly receives inertial data from the DQI-105 as well as commands from the *main2* board and determines the RPM commands which should be sent to the four motor control units. These commands are then sent back through *main2* to the MCUs.

The *main2* board contains an Atmel AVR ATmega128 microcontroller which acts as a communications hub for the AFV electronics. It handles packetizing and error detection for the command/telemetry RF links and also decodes the output from the RC command receiver. It accepts commands from the base station and passes them along to either the motor control units directly or to the DSP, depending upon the current AFV operation mode. It forms telemetry packets using the navigation and control data from the DSP along with the RPM and voltage data from the MCUs and

sends them to the base station. It also provides several status LEDs for direct user feedback.

Finally, the *main2* board contains the motor watchdog timer which switches the 12V gate drive signal to all the MCUs. The MCUs cannot operate without this signal because they cannot drive the MOSFET gates to turn them on. Human input (a button press) is required to enable the watchdog timer and it will time out if it does not receive a "keepalive" signal from the main microcontroller within 1.6 seconds.

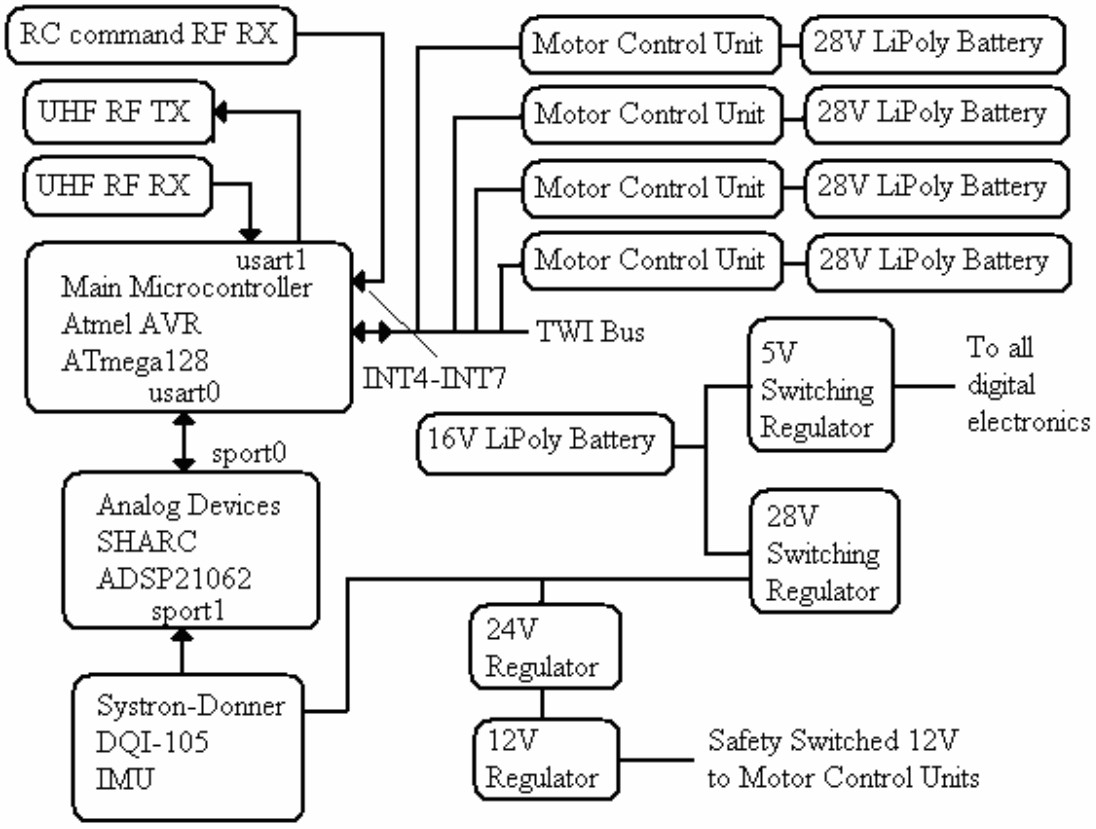


Figure 2 - Overall AFV Electronics Block Diagram

The following sections detail the two main boards and also the motor control units. Directly above is an overall block diagram (Figure 2) of the AFV onboard electronic systems.

1.3.2 Communication Hub PCB

Communication between the AFV and the base station takes place in two directions simultaneously. This happens over an uplink (PC to AFV) on 914.5 MHz at 38400 bps and a downlink (AFV to PC) on 433.92 MHz also at 38400 bps. Uplink commands may be sent at any time but if the AFV is not in idle mode (i.e., if the motors are active), then a "keepalive" command must be sent at least several times per second in order to ensure that the AFV will not enter the failsafe condition (which is different depending on the current mode). The actual timeout period is set to 2 seconds.

The AFV sends only telemetry packets on the downlink and it sends them at 25 Hz. The telemetry packets contain navigation information, orientation data, propeller commanded RPMs, actual propeller RPMs, all five battery voltages, the present AFV mode, and the last four bytes of data written to the controller data EEPROM area (for the base station to verify that they are correct). Both uplink and downlink are protected by 16 bit CRC (Cyclic Redundancy Check) checksums which allow the base station to determine when it has received invalid telemetry and also cause the AFV to reject any corrupted command.

The timing for telemetry packets is derived from the DSP which in turn gets it from the IMU's 600Hz output rate. In other words, the DSP reads data from the IMU at 600Hz. Every 6^{th} set of data, it sends a data packet to the main microcontroller. This is a 100Hz rate. At every 4^{th} one of these packets, the main microcontroller sends a telemetry packet to the base station.

The RF link can be bypassed by plugging a hardwired PC communication cable into the communications debug port on the *main2* board. We have one of these cables (which contains a serial level converter to allow the unit to be directly plugged into a PC serial port). This should allow the PC to communicate exactly as if it were doing so over the RF link but without needing to use the actual RF link. The RF transmitter on the AFV may be disabled by removing both red jumpers which are attached to it. This should be avoided if possible because of the fragility of the jumper pins, but was provided in case debugging had to be performed in an environment where it was not permissible to be transmitting on 433.92 MHz (such as in the RoboCup lab while the RoboCup robots are in operation).

Normally, the base station side of the RF link should consist of one of the AFV transmitter boards (designed by Dr. JinWoo Lee) plugged into a PC. These boards currently are using a Radiometrix TX3 module in their uplink direction which limits the speed to 38400 bps. By upgrading to the TX3A module (which we have), the transmission speed could be increased to 57600 bps, which would allow faster telemetry rates or larger telemetry packets if desired. Because both uplink and downlink use the same UART hardware on both ends (a single PC serial port and a single USART peripheral on the ATmega128 on the AFV), the uplink and downlink speeds must be the same.

Because of the constraints of the Radiometrix RF modules, the transmitted data must maintain close to a 50% ratio of ones to zeros. This is accomplished by both sides of

the link sending 0x55 bytes (in binary, 01010101) during idle periods and also by making every other byte in the data packet the complement of the previous byte.

In the ATmega128, RF communications is handled by USART1 along with buffered, interrupt-driven serial communications routines. The contents of the receive buffer are processed in the main loop of the code. If a CRC error is encountered, this is indicated by turning on the ERROR LED momentarily. The transmit interrupt automatically sends the next character in the transmit buffer or if none is present, the idle byte (0x55). Descriptions of both uplink and downlink packet formats are provided in APPENDIX B (Table 3 and Table 4), along with a brief explanation of each of the commands which may be sent to the AFV.

In addition to the telemetry/command RF link, human velocity commands may be sent to the AFV over the radio-control (RC) command link. This is a standard Airtronics model aircraft radio-control system with a small receiver mounted on the AFV and connected to the *main2* board, and a hand-held control unit with two joysticks. The receiver outputs four PWM signals which correspond to the stick positions. These are decoded by external interrupts 4 through 7 on the ATmega128 in edge-detect mode. The AFV code is set up so that when the Airtronics system is operated in standard "airplane" mode, the Throttle (Channel 1, interrupt 4) channel controls the Z axis velocity, the Aileron (Channel 2, interrupt 5) channel controls the X axis velocity, and the Elevator (Channel 3, interrupt 6) channel controls the Y axis velocity. It is intended that the Rudder (Channel 4) would control the Yaw angle via interrupt 7, but the software for this was not implemented because a Yaw angle of 0 is desired for use with the current decoupled hover controller.

This setup means that if the user keeps the AFV oriented so that MCU number 4 is on the side directly opposite the user (number 4 is on the axis from the AFV to the user and it is the farthest point from the user), then the control stick on the right of the control unit will move the AFV in the horizontal plane (right=right, forward=forward) and pushing the stick on the left forward will cause descent and pulling it toward the user will command a climb.

Commands from the RC link arrive at 50Hz and override the last velocity command from the PC when they arrive. The normal way that the receiver is set up is such that it will not issue commands until the hand-held transmitter unit is turned on, but after that, if the transmitter is turned off, it will continue to issue the last command. Therefore, if you want to be sure that you are executing velocity commands from the PC, you should unplug the Airtronics RC receiver.

If the AFV is operating in controlled flight or autodescent modes (so that the DSP is generating motor commands), then the main microcontroller communicates with all four MCUs at the main control loop rate of 100Hz. If it is operating in any other mode, it communicates with all four at the telemetry rate of 25Hz.

Communication with the MCUs is always initiated by the main microcontroller which acts as a TWI (Two Wire Interface) master. This also means that the main microcontroller provides the clock signal for both reception and transmission to/from the MCUs. Only the data line changes direction.

Each MCU has an address which is programmed into it in its own code. The address 16 is assigned to MCU 1, 17 to MCU 2, 18 to MCU 3, and 19 to MCU 4. When the main microcontroller contacts an MCU, it first sends a TWI start condition, then a data word that contains that MCU's address along with the indication that it wants to send to that MCU (as opposed to receive from it). It then sends the two byte RPM command followed by a stop condition. It then waits 400 microseconds, sends a start condition, then a combination address/receive data word, and then reads out three bytes from the MCU and sends a stop condition.

The greatest priority among all the functions of the AFV electronic systems is to ensure that the motors cannot turn on unexpectedly and that the failsafe mode would always result in the motors ultimately shutting down. The device that attempts to ensure this is the motor watchdog timer on the *main2* board.

The timer consists of a MAX690 Watchdog IC and a TLC555 timer. When the user places the AFV into a mode which enables the motors, the main microcontroller will send "keepalive" pulses to the MAX690. It will continue to do this until either a failsafe mode is entered or the user commands the AFV to idle. If the user presses the motor enable button, a relay will close and the green "will activate" LED will come on. If "keepalive" pulses are being fed to the MAX690, this relay will remain closed and the TLC555 will begin to time a period of about 10 seconds. After this period has elapsed, a second relay will close and will remain closed as long as the MAX690 does not reset. The MAX690 would reset if there were no "keepalive" pulse within 1.6 seconds. Note that the motor enable button is SW2, located near the ribbon cable

connector and beeper, not SW1, located near the ATmega128, which is the main microcontroller reset button, used only for debugging.

When this second relay closes, the yellow "motors active" LED will come on, the beeper will sound, and the 12 Volt gate drive signal will be supplied to all four MCUs.

1.4 AFV Modes and Failsafe Behavior

The main microcontroller, which is on the *main2* board, commands the DSP to enter several DSP modes based upon the current AFV mode. A description of these modes is in Table 8 in APPENDIX B. The main microcontroller also relays velocity setpoint commands to the DSP, which is on the *main1* board. The ATmega128 communicates with the DSP using its USART0. The USART is set up to send and receive 8 bit data words with 1 start bit and one stop bit at 200kbps. The DSP itself provides the serial clock for this, however. The data output line from the ATmega128 is tied not only to the data input on the DSP's SPORT0, but also to the RFS (Receive Frame Sync) pin. The DSP's SPORT looks for a falling edge on the RFS to begin receiving and since each word out of the USART begins with a start bit (which is a logic low) and is preceded either by the idle state (logic high) or a stop bit from a previous word (also logic high), every word starts with a falling edge.

Just like the communications with the base station, the communications with the DSP are packetized, except that there are no complement bytes or CRCs because no errors are expected. We do still, however, begin every packet with a 0xFF byte because this helps to ensure synchronization. This way, if sync were ever lost (i.e., one side or the

other did not recognize a start bit) and were not regained by the next 0xFF byte, it is guaranteed that the start bit after the 0xFF would be the first falling edge within the last 10 bits, so it would have to be recognized. This is done for the same reason over the RF link and there we always send two 0xFFs just to be extra sure since there is the possibility of errors due to the RF link.

Also, the communications are still handled by buffered, interrupt-driven routines in the ATmega128 and the main loop still processes the data when it determines that a full packet has arrived. See Table 5 and Table 6 in APPENDIX B for the packet formats for the communications between the AVR ATmega128 main microcontroller and the ADSP-21062 DSP.

1.4.1 Controller Data Load Function

To aid in feedback controller development and testing, the AFV electronics allow the controller gains and other constants to be changed over the RF link. These are stored in the data EEPROM memory on the main microcontroller and transferred to the DSP on startup or DSP reset. They can also be retransferred by command from the base station.

One of the commands that can be issued over the RF link is to set the address pointer into the AVR's EEPROM and then another command can be used to program four data bytes (the amount required to store one single precision floating point number) into the EEPROM at that location. When the command is issued to set the address pointer, the data at the commanded address is also read into the response bytes in the

telemetry. This allows the PC side software to verify that the correct data has been written by issuing a “set address” command again to the same address soon after writing. Bear in mind that it takes at least 40 milliseconds to write the 4 bytes to the EEPROM. This may also cause a slow-down in the AFV’s telemetry output during controller reprogramming.

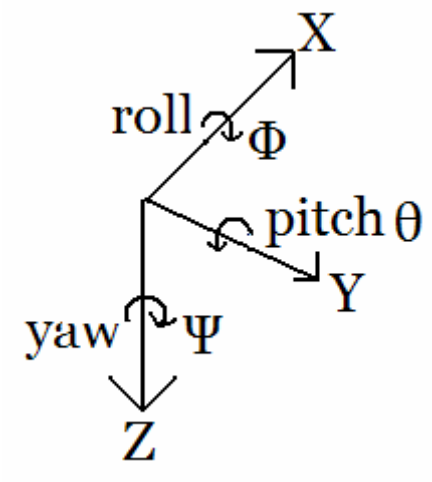


Figure 3 - AFV Coordinate System

The format for the controller data block is given in Table 9 in APPENDIX B. K is the gain matrix from pseudostate: [Φ rate local, Θ rate local, Ψ rate local (these last three are the components of the local angular velocity vector in AFV body frame coordinates), Φ ,Θ ,Ψ ,X velocity ,Y velocity, Z velocity, X velocity error integral, Y velocity error integral, Z velocity error integral ,X acceleration local, Y acceleration local, Z acceleration local (these last three are the components of the acceleration vector in AFV body frame coordinates)] to output [thrust difference 1, thrust difference 2, thrust difference 3, thrust difference 4]. Note that the rates and accelerations in the pseudostate vector are low-pass filtered versions of those in the navigation state vector. They are low-pass filtered according to the time constants in the controller data block. To best understand this whole arrangement, please review

the AFV coordinate system convention (Figure 3) and also the controller data block format (Table 9).

The length of the two RF antennas on the *main2* board is important for proper operation. The one connected to the TX2 (operating at 433MHz) should be very close to 6.25 inches long and the one connected to the RX3 (operating at 914MHz) should be very close to 3 inches long. Both of these were computed using the approximate formula for a quarter wave wire antenna: length(in inches)= 2808/frequency (in MHz), and have tested favorably.

1.5 DSP PCB

The *main1* board contains both the electronics power supplies and the DSP hardware. The basic function of the power supplies should be clear from the block diagram in the section "1.3.1 Overview" above. For any more in-depth questions, consult the datasheets for the components, included on Sean Breheny's MS Thesis CD (see X:\2003 Electronics CD\datasheets*.*). The power supply design comes almost straight from example circuits in these datasheets.

The DSP has to perform four tasks: communication with the main microcontroller, communication with the IMU, navigation, and control. The way that it performs each of these is outlined below.

1.5.1 Communications with Main Microcontroller

This topic was covered to some extent, along with packet structure diagrams, in the above discussion of the main microcontroller. The DSP also uses buffered, interrupt-driven routines to handle this communication and it processes the incoming data in its main loop. Because of its wider range of capabilities, the setup of the DSP's SPORT0 unit to communicate with the AVR's USART0 is a little bit involved. If it becomes necessary to delve deeper into this, consult the comments in the main function of the DSP's code (*DSP_code1.c* which is located on Sean Breheny's MS Thesis CD at X:\2003 Electronics CD\AFV Source Code\latest code*.*) and also the section of the ADSP-2106x DSP Manual on the SPORTs (located at X:\2003 Electronics CD\datasheets\DSP*.*).

1.5.2 Communications with IMU

The IMU sends data using a standard navigation protocol called the AMRAAM format (named after the AIM-120 Advanced Medium-Range Air to Air Missile, which presumably has an IMU using the same format). A photo of two AIM-120s under the wing of a U.S. Navy fighter can be seen below in Figure 4 (taken from http://www.chinfo.navy.mil/navpalib/factfile/missiles/wep-amr.html).

Figure 4 - Advanced Medium-Range Air to Air Missile (AMRAAM)

This AMRAAM format is a synchronous serial format, meaning that the IMU provides a serial clock to the DSP and also the IMU provides a signal that has a rising edge at the beginning of each set of new inertial data words. The IMU transmits its data in blocks of what it calls “autopilot data words” at 600Hz but then also at 100Hz it intersperses these with inertial data words. The DSP uses the autopilot data words exclusively so it ignores the inertial data words, but it does use the 100Hz synchronization signal to determine when the beginning of a 600Hz block is. It then maintains synchronization after that without the aid of the 100Hz signal. The distinction between inertial and autopilot data is not quite clear but it appears as though inertial data (the slower data) is simply the sum of the last six autopilot data words. The datasheet seems to indicate that possibly the inertial data is phase-matched better than the autopilot data (i.e, that the delay from actual sensor to X channel would be very close to the delay from actual sensor to Y channel, for example). It does not seem to make a difference in our application.

On power-up, the DSP will begin receiving data from the IMU the first time that the 100Hz signal goes high. If this happens when the IMU is powering up, this is fine. However, if it happens due to a DSP reset during the middle of IMU operation, then it might synchronize in the middle of a set of data instead of at the beginning. To prevent this, the DSP obtains this initial tentative synchronization and then turns off the SPORT1 (which is used to communicate with the IMU). It waits several milliseconds until it can be sure that the 100Hz signal has gone low again. It then re-enables the SPORT so that it will synchronize on a rising edge of the 100Hz signal. The SPORT has been placed into a mode where it no longer needs the synchronization signal after

it first obtains synchronization, so the 100Hz signal is ignored after this resynchronization has taken place.

Note that no data is ever sent *to* the IMU. The communications channel between the IMU and DSP is unidirectional.

The signals coming from the IMU are not at the 5 Volt logic levels of the DSP's SPORT but rather follow the RS-422 standard, which dictates differential signaling. The AMD AM26C32C IC converts these to the 5 Volt logic levels needed by the DSP. The 150 Ohm resistors in this section of the circuit are termination resistors that are part of what the IMU's output drivers expect to see for proper operation.

The data received from the IMU is received in 24 bit words. Each one contains an 8 bit tag field which describes what kind of data it is (i.e., whether inertial or autopilot and which channel it is from, such as X acceleration or Theta rate) and 16 bits of signed integer raw data. The routine which receives the data from the IMU is interrupt driven but not buffered. Instead, when a new data word arrives, it simply looks for which type it is using the tag field and places the new value into the appropriate variable. When the last piece of data for a particular data set arrives (Ψ rate), it sets a flag indicating that the main loop should process this data through the navigation routine.

1.6 Motor Control Units

Because they produce the thrust needed to lift the vehicle and also because they are the control actuators, the motor drive systems are the most important electronic systems onboard. All the other onboard systems have the ultimate purpose of determining and providing RPM commands to the four motor control units. The following sections detail the specifications, interfaces, and firmware functions for these units. There is also a brief discussion of the internal feedback control law and a simple dynamic model of the system which can be used for outer-loop control design.

Each motor control unit(MCU) receives 5 Volt power for internal electronic functions from the main AFV electronics assembly via the interface cable. The power for the motor comes directly from the 14 cell Lithium Polymer battery pack attached to each motor control unit. This pack is arranged in a 7 series by 2 parallel configuration as discussed in the "1.2.2 Batteries" section. The heart of each motor control unit is an Atmel AVR ATmega128 microcontroller running at 16MHz. This microcontroller continuously performs three parallel tasks: motor electronic commutation, communication with the main AFV electronics assembly, and operation of the local motor feedback control loop.

The specifications for the motor control units are given in APPENDIX B (Table 13) as well as the connector pinouts and commutation information on the MaxCim MaxN32-13D motors (Table 11). Figure 15 is a schematic diagram of the motor control units.

1.6.1 Task 1: Commutation

A brushless DC motor is essentially the same device as a brushed DC motor, except that the process of changing the current flow path through each of the windings is performed externally and electronically instead of mechanically. This process is known as commutation and is performed by the ATmega128 microcontroller aboard each MCU. The process is performed entirely by the main loop of the MCU code which simply looks at the output of the three Hall effect sensors from the motor each main loop cycle. Using a set of C CASE statements, it determines the correct output to drive the three phases in order to provide either forward or reverse torque as is currently being requested by the local control loop. If this output is different than the previous output, it briefly drives all three phases to the open (Z) state and then applies the new correct output. This brief open period is done to ensure that at no time will both high and low side MOSFETs be turned on for any phase, which would result in a shorted battery for a brief instant.

The MOSFET high and low-side switches are driven by an International Rectifier IR2133S gate driver IC. A logic network along with some delay components connect the output of the microcontroller to the driver IC's logic inputs. The network of gates combines the commutation outputs with the PWM output. It is set up so that the PWM does not affect any phase that is currently being driven low by the commutation code, but any that is being driven high will be high when the PWM output is high and low when it is low, so that the motor will be switched between driven and brake states during the on and off stages, respectively, of the PWM. Combined with the motor inductance and high PWM frequency, this yields a low-ripple current in the motor windings and naturally applies breaking when the PWM duty is set to a lower level than that which would sustain the present motor RPM.

The purpose of the delay components is to extend the IR2133S's internal crossover switching delay so that the PWM switching does not cause cross-conduction from high-side to low-side MOSFETs, shorting the battery.

1.6.2 Task 2: Communication

Communication between the MCUs and the main AFV electronics occurs via the Two Wire Interface (TWI) serial bus, which is Atmel's name for what is more commonly called an Inter-Integrated Circuit (I^2C) bus, a Philips N.V. trademark. Each communication cycle is initiated by the main microcontroller, which at all times remains the TWI master. Each motor control unit has a unique address and the main microcontroller first sends two bytes to the address of the MCU it wishes to communicate with.

The two bytes are the least significant and most significant bytes (in that order) of the 16 bit signed integer RPM command. The MCU's microcontroller then takes up to a little less than 400 microseconds to prepare to respond to this command by sending two pieces of information: the actual propeller RPM and the battery voltage. It loads these into a 3 byte buffer in the following order: LSByte of RPM, MSByte of RPM, ADC codeword (1 byte) from battery voltage measurement. When the main microcontroller then initiates the receive part of the cycle, also addressed to the same MCU microcontroller, the MCU responds by sending the contents of this buffer.

A failsafe is set up in the MCU code so that if no commands are received within about 0.9 seconds, the RPM setpoint will be set to 0. This is to prevent a situation where communication fails between an MCU and the main electronics, causing a safety problem.

1.6.3 Task 3: Control

The local feedback control loop is handled by a timer interrupt which occurs at 244Hz in the ATmega128 on each MCU. The control method is a simple proportional-integral controller. The encoder's two output channels are XORed in hardware, yielding 1024 pulses per propeller revolution, and this output is counted by the microcontroller's timer 1. Direction information is also derived from the encoder by a D-type flip-flop and fed to the microcontroller's pin PA4.

Each time the timer interrupt happens, the number of encoder pulses since last interrupt is saved along with the current motor direction from the flip-flop. The error between this and the setpoint is obtained and is also fed into an integral accumulator. This accumulator is limited to prevent excessive integral windup. The integral and proportional error terms are then multiplied by I and P gains and summed to produce the intended PWM output duty. This is checked against absolute limits and also against limits that are based on the RPM to limit motor current. Finally, the output is fed to the hardware PWM generator.

1.6.4 Control Design

Using the simple electromechanical motor model introduced in the section "1.2.3 Motors and Propellers", we obtain the following motor transfer function from effective input voltage to propeller output shaft speed in radians/second:

$$\frac{\Omega(s)}{Vin(s)} = \frac{\frac{Ki}{JR}}{s + \frac{KiKv}{JR}} \qquad \textbf{Equation 1-4}$$

Considering that $K_i=K_v$ and letting $a=K_i/(J*R)$ and $b=K_i$, we have:

$$\frac{\Omega(s)}{Vin(s)} = \frac{a}{s+ab} \qquad \textbf{Equation 1-5}$$

If we close the loop and apply PI control, we obtain the following transfer function from commanded angular speed to actual angular speed:

$$F(s) = \frac{G_p as + G_I a}{s^2 + (ab + G_p a)s + G_I a} \qquad \textbf{Equation 1-6}$$

and from commanded angular speed to control effort voltage:

$$E(s) = \frac{G_P + \frac{G_I}{s}}{1 + \frac{G_P as + G_I a}{s(s+ab)}} \qquad \textbf{Equation 1-7}$$

where G_p and G_i are proportional and integral gains, respectively, in Volts per radian per second and Volts per radian. By considering various cases, it is easy to see that sufficient conditions for stability of F(s) are that $G_p>-b$ and $G_i>0$.

A reasonable starting point for this control design would be the following: First, determine the maximum expected disturbance-induced command fluctuations. We

could then restrict ourselves to stable, no-overshoot controllers and find the fastest such controller which did not saturate the system when experiencing the expected command excursions.

Although we never used rigorous methods to determine the expected hover command variations, simulator experience indicated that somewhere around 1 Newton thrust variation would be reasonable. When computed around the expected hover point of 415 radians per second, this corresponds to variations of 15 radians per second.

The condition for no-overshoot allows us to link the proportional and integral gains so that we no longer have two completely free parameters to play with:

$$G_I < \frac{(ab + aG_P)^2}{4a}$$ **Equation 1-8**

It is also the case, from the Initial Value Theorem applied to E(s), that the peak control effort for a step input is simply G_p times the command.[15] We can then first compute the largest proportional gain that does not saturate under the expected conditions and then choose the largest integral gain which does not yield overshoot. This will yield a faster controller than any with a lower G_p since when G_i is set to the no-overshoot limit, the closed loop time constant is given by:

$$\tau = \frac{2}{(ab + aG_P)}$$ **Equation 1-9**

The hover point requires an effective voltage of about 16.7 Volts, and the maximum voltage when the cells are at their 3.8 Volt nominal state is about 26 Volts, so our

control effort range is 9.3 Volts. This would give a maximum G_p=0.62. Using the values K_i=K_v=0.0255, R=0.25 Ohms, and J=0.0011 kilogram meter^2, we would obtain G_i=9.66.

After several runs of motor system testing and evaluation, we decided to back off the proportional gain to 0.33 due to the fact that higher proportional gains were interacting with the relatively low encoder resolution (14 RPM) to produce very jittery movement at lower RPM ranges. This would yield a predicted optimum G_i=2.92 but through testing we found that the actual optimum point (with regard to placing the system right on the no-overshoot boundary) was G_i=1.7. These discrepancies are well within expectations given that several significant factors were unmodeled, including propeller drag, belt flex, encoder resolution, and the nonlinear PWM to voltage relationship.

Evaluating the theoretical closed-loop transfer function yields:

$$F(s) = \frac{30.6s + 157}{s^2 + 33s + 157} \approx \frac{27}{s + 27}$$ **Equation 1-10**

We cannot use the theoretical time constant formula because we have not set G_i to the predicted optimum. However, the above theoretical transfer function coincides remarkably well with the measured time constant of 40 milliseconds.

Note that one must convert the G_p and G_i computed here into raw units before comparing with those found in the source code *MCB_sens2.c*. The conversion for the proportional gain is as follows:

$$G_{\text{p\_raw}} = G_{\text{p\_theoretical}} \left(\frac{1 \text{ radian/ second}}{9.5541 \text{ RPM}} \right) \left(\frac{256}{26 \text{ Volts}} \right)$$ **Equation 1-11**

where the 256 comes from the fact that the PWM duty is represented by an 8 bit code from 0 to 255 corresponding to 0 to 100%. Interestingly, this conversion is very close to 1, so we chose 1/3 as the raw proportional gain in the microcontroller code. The conversion for the integral gain is as follows:

$$G_{I\_raw} = G_{I\_theoretical}\left(\frac{6.28\,\text{radians}}{1024}\right)\left(\frac{1}{14.3333}\right)\left(\frac{256}{26\,\text{Volts}}\right) \qquad \textbf{Equation 1-12}$$

The number 256 appears for the same reason as above, 1024 appears because it is how many counts the microcontroller sees from the XORed encoder output every revolution, and the number 14.3333 comes from the fact that the integral accumulator is integrating RPM instead of encoder counts since last cycle, and the conversion between those is very close to 14.3333 when running at s 244 Hz control loop iteration rate. This would yield 1/140, the actual value which is in the code is 1/143, which differs because it simplifies the computations in the code.

1.6.5 Dynamic Model

A simplified dynamic model of the MCU system would be that it momentarily saturates for a step of approximately 30 radians per second (287 RPM or about 2 Newtons thrust differential about hover), has a variable latency between 1 millisecond and 4 milliseconds, and has the following transfer function for small command variations:

$$F(s) = \frac{25}{s + 25} \qquad \textbf{Equation 1-13}$$

1.7 Navigation Estimation

Navigation is the task of updating the DSP's AFV state estimate using newly arrived IMU data, continually trying to estimate the AFV's orientation and global velocity. The DSP uses a right-handed coordinate system where Z is down. (See Figure 3) The roll angle (Φ) is around the X axis in the direction you would expect by using the right hand rule with the thumb pointed along the positive X direction. The pitch angle (Θ) is around the Y axis and yaw (Ψ) around Z, both with the expected directions according to the right hand rule. This coordinate system is obtained from the IMU data (which uses a "Z up" coordinate system) by reversing the signs of the Y and Z accelerations and angular rates, preserving a right-handed coordinate system. The three Euler angles are computed so that applying rotation matrices in the Roll-Pitch-Yaw sequence results in a correct orientation.

The navigation code also provides local angular rate data, global Euler angle derivative data, and local acceleration data. A block diagram of the way that the navigation code works is given below in Figure 5.

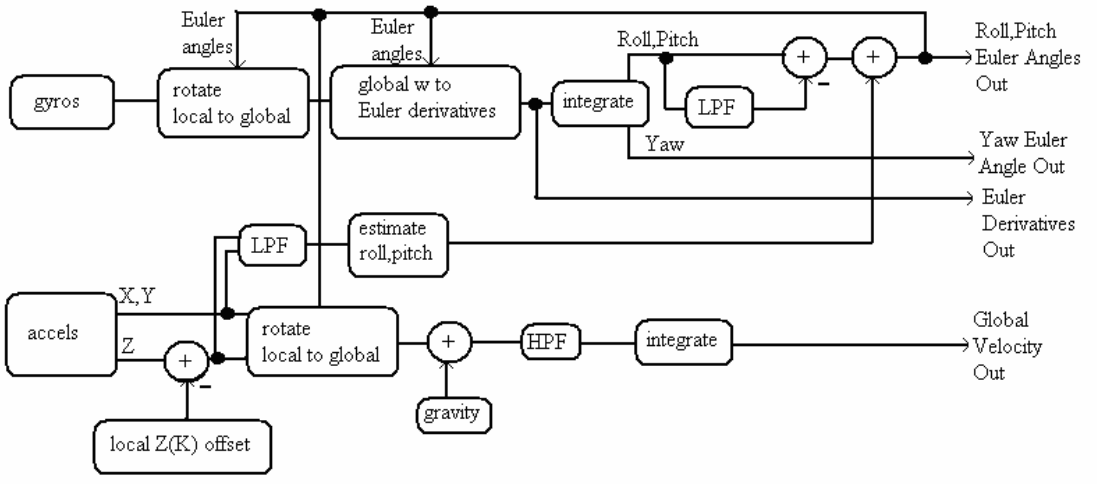


Figure 5 - AFV Navigation Software Block Diagram

To obtain the best performance possible from the IMU data, the navigation code makes some assumptions about the motion of the AFV. It assumes that the long term (about 20 seconds) average velocity will be near zero, since this initial design is intended only for hover. By doing so, it can use the long term average acceleration to continuously estimate the roll and pitch angles. These are then combined with the integrated gyro data, which has been high pass filtered by subtracting off its long term mean, to obtain the final roll and pitch angle estimates. The yaw angle is simply integrated directly, though, since there is no other yaw angle reference.

The navigation code also high pass filters the acceleration data before integrating to get velocity. This further reduces the effect of drift errors and provides reasonably good data when the long term average velocity is near zero.

The "local Z(K) offset" mentioned in the diagram is obtained during the calibration routine. K refers to the local vertical axis in the AFV's body reference frame. When the DSP is first powered on or whenever it is reset, it performs a quick 2 second IMU calibration. During this time, it averages the outputs of all six sensors. It then assumes that the average rate values are the offsets on the three rate channels and it uses the three acceleration averages to compute the roll and pitch angles, assuming there is no intrinsic offset. It then works backwards to find out what the actual intrinsic offset is for the Z channel (it is mathematically impossible to do so for the X and Y channels). This becomes the local K offset.

The calibration routine also takes care of initializing all of the variables in the navigation code, such as the initial state of the Euler angle integrator and Euler angle LPF to the initial estimated roll and pitch angles. It assumes the initial yaw angle is zero.

The user can also command a longer 20 second calibration over the RF link. This is necessary before flight and is best done after the IMU has warmed up for a few minutes. This calibration works the same way but obtains more accurate estimates by averaging for a longer period of time.

The DSP can be placed in one of four different modes: SHORTCAL, LONGCAL, CONTROLLER ON, and CONTROLLER OFF. SHORTCAL is the mode during the initial 2 second calibration. The DSP only enters this upon startup. LONGCAL is the mode for the 20 second calibration. The DSP returns to CONTROLLER OFF mode after both calibration modes have finished. In CONTROLLER OFF mode, the data sent from the DSP to the AVR contains the computed hover RPM in the spaces allocated for the motor commands. The CONTROLLER ON mode will be discussed in the next section. At all times when it is running, the DSP blinks its status LED. The other LED remains off at all times except when in CONTROLLER ON mode.

Useful references for design of the navigation system are given in reference [16].

1.7.1 Note about Eryk Nice's Estimator

Approximate figures for the AFV's navigation performance using the system described above can be found in Table 12 in APPENDIX B. Eryk Nice implemented a Square-Root Sigma Point Filter state estimator for the control system which has significantly better performance by estimating the accelerometer and rate gyro bias levels. It does this in part by using the human control input as an additional measurement. The details of this estimator can be found in Chapter 5: SIMULATION DEVELOPMENT AND VERIFICATION of Eryk's M.S. Thesis.[17]

1.8 Airframe Design

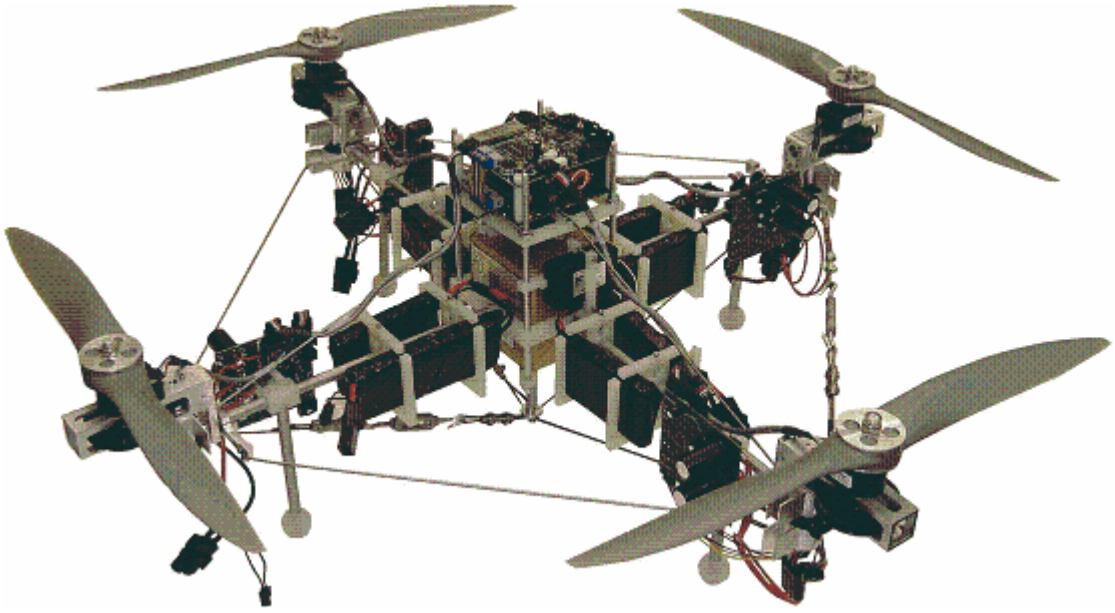


Figure 6 - AFV Photo (©2003 IEEE – see reference 2)

Eryk Nice performed the AFV airframe design, based on a light cable-tensioned structure somewhat reminiscent of the rigged biplanes of the First World War. This design took advantage of the fact that while very high stiffness is necessary to keep the structural resonant frequencies high to reduce vibration, ultimate strength need not be very high. Rather than use a heavy solid structure which was both stiff and strong,

Eryk's design traded off strength for weight while retaining stiffness. The IMU and two main electronics PCBs were mounted in the center, with the IMU case doubling as a structural component. At each corner of the box-like structure was placed a motor control unit with motor, belt-drive, and propeller. The batteries fill in the remaining space along each strut. Figure 6 shows the overall structure. More information can be found in Chapter 4: FABRICATION, ASSEMBLY, HARDWARE TESTING, AND REDESIGN under subsection STRUCTURE in Eryk's thesis[17].

1.9 Vehicle Control Design

The final task of the DSP is to take the navigation data along with the commanded velocities and compute motor RPM commands. Currently, the controller is set up to assume a quadratic model for the RPM to thrust curve. It is given a nominal thrust and thrust coefficient. It assembles a length 15 pseudostate vector from the navigation information and multiplies this by a 4x15 static gain matrix K. This yields four thrust difference values which are then added to the four nominal thrust values. The square root of this is taken and then multiplied by the thrust coefficient, yielding four RPM commands. The signs are changed to reflect the fact that two of the props counter-rotate. These are then the four commands send to the AVR to be sent to the MCUs.

The pseudostate vector is described above in the section "1.4.1 Controller Data Load Function". Most of the quantities in it come straight from the navigation code. The data that is used for derivative-type control (local rates and local accelerations) is first low pass filtered by single pole filter stages with the assigned time constants (which can be changed by loading new controller data). The only other derived quantity is the integral of global velocity error, which is simply the error between measured global

velocity and commanded velocity, integrated. This is provided in case the user wanted to attempt integral control on the velocity.

The controller also applies hard-coded saturation limits on the thrust for each prop at a maximum of 35 Newtons and a minimum of 1 Newton. The upper limit is actually higher than the props can produce and serves mainly to prevent a bad controller from producing ridiculous values. The lower limit is useful, though, to prevent the props from being commanded reverse RPM, which the MCUs are fully capable of delivering. This is not desirable for a hover controller but may be useful later on. If so, these limits will have to be removed from the code

Of course, there are many possible methods of computing the gain matrix K. Eryk Nice was primarily responsible for the AFV outer-loop control design and used a linearized model of the AFV for controller synthesis and a full nonlinear Simulink model for testing of the controller before trying it on the actual vehicle. When the AFV model is linearized about hover, the result is a decoupled system where the three axes are independent and the two angles about the horizontal axes affect only one linear direction each. Because of this, early control designs were simple PID controllers applied to each axis. A little later we moved to LQR to allow easier adjustment of intuitive parameters with the guarantee that we would remain within the set of stable controllers[9].

Because I was only involved in the initial stages of flight testing, the primary concern was with basic problems such as obtaining good navigation data in the presence of IMU drift. Control design was not the primary focus as it was easy to design a stable

controller for hover with good enough performance for basic testing. Better control design for maneuvers or formation flight was left to future improvements.

For more information on control design and AFV modeling, see [17] and APPENDIX C in this document.

CHAPTER 2

DARPA SIMULATION SUBPROJECT

2.1 Definition of Terms

A – area in meters^2

EZNEC (EZ Numerical Electromagnetic Calculator) - a computer antenna simulation program (see http://www.eznec.com)

$\underline{K}$ – antenna element cross coupling matrix in Ohms. When multiplied by a column vector of the phasors of the currents in each element, the result is a vector of the voltage phasors seen at each element's feedpoint.

M file – MATLAB script

OAV (Organic Aerial Vehicle) – a ducted-fan hovering unmanned aerial vehicle produced by Honeywell.

PAA – Phased Array Antenna

P_L – power radiated by antenna element L in Watts

r – radius in meters

Steradian – unit of solid angle. If one imagines a spherical shell of radius r around a point p as well as a shape being projected from point p onto the inside of the shell, then the area of the projection of the shape equals U times r^2. U is therefore a measurement of angular area.

U – solid angle in steradians

$\overline{V}$ – vector of the voltage phasors at the feedpoints of several antenna elements

WASP (Wonderfully Agile Sensor Platforms) – a DARPA program to investigate the use of unmanned aerial vehicles to carry sensors useful to soldiers.

Θ – elevation angle in polar coordinates

Φ – azimuth angle in polar coordinates

2.2 DARPA Requirements

In late 2002, we began a project for the DARPA Wonderfully Agile Sensor Platforms (WASP) program. This program explored various ways of using unmanned aerial vehicles (UAVs) to obtain information in military or law-enforcement applications such as hidden transmitter location and synthetic-aperture radar mapping. It specifically focused on the Honeywell Organic Aerial Vehicle (OAV) [18], which comes in several different sizes and payload capabilities. The DARPA WASP administrator was interested in a study of using OAVs as phased-array antenna platforms. We were to provide the tools necessary to accomplish this study.

2.3 Deliverables

We provided DARPA with two items: simulation software written in MATLAB and documentation on the overall problem of using OAVs as antenna platforms and on using the simulation software. The software can take two forms of input: either real-time OAV position data through a network interface, or pre-recorded OAV trajectories from files on disk.

These data include both actual and sensed OAV position. The actual position is needed to show the true antenna radiation pattern. The sensed OAV position is where the OAV's themselves think they are, which is the only position data which can be used by the phase compensation algorithm. The OAV's sense their position using a combination of GPS and inertial navigation.

Using this data, the software displays the antenna pattern which could be produced using phase compensation if the OAVs each had radio transmitters feeding half-wave

dipole antennas. It also computes and displays the signal strength which would be achieved at a distant base station receiving a signal from the OAV formation, as well as the ideal (no position errors) antenna pattern and the pattern with position errors but without compensation.

2.4 Problem Overview

We first conducted a study using EZNEC to determine the permissible assumptions for the problem at hand. We knew from the specifications of the OAVs that we would be working with physical spacing of about 5 to 10 meters between vehicles, in clusters of 10 vehicles or less, and linear antennas (such as ½ wave dipoles). Using higher frequencies would merely narrow lobe diameter without increasing gain, so we would want to use the lowest frequency which would be permitted by the antenna physical size constraints, decreasing sensitivity to position error. This is roughly 100 MHz.

We tried two principal phase compensation schemes: a full search of the phase space for the optimum, and simply adding up the element contributions in phase at the receiver (base station). The latter was much less computationally intensive and yielded results which typically differed by only 1% from the more difficult method. This can be attributed to the fact that the elements are more than a wavelength apart and have less coupling than they might otherwise have.

However, we did find that coupling still had to be taken into account in selecting driving current amplitudes. If we simply fed all elements with the same current amplitude, some would be emitting much more power than others. We found that varying current amplitude from element to element was not very detrimental to the pattern shape, so it was best to maximize the overall transmitted power by setting the

currents to cause each element to emit its own maximum power (which would in reality be limited by the specifications of the transmitters aboard each OAV).

2.5 Note on Array Type Choice

In all of the discussions of arrays in this paper, endfire type arrays are used. These were chosen because of the fact that they inherently produce a single main beam in one direction. While they were not explored in this research effort, broadside arrays could also be a possible type of antenna for communication between groups of unmanned aerial vehicles and a base station. These would offer the potential benefit of reduced element to element interaction if implemented using dipoles which were laid end to end. It would be worthwhile exploring this in a future research effort.

2.6 Software Implementation

Because it is a standard environment for engineering computation and allowed rapid development, we chose to write the simulation software in MATLAB. Recent versions of MATLAB allow sophisticated graphics visualization methods and easy user interface implementation. The core of the PAA program was generated automatically by MATLAB's *GUIDE* function.

The PAA software must take input of the position of several OAVs, compute the resultant antenna pattern without any correction and also with phase correction, and display this information. It can do this either in real-time using input data from a TCP/IP network, or from a pre-recorded file stored on disk.

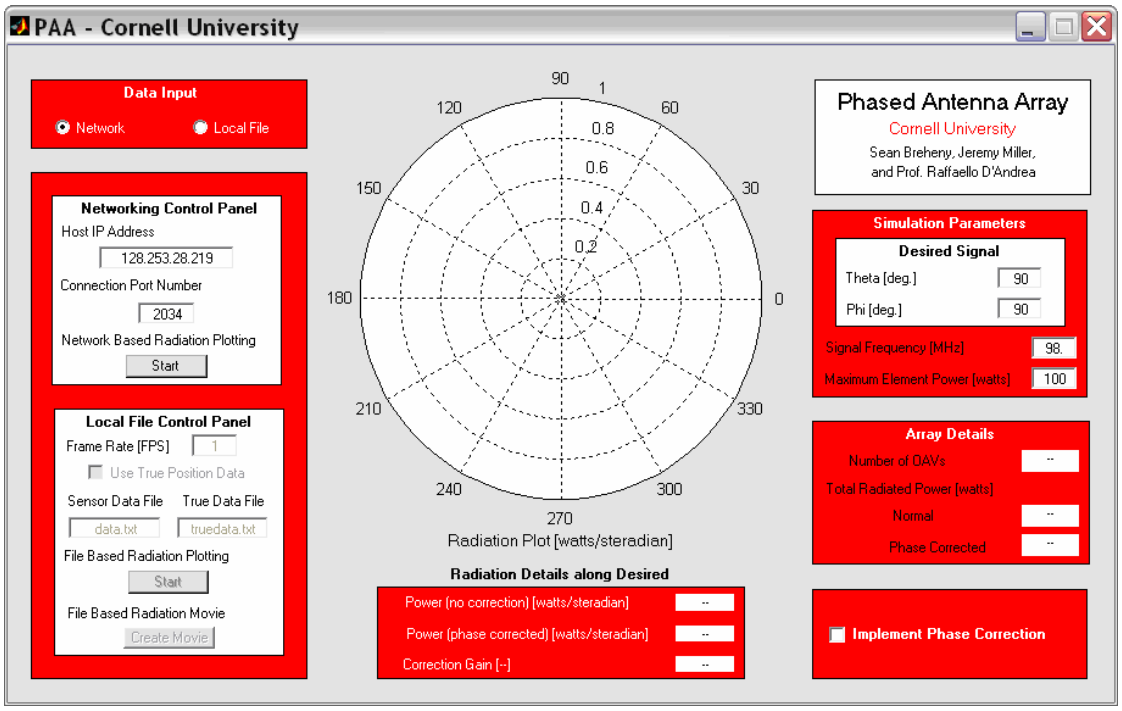


Figure 7 - WASP Phased Array Antenna Simulation GUI

In the case of data from a file, it is capable of using true position data for the pattern computation and sensor-based position data for the phase correction, since of course imperfect sensors are a constraint in real life. When taking data in real time over the network, the data arrives in GPS format (latitude and longitude) so the software must also translate this to local Cartesian coordinates, which is performed by *Spherical_to_cartesian.m*.

An image of the user interface can be seen in Figure 7. On the left-hand side is the control panel for the two operating modes (network real-time and file-based). On the right signal frequency and maximum element power can be specified, as well as desired signal direction (both Θ and Φ since the elements are dipoles and not isotropic radiators – Θ is the dimension which is plotted since it is the angle within the horizontal plane which the OAVs occupy. The dipoles are oriented perpendicular to this plane). Using this desired signal direction, the software displays the power in

Watts/steradian with and without correction in that direction numerically at the bottom of the UI. The center is a linear polar antenna pattern plot, both corrected for position errors and uncorrected, in Watts/steradian, auto-scaled to the appropriate magnitude.

Steradians are a concept often used in antenna analysis. A steradian is a measure of solid angle or the angular area subtended by a two-angular-dimension region. If one is standing at the center of a spherical shell and projecting an image onto the inside of that shell, the physical area of the projected image will depend on the radius of the spherical shell. However, it is not arbitrarily dependent on radius, it will be equal to

$$A = Ur^2 \qquad \textbf{Equation 2-1}$$

where U is a constant for the particular shape being projected and U is in steradians. The maximum value of U is 4π for the entire inside of the spherical shell.

Because there are many calculations involved in the simulation, the software makes use of MATLAB's vectorization feature. This is to say that we set up as many of the computations in vector/matrix format as possible and this causes MATLAB to perform some internal optimizations which it would not otherwise do if everything were coded in the form of FOR loops.

Execution begins with the file *PAA_GUI.m*, which handles the user interface, calling the necessary functions when the user issues commands. The simulation loop itself is triggered either by the arrival of network data (at which time a new frame is generated) or by a timer (set by the FPS box on the left-hand control panel). The loop is contained in either *PAA_Main.m* or *PAA_Main_With_True.m*, depending on whether true position data is being used. This uses *U_Pattern.m* to compute the antenna pattern, *phase_determination.m* to perform phase correction, and

current_magnitude.m and *Kinterpolation.m* to scale the currents to make sure each element meets the element max power specification.

Kinterpolation computes the K element coupling matrix for the present array geometry. It uses a simple lookup-table with MATLAB's *interp1* function to perform linear interpolation between table entries. These data were gathered by first obtaining data for one pair of ½ wave dipoles at various spacings in EZNEC and then fitting a function to that data. This function was then evaluated at many points to fill the tables in *Kinterpolation.m* The file makes the assumption that the dipoles are parallel to each other and their feedpoints are coplanar (so that their interaction can be fully captured by simply their distance from each other in that plane). This is a fairly good assumption as there is not much difference in coupling until the antennas deviate significantly from the plane (more than 30 or 40 degrees angle between elements). This is because the pattern function for ½ wave dipoles varies very little near Φ=0.

Phase_determination.m simply computes the difference in the distance between each element and the target (an imaginary point at infinite distance along the desired signal direction). Because this point is so far away, an approximation can be made to simplify this distance formula, ignoring element movement perpendicular to the desired direction.

Current_magnitude.m attempts to solve the equation

$$P_L = \mathrm{Re}\left\{ i_L^* \left(\sum_{m=1}^{n} K_{Lm} i_m \right) \right\} \qquad \textbf{Equation 2-2}$$

discussed further in CHAPTER 3, for the element current magnitudes given K, the phases of the currents, and the power of each element (set equal to the maximum

allowed element power to maximize transmitted power). The equation is a quadratic set of N equations in N unknowns. Rather than solve this directly, the M file linearizes about the point where all the currents are equal to what they would be if each element were radiating the maximum power and there were no interactions (K were purely diagonal). It then solves the linear equation and evaluates the original equation at the solution value of the current magnitudes. It then scales all the currents by the same factor so that the power of the maximum-power element is equal to the maximum power specification. This is necessary because the linearized solution may be slightly incorrect.

Pattern_function.m contains lookup tables for the pattern of an individual element. This is used by *U_pattern.m* in computing the overall antenna array pattern. If the number of gradations used per angle is equal to 400 or 100, then the lookup tables are used. Otherwise, *Pattern_function.m* must compute the required value each time.

U_pattern.m is the core of the simulation while also being one of the simplest M files. It first computes the contribution of each element to the pattern in a particular direction for an array of isotropic elements. It does this for all of Θ and Φ. It then multiplies this by the element pattern function to obtain the actual array pattern. Because the antenna pattern is the spatial Fourier transform of the current distribution, and the current distribution of an isotropic element is a delta function, the product of the pattern of an isotropic array with the pattern of an individual element (when the elements are identical) yields the correct overall pattern.

For additional information about the software please see the documentation for this project written by Jeremy Miller and Sean Breheny.[4]

CHAPTER 3

MATRIX METHOD OF COMPUTING ARRAY ELEMENT INTERACTION

3.1 Definition of Terms

d – distance between feedpoints of parallel half-wave dipole antennas, in wavelengths

I – rms magnitude of the current at the feedpoint of an individual antenna element in Amps

$\underline{K}$ – antenna element cross coupling matrix in Ohms. When multiplied by a column vector of the phasors of the currents in each element, the result is a vector of the voltage phasors seen at each element's feedpoint.

P – power radiated by an individual antenna element in Watts

P_L – power radiated by antenna element L in Watts

R – real part of antenna feedpoint impedance in Ohms

$\overline{V}$ – vector of the voltage phasors at the feedpoints of several antenna elements

V_L – element of $\overline{V}$

3.2 Apparent "Violation" of Conservation of Energy

When one first begins to study antenna arrays, he begins with the case where each element can be considered to produce the same radiation pattern and have the same input impedance as if there were no other elements around it. Typically, the input impedance of an antenna is calculated by assuming that it is driven by a current source. The current distribution is then found using the boundary conditions imposed by the geometry of the conductor. From this, a far-field radiation pattern can be computed and integrated over a sphere surrounding the antenna to obtain the total radiated power. One can then work backwards to get the antenna impedance using

$$P = I^2 R$$ **Equation 3-1**

A difficulty arises if all these same assumptions are maintained while bringing two antenna elements close together (within approximately 10 wavelengths). Because the electric fields of the two antennas add by superposition, the total radiated power for two elements very close together would seem to be four times the power of a single element (since power is proportional to the electric field strength squared and the electric field strength is twice that of a single element). This is clearly impossible since it would involve a violation of conservation of energy. The error lies in the assumption that the element input impedances remain the same when in the presence of other elements.

3.3 Element Self and Mutual Coupling

In reality, all of the elements in an array are coupled and the current in each element produces a voltage in every other element. In fact, the reason for an element having an input impedance at all is that the current in that element produces an electric field around the element, which in turn induces a voltage in the element which is seen at the input terminals. Input impedance is just a special case of coupling which is in fact self-coupling.

Determining the exact nature of the coupling among elements is a very difficult problem but we can easily arrive at a general formulation along with an approximation for the actual constants involved, which is good enough for array synthesis.

3.4 Matrix Formulation

Element to element coupling can be neatly summarized in a matrix formulation, according to the following definitions:

P_L is the power emitted by antenna element L.

A coupling coefficient matrix (where the elements are complex):

$$\underline{K} = \begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,n} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ K_{n,1} & K_{n,2} & \cdots & K_{n,n} \end{bmatrix} \quad \textbf{Equation 3-2}$$

Note that, by reciprocity:

$$\underline{K} = \underline{K}^T \quad \textbf{Equation 3-3}$$

The impedance of an element by itself:

$$\underline{K}_{L,L} = Z \quad \textbf{Equation 3-4}$$

A vector and a matrix representation of the element current

$$i_L = |i_L| e^{j\phi_L} \text{ (rms current)} \quad \textbf{Equation 3-5}$$

$$\bar{I} = \begin{bmatrix} i_1 & i_2 & \cdots & i_n \end{bmatrix} \quad \textbf{Equation 3-6}$$

Then, the power emitted by element L is:

$$P_L = \mathrm{Re}\left\{ i_L^{\,*} \left(\sum_{m=1}^{n} K_{L,m} i_m \right) \right\} \quad \textbf{Equation 3-7}$$

3.5 Approximation of Matrix Elements

As a means of performing calculations for array synthesis and back-of-the envelope analysis, we conducted a number of tests using EZNEC and extracted the following approximation to the coupling constants from the simulation output data: For ½ wave dipoles oriented parallel to each other with feedpoints in the same plane, where d is the distance in wavelengths between the centers of each pair of elements (L and m):

$$K_{L,m} = K_{m,L}$$ **Equation 3-8**

$$\left|K_{L,m}\right| = \frac{1}{0.0158d^2 + 0.0297d + 0.0121}$$ **Equation 3-9**

$$\angle K_{L,m(radians)} = 5.9424d - 0.8678$$ **Equation 3-10**

CHAPTER 4

MICRO-ANALYSIS OF MULTI-VEHICLE PHASED ARRAY PERFORMANCE

4.1 Definition of Terms

A_{EFF} – effective area of an antenna in meters^2

b_0 – conversion from motor applied Volts to propeller rotational speed in radians per second, multiplied by b_1

b_1 – reciprocal of motor system time constant in 1/seconds

c – speed of light in meters per second

c_k – various coefficients used in vehicle equations of motion.

d – horizontal distance from vehicle center of mass to center of each rotor in meters

D – directivity of an antenna

D_o – directivity of an antenna array before introducing element position errors

f – frequency in Hertz

F – force in Newtons

g – gravitational acceleration in meters per second^2

h – vertical distance from the vehicle center of mass to rotor plane in meters

J_r – rotor moment of inertia in kilogram meters^2

K_l – linearized coefficient from rotor rotational speed in radians per second to lift in Newtons

K_{ln} – nonlinear coefficient from squared rotor rotational speed in (radians per second)^2 to lift in Newtons

K_t – linearized coefficient from rotor rotational speed in radians per second to torque in Newton meters

K_{tn} – nonlinear coefficient from squared rotor rotational speed in (radians per second)^2 to torque in Newton meters

M – total vehicle mass in kilograms

m_c – mass of vehicle (except for mass of motor/rotor/beltbox units) in kilograms

m_r – mass of a single motor/rotor/beltbox unit in kilograms

Q – link quality, the ratio of the power transmitted to the power received in a transmitter-receiver antenna pair, evaluated at a separation of $\frac{1}{2\sqrt{\pi}}$ meters

R_{old} – the original (error-free) distance from an antenna element to the intended receiving antenna in meters

R_{new} – the distance from an antenna element to the intended receiving antenna in the presence of position errors in meters

W_x, W_y, W_z – local wind speed in miles per hour in the x,y, and z dimensions

$\Delta\varphi$ – amount of phase shift which must be applied to an element for phase correction in radians

Θ – Euler angle about the Y axis in radians

σ – RMS position error (in wavelengths) of the elements in an array in each spatial dimension, assuming that the RMS error is the same in each dimension.

τ – torque in Newton meters

Φ – Euler angle about the X axis in radians

Ψ – Euler angle about the Z axis in radians

ω – rotational speed of helicopter rotor in radians per second

ω_h – nominal hover rotor rotational speed in radians per second

$\omega_{0\text{-}3}$ – rotational speeds of each of the four rotors (0 through 3) in radians per second

4.2 Evaluating Link Quality

The goal of an antenna system in a long distance point to point communications link is to maximize the signal to noise ratio at the receiver. Once everything has been done to minimize noise, this is equivalent to transferring as much power as is feasible from transmitter to receiver. For purposes of comparing antenna performance, it is useful to introduce the concept of an isotropic radiator. This conceptual antenna transmits power equally in all directions, so that at any given radius from the transmitter site, the area power density is simply the transmitted power divided by the area of a sphere of that radius. Furthermore, since receiving antennas can be characterized by an effective area, the ratio of power received at the receiving antenna to power transmitted for an isotropic transmitter is given by:

$$Q = A_{EFF} \frac{1}{4\pi r^2}$$ **Equation 4-1**

referred to here as Q or the link quality factor.[8] Although the concepts in the paper could also be applied to UAVs which need to receive data, for conceptual clarity this analysis is focused on the problem where the receiver site has an antenna of fixed and known effective area and the UAV cluster must transmit data to the receiving site.

Real antennas are not isotropic, but instead transmit more power in some directions than in others. Numerically, directivity is the ratio of the power transmitted in a given direction to the power that would be transmitted in that direction by an isotropic antenna. When directive antennas are used, the link quality factor Q is improved by a factor of the directivity.

4.3 Noise

The factor which ultimately determines the required received signal strength is the noise present in the receiver input, from sources both internal and external to the receiver. In the VHF to microwave region, there is a broad minimum in the externally-generated noise seen by the receiver (roughly from 1GHz to 10GHz).[19] This makes this range of frequencies desirable and may affect the choice of frequency for a UAV communication link. However, there are often other factors determining the frequency such as pre-existing equipment which must communicate with the UAV cluster. Because there are so many factors affecting frequency selection, we focus on selecting the frequency which makes the array itself transmit as much power to the receiver as possible.

4.4 Effect of Element Position Errors

The primary difference between a traditional antenna array and a vehicle-based array is that the former is usually constructed on a rigid structure which ensures constant, nearly perfect element positioning, while a vehicle-based array must contend with wind gusts and turbulence. The resulting element position errors, or deviations from ideal positions, have two effects: reduction of directivity in the desired direction and an increase in directivity in other directions. In other words, it tends to make the array more isotropic.

The effects of element position errors have been investigated primarily by those interested in maximizing the performance of large array antennas.[20] It was found that, in the limit of an array with an infinite number of elements, the statistical average of

the maximum directivity has an exponential dependence on the variance of the position errors. Analytically:

$$D = D_o \left(\frac{1}{1 + \frac{1}{2}\pi\left(e^{(2\pi\sigma)^2} - 1\right)} \right)$$ **Equation 4-2**

where D is the directivity with errors, D_o is the directivity without errors, and σ is the root mean square position error in wavelengths. Figure 8 shows this along with simulation results of two finite arrays with position errors. The vertical axis is directivity normalized to the ideal, error-free directivity of the array. The horizontal axis is the root-mean-square position error for each of the three spatial dimensions (assuming that the RMS error is the same in all three dimensions).

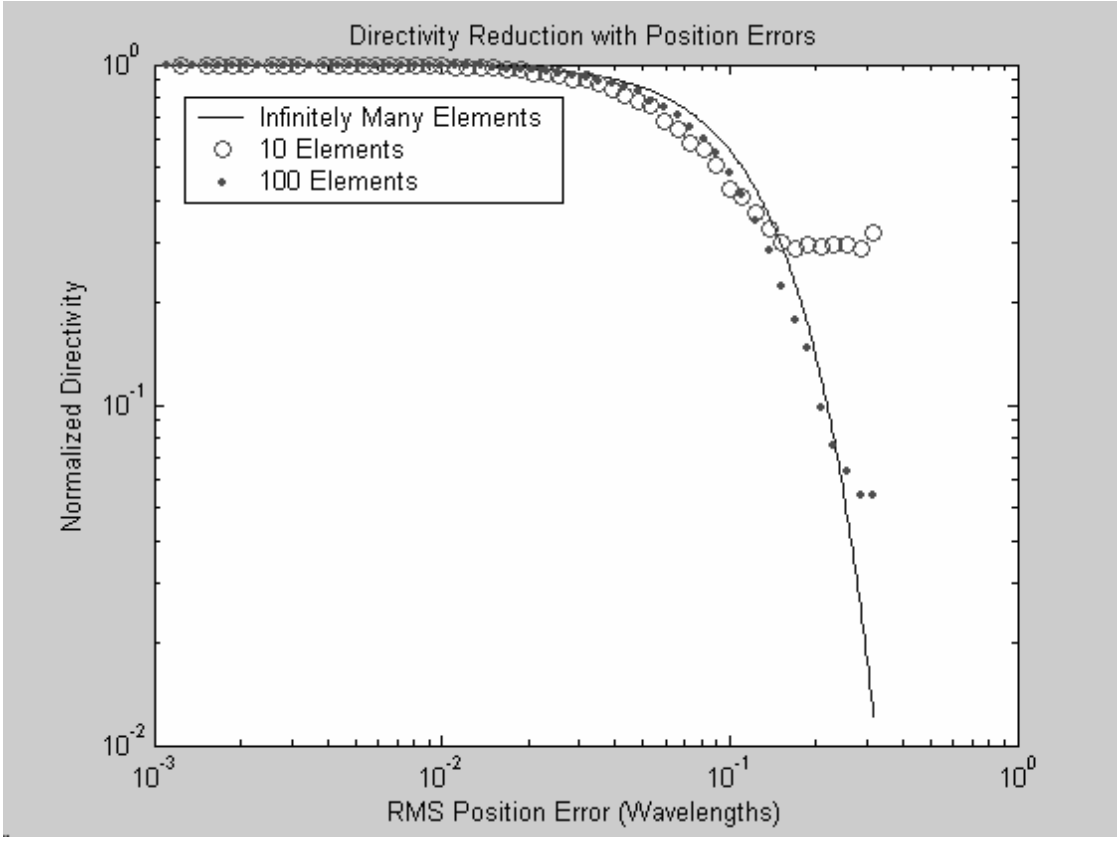


Figure 8 - Directivity Reduction with Position Errors (©2003 IEEE – see reference 2)

The two finite array cases used were both endfire arrays, one of 10 elements and the other of 100 elements. The elements are ideal half-wave dipoles fed with current sources of identical amplitude, but with the phases adjusted according to the Hansen-Woodyard criterion for an endfire array.[21] The individual elements have a directivity of about 1.67 and the error-free directivity of the two linear arrays are 23.2 and 185.7 respectively.

Because of conservation of energy, it is not possible for the maximum directivity of an array to be less than one, so it is clear that the function for a finite case cannot be identical to that of the infinite case, which continues to lower and lower values of normalized directivity. Figure 8 shows that the results for the finite case match those of the infinite case closely until a directivity of a few dB more than the directivity of an individual element is reached.

For radar applications and other situations where interference rejection is a factor, the directivity in undesired directions is important. Expressed as a ratio with the error-free antenna pattern, this is affected more strongly at small levels of position error since the ideal antenna pattern may contain deep nulls. Because this investigation focuses solely on maximizing the distance over which UAV clusters can communicate with a base station, this consideration is left for future study.

4.5 Phase Compensation

In basic phased array antenna theory, the direction of maximum radiation from the array is the direction in which the signals from each element most nearly add in

phase.[8] This naturally leads to the idea that one might be able to compensate for deviations in the positions of the array elements by changing the driving phase of the elements so that their signals add in phase at the receiver. This technique involves two steps: determining the amount by which the phase of the signals from each of the antenna elements, as received at the intended receiving site, has been shifted by position errors, and then applying a corrective reverse phase shift to each respective element of the array. Figure 9, part A illustrates this computation, which is based simply on geometric propagation path length changes and the speed of light. That is, the change in the path length, due to position error, from each element to the receiver is computed. Then, the phase of the signal driving each element is shifted by exactly the amount needed to counteract the phase shift due to these path length changes, causing the signals to all add in phase at the receiver.

This "traditional" phase correction works well for small position errors and provided that the element positions are known, it is very easy to compute the necessary corrective phase shifts. However, there are cases where one can do better than simply ensuring that the element signals still sum in-phase at the receiver. This is due to the fact that the actual power radiated by the antenna is the integral of that radiated in all directions. See Figure 9, part B. If one compares two antenna patterns, both with the same main lobe but one of which has a large secondary lobe, then the one without the secondary lobe will have a greater directivity, even though it radiates the same amount of power in the intended direction. What is being held constant between the two antennas is the magnitude of the currents in the elements. The difference in directivity is because the one with the additional lobe requires more input power to achieve the same radiated power in the desired direction.

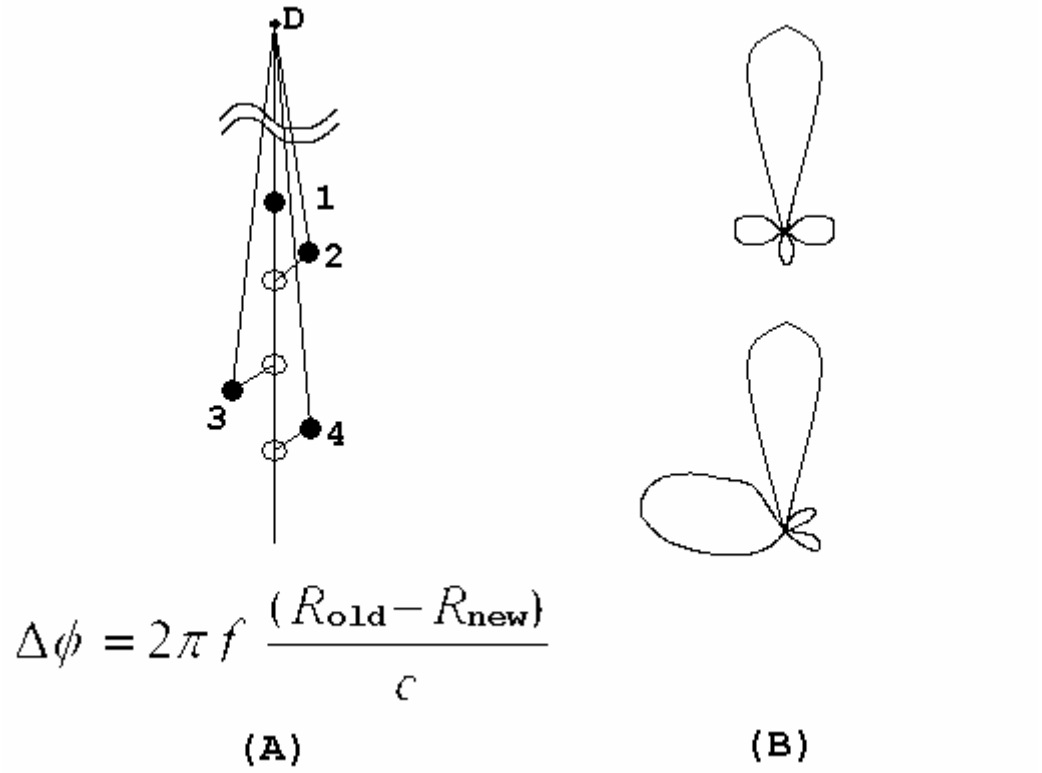


Figure 9 - Phase Correction Example (©2003 IEEE – see reference 2)

Traditional phase correction assumes that altering the phases will not change the relationship between element current and radiated power (the radiation resistance) significantly. Often this is true but not always. Table 1 compares traditional phase correction to the optimal case for a simple two-element array with a spacing of 1/8th wavelength. The optimal phase for the second element was computed by doing a local maximum search with the MATLAB command *FMINSEARCH*, which uses the Nelder-Mead simplex method.[22]

Table 1: Simple Phase Correction vs. Global Search

(©2003 IEEE – see reference 2)

Type of Phase Compensation	Phase	Directivity
Traditional	-45.0 degrees	3.14 dBi (2.06)
Local Search	-162.3 degrees	7.45 dBi (5.56)

The best method of phase compensation is dependent on the exact application, with the most important factor being how close the elements will be spaced. If the elements do not interact by receiving power from adjacent elements then the total transmitted power will not depend on the element positions and the "traditional" method will be the optimal. Determining the best phase compensation method in each circumstance could be the subject of future research.

4.6 Simple Cornell AFV Simulation in MATLAB

Because the work on the feasibility of airborne vehicle-based phased arrays occurred at the same time as the design and construction of the Cornell AFV, it was necessary to produce a simulation of its performance to obtain approximate data for phased array performance analysis. Since the testing of the real AFV was done indoors, this simulation could also give some idea of the effect of wind on the vehicle. It is also desirable that this initial simulation be based on a simple model so the effect of all the factors can be easily seen. To this end, we produced a simple simulation based on a linear model of the Cornell AFV. This was used in the ten element array example which follows. Note that some of the constants used here are not the same as those which apply to the final Cornell AFV. This is because of the fact that this simulation was developed before completion of the AFV. For a more accurate and detailed nonlinear model and simulation of the Cornell AFV, see Eryk Nice's thesis.[17]

The AFV motors are modeled as providing a steady-state angular speed in radians/second equal to the input voltage (for simplicity) with a time constant of 200 milliseconds:

$$\omega(s) = \frac{5}{s+5} V(s) \qquad \textbf{Equation 4-3}$$

First, we give the basic constants:

$g = 9.81 \text{ meters / second}$
$b_0 = 5$
$b_1 = 5$
$J_r = 6 \times 10^{-4} \text{ kilogram} \cdot \text{meters}^2$
$h = 0.2 \text{ meter}$
$d = 1 \text{ meter}$
$m_c = 3 \text{ kilograms}$
$m_r = 0.5 \text{ kilogram}$

Torque and thrust(F) are based on a simple quadratic thrust model:

$$F = K_{\ln} \omega^2 \quad \textbf{Equation 4-4}$$

$$\tau = K_{tn} \omega^2 \quad \textbf{Equation 4-5}$$

$$K_{\ln} = 8.80 \times 10^{-5} \text{ Newtons/(radian/second)}^2 \quad \textbf{Equation 4-6}$$

$$K_{tn} = 3.25 \times 10^{-6} \text{ Newton meters/(radian/second)}^2 \quad \textbf{Equation 4-7}$$

Linearized about the hover angular speed:

$$g = 9.81 \quad \textbf{Equation 4-8}$$

$$\tau = (2K_{tn}\omega_h)\omega \quad \textbf{Equation 4-9}$$

$$F = (2K_{\ln}\omega_h)\omega \quad \textbf{Equation 4-10}$$

$$\omega_h = \sqrt{\frac{g(4m_r + m_c)}{4K_{\ln}}} \quad \textbf{Equation 4-11}$$

We then define a number of constants for ease of notation:

$$M = m_c + 4m_r \quad \textbf{Equation 4-12}$$

$$K_l = 2K_{\ln}\omega_h \quad \textbf{Equation 4-13}$$

$$K_t = 2K_{tn}\omega_h \quad \textbf{Equation 4-14}$$

$$q = \frac{4m_r h}{M}$$ **Equation 4-15**

$$a_0 = q^2 m_c + m_r\left(2d^2 + 4(h-q)^2\right)$$ **Equation 4-16**

$$a_2 = 4d^2 m_r$$ **Equation 4-17**

$$c_0 = c_2 = -K_l \frac{d}{a_0}$$ **Equation 4-18**

$$c_1 = c_3 = -c_0$$ **Equation 4-19**

$$c_4 = c_6 = \frac{K_t - J_r b_1}{a_2}$$ **Equation 4-20**

$$c_5 = c_7 = -c_4$$ **Equation 4-21**

$$c_8 = g$$ **Equation 4-22**

$$c_9 = -g$$ **Equation 4-23**

$$c_{10} = \frac{K_l}{M}$$ **Equation 4-24**

$$c_{11} = c_{13} = \frac{b_0 J_r}{a_2}$$ **Equation 4-25**

$$c_{12} = c_{14} = -c_{11}$$ **Equation 4-26**

$$c_{15} = \frac{1}{M}$$ **Equation 4-27**

Finally, we have the linearized equations of motion:

$$\ddot{\psi} = c_0\omega_1 + c_1\omega_3$$ **Equation 4-28**

$$\ddot{\phi} = c_2\omega_0 + c_3\omega_2$$ **Equation 4-29**

$$\ddot{\theta} = c_4\omega_0 + c_5\omega_1 + c_6\omega_2 + c_7\omega_3 + c_{11}V_0 + c_{12}V_1 + c_{13}V_2 + c_{14}V_3$$ **Equation 4-30**

$$\ddot{x} = c_8\phi + \frac{0.2}{M}W_x$$ **Equation 4-31**

$$\ddot{y} = c_9\psi + \frac{0.2}{M}W_y$$ **Equation 4-32**

$$\ddot{z} = c_{10}(\omega_0 + \omega_1 + \omega_2 + \omega_3) + \frac{1.25}{M}W_z$$ **Equation 4-33**

$$\dot{\omega}_0 = -b_1\omega_0 + b_0V_0$$ **Equation 4-34**

$$\dot{\omega}_1 = -b_1\omega_1 + b_0V_1$$ **Equation 4-35**

$$\dot{\omega}_2 = -b_1\omega_2 + b_0V_2$$ **Equation 4-36**

$$\dot{\omega}_3 = -b_1\omega_3 + b_0V_3$$ **Equation 4-37**

We first put this into state-space form, giving full state output and having the following state and inputs:

$$x = \begin{bmatrix} \dot{\psi} \\ \dot{\phi} \\ \dot{\theta} \\ \psi \\ \phi \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ x \\ y \\ z \\ \omega_0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$ **Equation 4-38**

$$v = \begin{bmatrix} W_x \\ W_y \\ W_z \\ v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Equation 4-39

We then formed an augmented system for H_∞ synthesis, adding measurement noise to the state measurement.[23] The resulting system had the same state as the original, but 23 inputs (the original 7 plus noise for each of the 16 states), and 26 outputs (angles, positions, control efforts, and the entire state as sensed).

For the output weights in the synthesis process, we set all to zero except for θ, x, y, z, and the four control efforts. This is because the vehicle performance for antenna array directivity is determined solely by the yaw angle and the x, y, z position accuracy. Constraints are of course also necessary on the control effort lest the system attempt to apply excessively high control commands which would saturate actuators in real life. The relative weight on θ was 0.01, on x, y, and z 0.1, and 0.005 on the control efforts. The heaviest weighting is on position since that has the most direct effect on array performance. Angle should not be too heavily weighted since it is used to control position.

The inputs were weighted according to the relative disturbance amplitude. The three wind disturbances were weighted as 1, and the rest of the inputs were all weighted as 0.1 since they represent sensor noise and actuator imperfections and are of more minor amplitude than wind.

The resulting closed-loop system accepts a vector of instantaneous local wind speeds (x, y, z) in miles per hour. The effect of wind is modeled as flat-plate drag on a 10 centimeter square (rough dimensions of the central mass) in the X and Y directions. In the Z direction, wind primarily affects the vehicle through the rotors. We linearized the change in rotor thrust with free-stream velocity and used this as the model for wind effects in the Z direction. The result is the seemingly arbitrary constants of 0.2 and 1.25 in the linear acceleration equations. These also include conversion factors from miles per hour to meters per second. [24]

Because of the effect on the rotors, wind has considerably more effect in the z direction than in x or y. Since we wanted to generate worst-case performance and turbulence near the ground can result in significant vertical wind components, we modeled the local wind as three independent sources driving x, y, and z directions. Each of these sources is a Gaussian white noise generator, bandlimited to 10Hz, with an amplitude of 10 miles per hour rms. Figure 10 shows the results of this simulation. The MATLAB script to perform this basic control design, along with a simple linear model of the AFV, is available on the documentation CD included with this thesis (see X:\AFVPAA.m).

4.7 Ten Element Array Example

We now analyze how one could construct a communications link between a cluster of 10 Cornell AFVs and a line-of-sight base station 100 kilometers distant, such as a satellite. Wind disturbances and vehicle dynamics are as in the previous section. Since this model assumes that the wind gusts experienced by each vehicle are uncorrelated, this is a worst-case model for wind behavior (correlated disturbances would not cause

as much relative motion between the vehicles). Table 2 shows the specifications of the Cornell AFV at the time of the project.

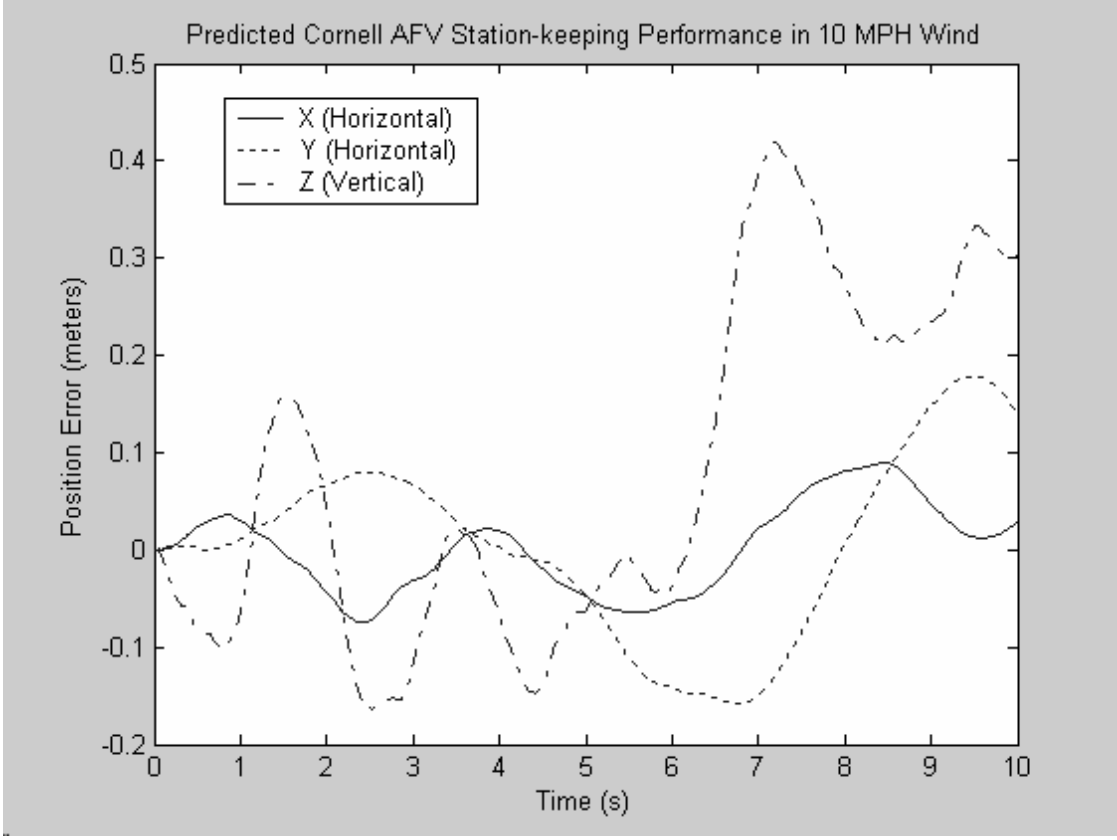


Figure 10 - Linearized AFV Model Driven by 10 mph Wind (©2003 IEEE – see reference 2)

Consider that the base station antenna is a simple half-wave dipole (perhaps because it cannot be steered and must be able to receive from all directions) operating at 150 MHz. This would have an effective area of 0.52 meter^2 . Assume that the same kind of antenna is used on each AFV and that the receiver must receive -70dBm to ensure a reliable communication link. At 100km distance, this means that 14 Watts are required if each AFV transmits by itself.

Table 2: Cornell AFV Specifications (©2003 IEEE – see reference 2)

Parameter	**Specification**
Dimensions	1.5 x 1.5 x 0.4 meters
Mass	6 kilograms
Linear acceleration	1 g maximum
Angular acceleration	2 radians per second^2
Rate gyro noise	0.035 degree per root hour
Rate gyro drift	3 degrees per hour
Motor time constant	40 milliseconds

The controller we used (from the previous section) does not take into account the presence of the other AFVs and is therefore a totally decentralized controller, providing a worst-case formation performance bound. In future research, existing methods for distributed control design could be used to develop a more effective controller.[25] [26]

Figure 10 shows the results of closed-loop simulation of a single AFV with wind disturbances. Since the wavelength is 2 meters at the operating frequency, this represents worst-case position errors of about 0.05, 0.09, and 0.2 wavelengths in the X, Y, and Z dimensions, respectively. The Z dimension disturbance is greater because

of the larger effect that wind in the vertical direction has on the rotors. While the horizontal wind would likely be greater in magnitude in reality, there are cases, especially when operating close to the ground, when turbulence and ground effect can produce significant vertical wind components.

A linear endfire array of 10 dipole elements was designed which provides 12.1 dB gain over a single dipole, with no position errors. Figure 11 shows how the directivity of the array would change if it were vehicle-based using the formation of 10 Cornell AFVs. This is shown without phase compensation, and also with local direct search-based phase compensation. Both axes are linear.

Using the worst case from the phase compensated array, the antenna shows a directivity of 15.3, which is 9.6 dB gain over a single dipole, allowing the AFV cluster to communicate reliably with the base station 100 km away with only 1.5 Watts.

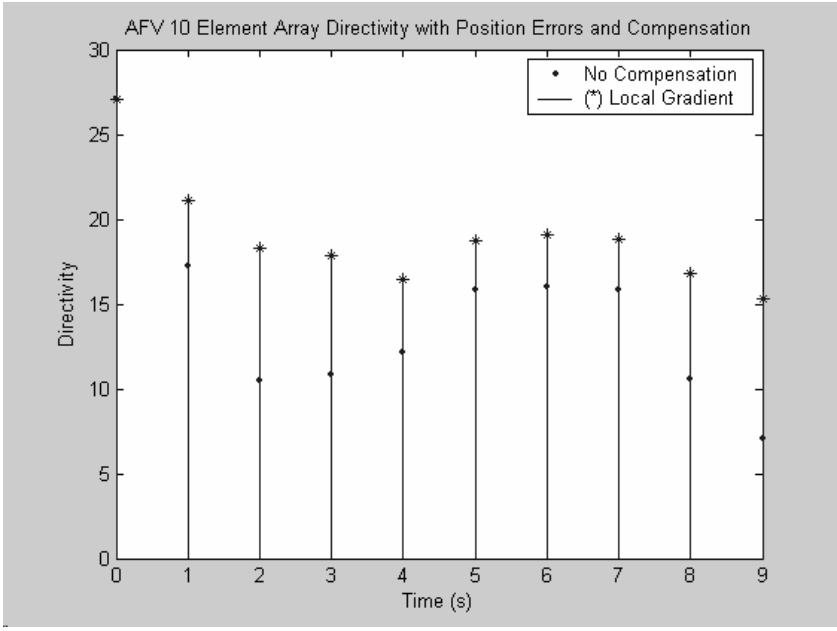


Figure 11 - Effect of 10mph Wind on AFV Array Directivity (©2003 IEEE – see reference 2)

CHAPTER 5

SPECIAL CONSIDERATIONS

There are at least two topics which were not directly dealt with in this study of airborne multi-vehicle phased arrays but which would be important in an actual implementation: phase synchronization and antenna impedance matching.

5.1 Phase Synchronization

Normally, all the elements in a phased array antenna are fed from a single transmitter or from an array of transmitters which share a master oscillator. Maintaining the proper phase relationships among the elements is no more difficult than making sure that the cable lengths to each element are accounted for.

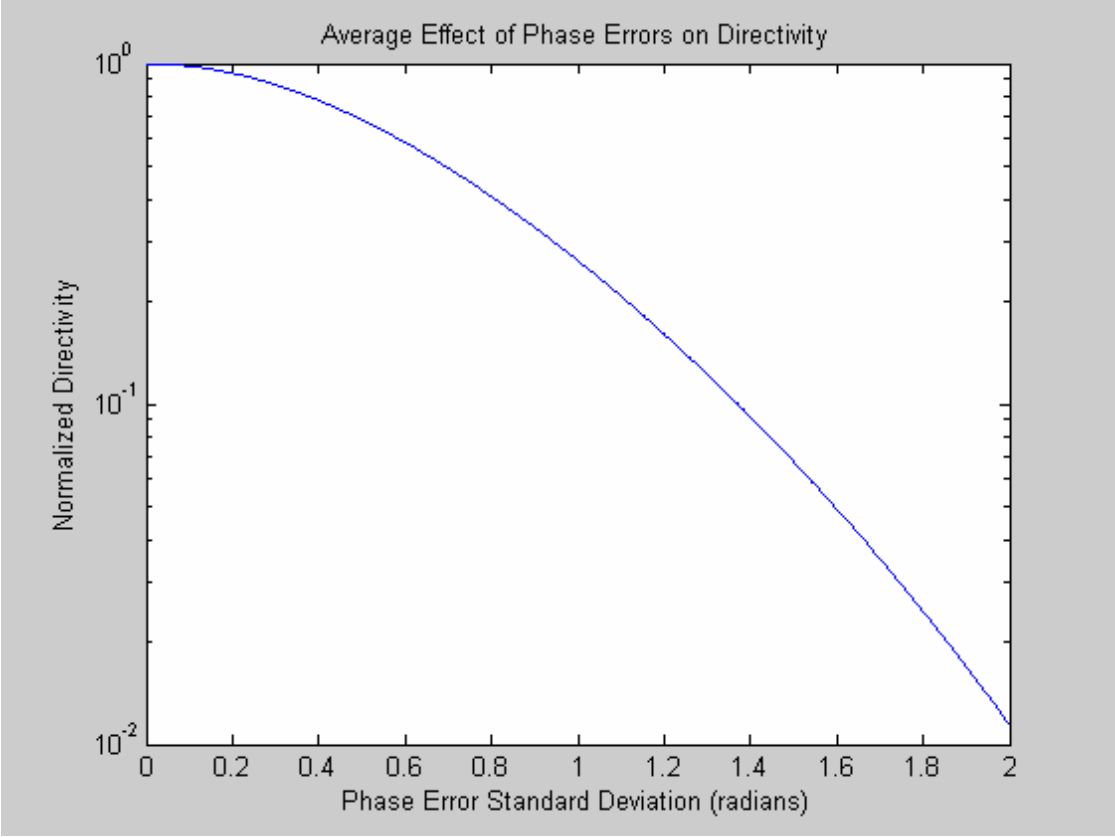


Figure 12 - Effect of Phase Error on Directivity

In a multi-vehicle array, however, it is not possible for all elements to use one oscillator since there is no physical connection among them. In addition, most schemes for synchronizing multiple oscillators would fail due to the fact that the time delay of signal transit between elements varies with their positions.

The effect of phase error in the elements is less dramatic than uncompensated position error as Figure 12 shows. Still, it is necessary to ensure at least rough (+/- 5 degree) phase synchronization to optimize array performance.

There are several possible methods of achieving phase synchronization in the multi-vehicle case. It might be possible to apply the methods of the NTP internet time synchronization protocol over a separate RF network among the elements.[27] Similarly, with a separate network among the elements, one could use the element position information to account for the time-of-flight delay between elements in the network and synchronize each clock to a master clock aboard a particular UAV.

These methods have the disadvantage of requiring a separate communication link among the elements (although this might be necessary for other supervisory information anyway) and are also susceptible to single-point-failure if the "time master" UAV is damaged.

An alternative which seems to be the best is to use the GPS time standard to synchronize the clocks aboard all UAVs. In this case, they need not communicate time information among each other, but simply use the time output of GPS receivers (which they might already be using for navigation and position sensing for phase correction).

More work should be done to determine how accurate this form of synchronization could be.

5.2 Antenna Impedance Matching

Most of the analysis in this paper has assumed that the antenna elements would be driven by current sources. There are several practical problems with this: first, an accurate, efficient current source is difficult to make at radio frequencies. Most transmitters are designed to be connected to an antenna with a well-controlled impedance, yet the impedance of our elements would vary as their relative positions varied due to the cross-coupling among them. Secondly, if each element were fed with a current source, some elements would consume much more power than others due to the impedance differences. This would cause some UAVs to have a longer battery life than others.

Probably the best way to deal with this problem is to use automatic antenna tuners on each element to match the transmitter's expected impedance to the antenna's actual impedance. Because this introduces phase shifts, it would also be necessary to measure this phase shift (could simply be a lookup table based on the present antenna tuner settings) and apply the opposite shift to the transmitter signal. This would still leave current amplitude differences among elements since they would each be radiating the same power rather than the same electric field strength. This could be partly accounted for by using array design techniques which assume similar element output powers, except for those cases where position errors would require differing element powers in order to maintain the same overall antenna pattern. This is another area where more work would have to be done. If it turns out the be a major problem, it would always be possible to fall back to using actual current sources and to shift

UAVs around in the formation if necessary to balance their battery life if transmitter current drain were a significant percentage of the battery drain. However, it would be desirable to avoid all of this if possible.

CHAPTER 6

MACRO-ANALYSIS OF MULTI-VEHICLE PHASED ARRAY PERFORMANCE

6.1 Definition of Terms

A_{EFFAFV} – effective area of AFV antenna array in meters^2

$A_{EFFBASE}$ – effective area of the base station antenna in meters^2

d – maximum linear dimension of AFV in meters

D – directivity of an antenna

D_o – directivity of an antenna array before introducing element position errors

G_{AFV} – gain of AFV antenna array

G_{BASE} – gain of the base station antenna

N_{TH} – for a given amount of element position error, the minimum number of elements for which the array shows appreciably more gain than a single element.

Q – link quality, the ratio of the power transmitted to the power received in a transmitter-receiver antenna pair, evaluated at a separation of $\frac{1}{2\sqrt{\pi}}$ meters.

λ – wavelength in meters

σ – standard deviation of the 3D Euclidean norm of the element position error in wavelengths.

6.2 Assumptions and Restrictions

To make the problem of analyzing the performance of an array of AFV-supported antennas more tractable, we must place some restrictions on the configuration of the system. We will assume that all the AFVs have completely efficient, identical antennas. We will also assume that all of these AFVs possess all of the data to be transferred (in the transmitting case) or are capable of receiving and using all of the

data to be transferred (in the receiving case). We select the key AFV limitation to be size (in other words, we assume that we can use any type of antenna of maximum dimension d on an AFV with maximum dimension d). Finally, we will restrict ourselves to the case where the AFVs must communicate, via line of sight propagation, with a single base station, which may be either a satellite or a terrestrial installation.

Information theory tells us that the data rate capacity of a communication channel is linearly related to the signal to noise ratio (SNR).[28] To evaluate the ultimate limit on the performance of any communications system, therefore, we need only consider the SNR, so long as the noise being experienced is Gaussian in nature, which is a reasonable assumption (because it is dominated by thermal noise) unless the system is experiencing jamming or other interference. This more extensive case could be studied at a later time.

6.3 Link Quality

Because the noise experienced by the system will be roughly the same regardless of the type of antenna system, the SNR will be linearly proportional to the received power for a given transmitted power. In addition, by antenna reciprocity, the ratio of received to transmitted power will be the same in both directions. Therefore, we can define a figure of merit for the communications system, called the link quality (Q), which is simply this power ratio evaluated at a distance of $\frac{1}{2\sqrt{\pi}}$ meters and is equal to

$$Q = \frac{1}{4\pi r^2} G_{AFV} A_{EFFBASE} = G_{AFV} A_{EFFBASE} \qquad \textbf{Equation 6-1}$$

which is also equal to

$$Q = A_{EFFAFV} G_{BASE}$$ **Equation 6-2**

where G_{AFV} is the maximum gain of the antenna array (composed of the AFVs), $A_{EFFBASE}$ is the maximum effective area of the base station antenna, and A_{EFFAFV} and G_{BASE} are defined similarly. The term gain is used instead of directivity simply because of convention. In this case they can be used interchangeably because the system is presumed to be completely efficient. By evaluating the ratio of received to transmitted power at a distance of $\frac{1}{2\sqrt{\pi}}$ meters, the constants involving distance in front of the expression drop out and we are left with the simplest expression for the link quality. We are concerned with the effect that antenna type and frequency have on the link, not the effect which distance has, which will be the same independent of the antennas and other equipment.

6.4 Frequency Limitations

For a communications system such as we are considering, there are upper and lower frequency limitations. The upper limit is imposed by attenuation in the atmosphere due initially to scattering by rain, fog, and clouds, and at even higher frequencies, due to absorption by oxygen molecules. This limit varies with conditions but a good general figure is about 30GHz.

A lower limit is enforced by the need for reasonably efficient antennas. As mentioned in the review of antenna theory, antennas begin to become inefficient when their maximum dimension is less than ¼ wavelength (such as a whip antenna over a ground

plane). Therefore, given an AFV size limitation, d, we will impose a lower frequency limit of

$$\frac{75}{d} MHz$$ **Equation 6-3**

where d is in meters, which corresponds to the frequency where d is ¼ wavelength.

6.5 Linear vs. Aperture Antennas

Array antennas can be comprised of any kind of elemental antennas. If aperture antennas are used, they possess a maximum effective area which is approximately equal to the physical area of the aperture (for a diameter much larger than one wavelength). The maximum effective area of the array is then N times that of the individual elements (assuming they are all identical). The gain can be computed using the gain/effective area relationship given in the antenna theory review (see APPENDIX A, inverse of Equation 7-1):

$$G_{MAX} = \frac{4\pi}{\lambda^2} A_{EFFMAX}$$ **Equation 6-4**

If linear antennas (such as dipoles) are used as the array elements, then each element will have a small, fixed maximum gain (about 1.6, which can be roughly assumed to be 1 for this investigation) and its effective area will vary with frequency and can be computed using the inverse of the same formula:

$$A_{EFFMAX} = \frac{\lambda^2}{4\pi} G_{MAX}$$ **Equation 6-5**

6.6 Test Cases

We are now ready to set up some specific phased array configurations as test cases and compare their link quality. The principal remaining free variable is frequency. For each end of the link (base station or AFVs), we can either select an antenna type which maintains a constant gain over frequency (such as a half-wave dipole or an array of such antennas – here we assume that the dipole length is adjusted for the selected frequency) or a constant effective area over frequency (such as a dish antenna or other aperture antenna or an array of these). We will look at four test cases, each of which selects one of the two options for each end of the link.

6.6.1 Constant G_{BASE} and G_{AFV}

There is no particular reason why one would need to choose a constant gain antenna for the AFV side (i.e., an array of linear antennas), but there are reasons why the cluster of AFVs might need to communicate with a base station which has a fixed gain. The most prominent example would be a typical communication satellite. Communication satellites usually need to cover a large geographic area. Their antenna pattern must be such that their radio "footprint" covers this entire area. This piece of information alone determines the gain, unless a steerable antenna (either physically steerable or electronically steerable) is employed. There are currently many satellites which operate in a constant gain mode, and it is likely this will still be the situation for some time in the future.

For this case, the link quality Q, is

$$Q = \frac{G_{AFV} G_{BASE} \lambda^2}{4\pi} \qquad \textbf{Equation 6-6}$$

In addition, we know that the frequency is limited by the size of the AFV. Since maximum link quality in this case is achieved at the lowest frequency, we can say that the maximum Q for this configuration is about

$$Q = \frac{G_{BASE} N 4 d^2}{\pi}$$ **Equation 6-7**

6.6.2 Constant G_{BASE} and A_{EFFUAV}

This case is where we have the same base station situation as before, but now we are using aperture antennas on the AFVs. This might be motivated by trying to improve the link by achieving the maximum gain possible for the AFV array (by using aperture antennas and a very high frequency). The link quality is now

$$Q = G_{BASE} A_{EFFAFV}$$ **Equation 6-8**

Consider that the maximum effective area we can achieve on each AFV is about d^2 and that the total effective area of the array will then be N times this, and we get

$$Q = G_{BASE} N d^2$$ **Equation 6-9**

which is the same as before except for a factor of $\frac{4}{\pi}$, which is very close to one. Note, however, that this time the Q is actually independent of frequency, whereas we had to use the minimum frequency to get the best Q in the previous case.

6.6.3 Constant $A_{EFFBASE}$ and G_{AFV}

This corresponds to the case where the base station antenna is simply limited by its maximum size, and we choose linear antennas for the AFVs. Now,

$$Q = G_{AFV} A_{EFFBASE}$$ **Equation 6-10**

Since the linear antennas on the AFV side have a gain close to 1, we can rewrite this as

$$Q = NA_{EFFBASE}$$ **Equation 6-11**

Just like the previous example, this is frequency independent, and is the only case to be independent of AFV size.

6.6.4 Constant $A_{EFFBASE}$ and A_{EFFAFV}

The final case is where both sides use aperture antennas and are size limited. This time,

$$Q = \frac{A_{EFFAFV} A_{EFFBASE} 4\pi}{\lambda^2}$$ **Equation 6-12**

We know that the AFV array effective area is approximately N times d^2 so:

$$Q = 4\pi A_{EFFBASE} N\left(\frac{d}{\lambda}\right)^2$$ **Equation 6-13**

which is both frequency and AFV size dependent.

6.6.5 Comparison

In the situation where the base station has fixed gain, Q is essentially the same regardless of whether the AFVs have fixed gain or fixed effective area over frequency. Notice that the link quality has a linear dependence on N and a quadratic dependence on AFV size. So, if we compare, for example, AFVs which are 5 times larger than smaller ones, we would need 25 times more smaller ones to obtain the same link quality as the fewer larger ones.

When the base station is limited only by size (so it has a constant effective area), it is very likely that we can use a high enough frequency that d will be larger than λ, (since the wavelength is about 1 centimeter at 30GHz), so it is fairly clear that we would want to use aperture antennas on the AFVs, if all else is equal.

In the next section, we show how errors in AFV position control and position knowledge affect the gain of the array. Then, we combine all the factors affecting array performance and compare this performance with that of simple alternatives.

6.7 Effect of Position Errors on D_{MAX}

Looking only at the results obtained so far, it would seem that there is a definite benefit to using arrays of AFVs over single AFVs in a communication link. What has not yet been considered are the relative costs involved in each of the ways of improving the link quality, which would strongly affect one's choice of how much to rely on each method. Most of these costs will be discussed in the next section, but the principal factor which determines the difficulty involved in forming an array, the effect of random relative position error of the AFVs on the maximum gain or directivity, is presented in this section.

6.7.1 Uncorrected Effect

Two things happen when random position error is introduced into an array: the direction of the maximum directivity is changed and the value of the maximum directivity is also altered. When we use the term "uncorrected effect", we mean simply the effect of altering the value of D_{MAX}, not its direction. We assume that the array has

been steered (either physically or electronically, by changing the phase of the signal driving each element) to point the maximum at the base station.

From the presentation in Wang[20], the uncorrected effect is given by

$$D = \frac{D_o}{\left(1 + \frac{\pi}{2}\left(e^{\left(\left(\frac{2\pi\sigma}{\sqrt{3}}\right)^2\right)} - 1\right)\right)}$$

Equation 6-14

where D_o is the maximum directivity without errors, and σ is the standard deviation of the Euclidean norm in 3D of the position error as a fraction of a wavelength. Note that this is the same as Equation 4-2 except that there, σ is the position error as a fraction of a wavelength in each axis. Both equations assume that the position error is not biased toward one axis versus the other two because then the antenna orientation relative to the disturbances would matter. In other words, Equation 6-14 assumes that the error in each of the three dimensions is independent, identically distributed. A plot of this function is shown in Figure 13.

6.7.2 Correction Methods

It is obvious that better control of AFV position can reduce position errors. However, there may well be other methods of compensating position error. A survey of IEEE papers and conference proceedings turned up only a very small body of literature on methods that clearly accomplish such compensation for vehicle-based arrays (Jan and Enge [29] discussed something similar for land vehicles, and Boyd and Lorenz [30] discuss "minimum variance beamforming" which minimizes total radiated power while maintaining at least unity gain in a desired direction with array position uncertainties).

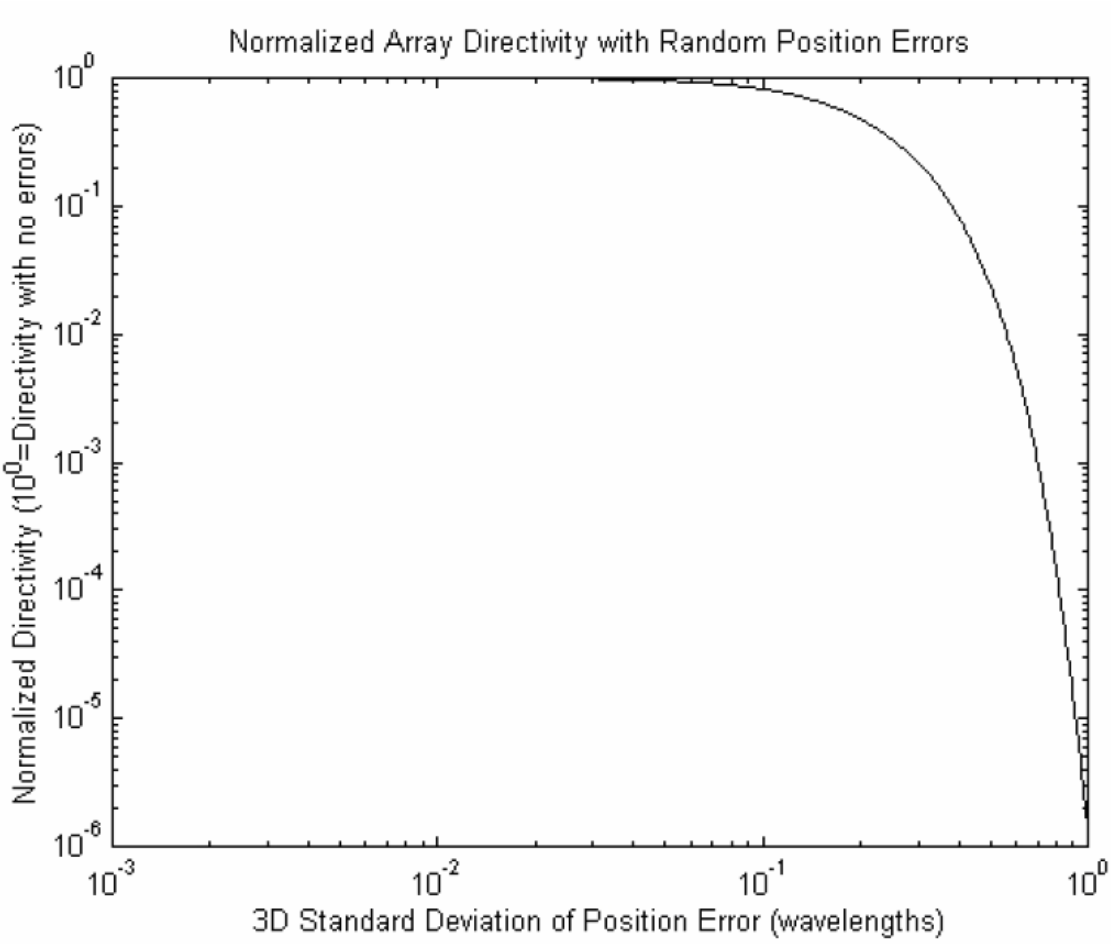


Figure 13 - Theoretical Effect of Random Position Error on Directivity

Simulation of small arrays with position errors indicates that a combination of phase and amplitude compensation can almost totally compensate for small position errors (on the order of a tenth of a wavelength) in many cases.

Phase compensation can be accomplished by the obvious means of synchronizing the carrier phase of the AFVs and applying specified phase offsets. Amplitude compensation would require AFVs to take turns in different places in the array to

allow them to share average power evenly, otherwise some AFVs would be required to transmit significantly more average power than others.

Two practical methods of phase compensation (simply making the phases add at the receiver and a search of the phase space to take element interaction into account) were discussed in CHAPTER 4. For a more in-depth discussion, see the paper by Ramu Chandra, Raffaello D'Andrea, and myself in [3].

6.8 Overall Effect of Errors

The overall effect of these errors is to increase the number of array elements that would be needed for there to be any benefit from forming an array. So, for example, for uncorrected errors around 0.2 wavelength, an array larger than two elements would be needed to have any benefit (since the gain reduction at 0.2 wavelength error is about 50%, a two element array would have roughly the same gain as a single element), so 0.2 wavelength is a tolerable error. However, at 0.3 wavelength, an array bigger than 10 elements would be needed to show a benefit. This quickly increases due to the exponential dependence on error. We can see, therefore, that the practical limit of uncorrected errors would be somewhere around 0.2 to 0.3 wavelength.

6.9 Comparison – Phased Arrays vs. Other UAV Communication Methods

For this comparison, we will assume that the communications link is already operating with the maximum power available and that noise has been reduced to a minimum since these factors affect all link quality improvement methods equally.

The approach we are taking in this comparison is that we have a set cost budget and want to get as much link improvement as possible within that budget. We will attempt to evaluate, in qualitative terms, the cost of each of the available options for link quality improvement. We will also look at whether the options' cost per improvement ratios are independent or interact.

6.9.1 UAV Array Option

It is always true that using an array improves the link. However, the amount of improvement is negligible below a certain threshold $N=N_{TH}$, which is determined by position error and frequency. Above this threshold, the improvement is linear and is roughly equal to N/N_{TH}. The cost of this option is probably roughly proportional to N and more strongly dependent on the relationship between position error, frequency, and N_{TH}. That is, for a given position error and frequency, it may be possible to reduce the threshold, but the better position sensing which would be necessary would cost a great deal. Note that the cost per improvement of this option interacts with any option which alters frequency.

Also, above an error standard deviation of 0.1 wavelengths, the threshold is exponentially dependent on error, so there is a maximum error (around 0.3 or 0.4 wavelengths) above which N would have to be prohibitively large to cause any improvement.

6.9.2 Larger UAV Option

Regardless of what type of base station is used, increasing AFV size helps quadratically. One can consider increasing the size of all of the AFVs in an array, or

dropping the idea of an array and just having one large AFV act as a communications relay for the others. The cost of this option varies widely depending on the application. Military applications would certainly have definite size limitations if the chances of the AFVs being detected by sight or radar had to be minimized. Many civilian applications, though, such as ecological surveys, would not have small size limitations.

6.9.3 Option to Change Base Station Antenna

As discussed in the section on phased array test cases, satellites and other base stations which need to serve a large number of users may have a maximum gain limitation. This results in a link quality of

$$Q = G_{BASE} N d^2$$ **See Equation 6-9**

If it is the case that the gain must remain fixed, then there is no way to improve the link by altering the base station antenna.

However, some experiments have been conducted to allow satellites to have electronically steerable antennas. Satellites which do not need to serve many users or satellites with advanced antenna types may be able to relieve all antenna limitations except size, corresponding to a constant $A_{EFFBASE}$ case. It may or may not be possible to use such satellites. The AFVs may need to communicate with a certain preexisting traditional type of satellite, or it may be that it costs too much to launch a satellite with this newer antenna type.

If it is possible to switch to a constant $A_{EFFBASE}$ antenna at the base station (or if the base station already has such an antenna), then two options for link improvement become available. One option is just to increase the size of the base station antenna. This causes a quadratic improvement in link quality for a linear increase in maximum size (or a linear improvement for a linear area increase).

Another option is to increase operating frequency. In the case of a base station with constant effective area, increasing operating frequency (up to the limit imposed by the atmosphere of around 30GHz) would provide a quadratic improvement in link quality. This interacts with the AFV array option because it raises the threshold N for the array. Increasing frequency provides a lot of improvement for very little cost in most cases. Note that changes in naturally-occurring noise must also be taken into account in selection of frequency (see section "4.3 Noise").

6.9.4 Option Selection

It seems fairly clear that in the case where the base station must have constant gain and the AFVs have a definite size restriction, the array is the only option for improvement. Using the lowest frequency where the AFVs can carry efficient antennas and the largest N possible would result in the best improvement.

If the base station must have a constant gain but the AFVs can be made larger with low cost, one would definitely make them as large as possible before resorting to an array because of the higher cost of an array and the fact that size results in a quadratic improvement in Q.

If the base station can have an antenna with constant effective area, then we would still want to make the AFVs as large as possible before resorting to an array. We are then left with the competition between the two interacting options, increasing frequency and the array. The exact result here would depend on the actual application and cannot be resolved without dealing with actual numbers on the cost of good position sensing and how it affects the threshold N.

In this last case, if increasing frequency to near 30GHz is indeed inexpensive and provides adequate link quality improvement, then it would seem as though it would be best to do so and then only use an array if excellent position sensing is available to make the threshold N viably small. If this increase in frequency is not enough to provide the desired link quality, then it is likely that decreasing frequency to the point where the threshold N is much lower and using a large number of vehicles in an array would be the best solution.

CHAPTER 7

SUGGESTED FUTURE WORK

There were three principal flaws with the AFV design as I left it: navigation drift, limited battery current handling ability, and the expense and fragility of the DQI-105 IMU. By far the greatest of these was navigation drift. Because the only velocity sensing was done through the integration of acceleration data from the inertial sensors, which were subject to both thermal offset changes and inexact gravity vector subtraction, the AFV would not on its own hover in place but rather slowly drift away from the proper position. Besides this, all flight testing was done with external lead acid batteries via a tether, although the LiPoly battery packs were mounted to the vehicle to simulate the proper weight. Running with the actual onboard batteries would be difficult because of the 10 minute endurance limit and also because our required current drain was really pushing the cells right up to their limits. They probably would not have a very long service life.

A good deal has already been done to address these problems by Dr. JinWoo Lee, Oliver Purwin, and Eryk Nice. They selected an indoor GPS-like system[31] which allowed absolute position updates to be fed to the control system, eliminating drift. This allowed the use of an IMU which was only half the cost and mass of the DQI-105. Finally, Dr. Jinwoo Lee designed and constructed a new main controller PCB. The present vehicle uses the same airframe, motors, motor control units, and propellers as the old system, but has little else in common with the older system.

The new system runs with newer LiPoly cells with a higher current-handling capacity. Still, Oliver Purwin reported excessive heat and inefficiency problems with the motor controller units. I suspect that the components that guard against FET cross-conduction (from the positive supply rail to ground) may not be installed correctly or may have been damaged. This should be corrected. Ideally, a new motor controller PCB should be designed to get rid of the complex modifications which are needed for each present motor control unit.

The AFV could benefit from better RF communications such as 802.11 (wireless Ethernet) or 802.15.4 (ZigBee). This would take a good deal of the communications burden away from the onboard microcontrollers and allow more reliable, expandable communications. The communication system I designed had in mind minimum latency for the future when multiple vehicles would have to communicate for formation flight. 802.11 and 802.15.4 may pose latency problems which would have to be checked before using these technologies.

Of course, the final improvement to the system would be actual distributed control of several vehicles in formation flight. This goal does not seem very far off since one vehicle has demonstrated reliable, stable, drift-free hovering flight.

APPENDIX A

ANTENNA THEORY OVERVIEW

A.1 Definition of Terms

A_{EFF} – effective area of an antenna in meters^2

d - distance between two elements in an example array in wavelengths

D – directivity of an antenna

D_{MAX} – maximum of the directivity function over all angles

I – rms magnitude of the current at the feedpoint of an individual antenna element in Amps

N – number of elements in an array

R – real part of antenna feedpoint impedance in Ohms

w – width of an antenna array (distance between two farthest-spaced elements) in meters

Θ – elevation angle in polar coordinates

Φ – azimuth angle in polar coordinates

λ – wavelength in meters

A.2 What is an Antenna?

At a fundamental level, an antenna is a device for transferring electromagnetic energy between a guided propagation medium (such as a coaxial transmission line) and an unguided medium (such as free space). Whether the EM wave is propagating from the transmitter to the antenna, or in the other direction, from free space to the antenna, current is induced in the antenna. In the receiving case, the current in various parts of the antenna sums in such a way at the feedpoint (where the transmission line attaches) that an impedance match is accomplished between free space and the transmission

line. Likewise, in the transmitting case, the current distribution of the antenna creates the electric and magnetic fields of a propagating wave which is not only efficiently matched to the transmission line but also may propagate in some directions more than in others.

As is suggested by the analogy between an antenna and an impedance matching device, antennas are reciprocal devices. Roughly speaking, this means that they function equally well in the receiving mode as in the transmitting mode. In fact, since both receiving and transmitting involve bidirectional interactions between the E and H fields and the antenna itself, it may be said that transmitting involves reception and vice versa. A more precise way of defining reciprocity is the following: if two antennas are set up in free space and we treat this two antenna system as a two-port "black box" where the ports are the feedpoints of both antennas, then the gain from port 1 to port 2 (S12) will be equal to that from port 2 to port 1 (S21).

A.3 An Antenna's Two Qualities: Efficiency and Radiation Pattern

Although there are as many different antenna designs as one can imagine, an antenna's suitability for a particular application can be described by just two pieces of information: its efficiency and its radiation pattern (or just “pattern”). If two antennas share the same efficiency and pattern, they can be considered to be equivalent, except for secondary considerations such as size or weight.

It is fairly obvious what is meant by efficiency: the ratio of the total emitted (or collected) power to the input (or output) power via the feedpoint. The term pattern

may be somewhat more confusing. In a guided medium such as a transmission line, there is no concept of direction of propagation, other than a forward and a reverse direction. However, since an antenna interacts with free space which is three dimensional, it is conceivable (and indeed inevitable in reality) that an antenna will prefer some directions over others (and note that by reciprocity, it would have to prefer the same directions by the same amount for both reception and transmission).

This directional preference is usually described by a function of two variables, $D(\Theta,\Phi)$, known as the antenna directivity or, especially when represented graphically, its radiation pattern. The arguments Θ and Φ are simply the usual spherical angular coordinates with the antenna considered to be at the origin. If we consider the transmitting case, D represents the power per steradian (solid angle) seen at a point at some distance from the antenna along the radius specified by the angle arguments. D is usually normalized by the total transmitted power over 4π, which means that D would be the constant function 1 if we were considering an isotropic antenna (one which radiates equally in all directions). We can omit the radius length from the arguments of D because once the radius becomes greater than several wavelengths, the pattern becomes independent of the radius (a region known as the far-field).

By reciprocity, we know that no separate function is needed to specify the receiving pattern. In fact, if one were to set up an antenna at some distance from a transmitter site, the power which would be received by the antenna would be equal to the amount which would be received by an isotropic antenna times $D(\Theta,\Phi)$, with D evaluated at the angles which specify the radius between the two antennas. If we know the field strength at the receiving antenna, due to the transmitter, as power per area, we can

determine the received power by multiplying by something with units of area, which is known as the effective area or A_{EFF}. This is also a function of Θ,Φ and we can compute it directly from D by using reciprocity and the knowledge of the effective area of an isotropic antenna (which we can obtain by computing the effective area of any particular antenna at any particular angle arguments and dividing by the directivity at that same pair of angles). This gives:

$$A_{EFF}(\Theta,\Phi) = \frac{\lambda^2}{4\pi} D(\Theta,\Phi)$$ **Equation 7-1**

Note that it depends on the free space wavelength.

The term gain is very closely related to directivity. Gain is simply directivity times efficiency, so that it takes both into account at once. The peak directivity over all angles is known as D_{MAX} and the peak gain is G_{MAX}. Informally, the peak gain is often referred to simply as "the gain" of the antenna. This may seem very sloppy until one considers that antennas are usually pointed to place the peak gain in the direction of the other end of the link. Similarly the maximum effective area is A_{EFFMAX} or informally, just A_{EFF}.

While almost any arbitrary pattern is possible, antennas are often designed so that power is transmitted (or received) primarily in one narrow range of angles, known as a "beam". There will also always be smaller, unintentional beams known as sidelobes. By choosing a particular amount of gain reduction below the maximum (such as 3dB below), and marking off the points which have this amount of gain on a plot of G, the "size" of the principal beam can be quantified. For beams with radial symmetry (i.e.,

they essentially form a cone), the size can be described by the angular distance between the -3dB points, called the half-power beamwidth.

There is often a misconception that one can make an antenna "better" by making it larger. This is often true if the goal is greater efficiency or higher directivity. However, once a particular efficiency and directivity are chosen, it is not possible to achieve a stronger received or transmitted signal, so long as the antenna is a passive system. In other words, for both transmitting and receiving, having a stronger signal in a particular direction implies a weaker signal in all other directions. It is possible to "cheat" somewhat in reception by sampling the received E and H fields at a number of locations and applying DSP techniques to generate multiple outputs which correspond to as many beams as are desired (without reducing the gain of any of the beams), but since this would have to be an active system, it is not a counterexample to the previous statement.

A.4 Factors Determining Efficiency

It is not possible here to cover all of the factors affecting an antenna's efficiency, since that is an active area of research. Instead, we will provide a quick overview of the main consideration: antenna size.

Perhaps the most basic antenna is the center-fed half-wave dipole. It is simply a straight piece of thin wire, one half wavelength long at a particular frequency (called the resonant frequency), with a current source in the middle. The power radiated by such an antenna is

$72 I_{rms}^2$ Watts **Equation 7-2**

where the current refers to the output of the current source, a sinusoid at the antenna resonant frequency. Because the constant value has units of Ohms (72 Ohms in this case), it is known as the antenna's radiation resistance. If the wire has alternating current resistance R at this frequency, then the antenna's efficiency can be estimated by:

$$\frac{72}{72+R}$$ **Equation 7-3**

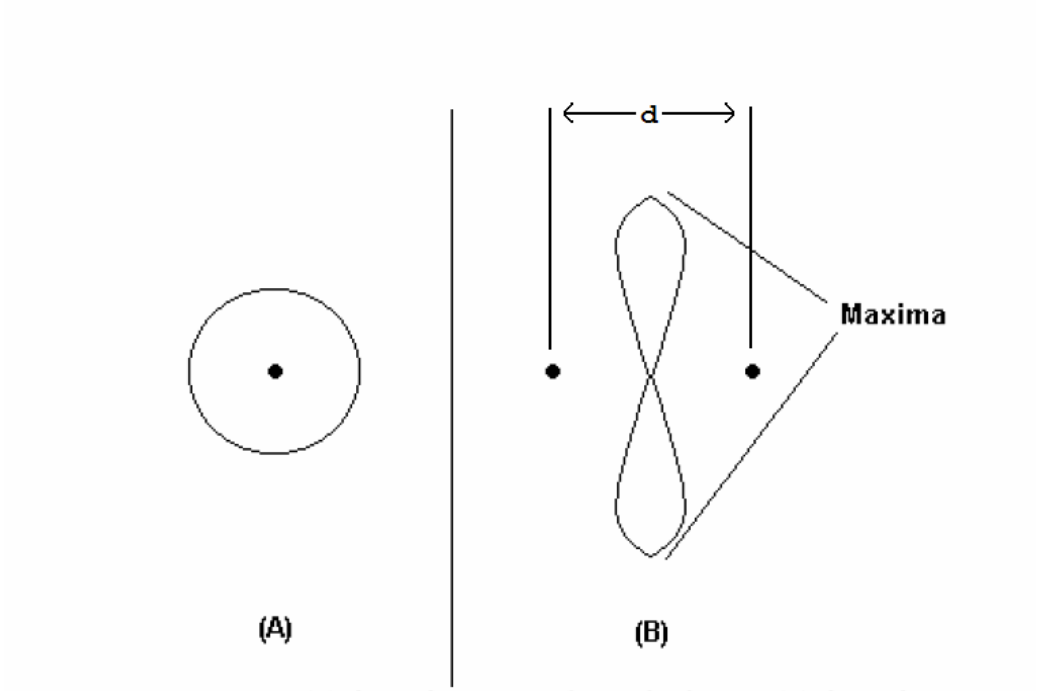
This is only an estimate since the current is not constant over the length of the antenna, which means that we are overestimating the losses by assuming that they are equal to R times the current squared. The extra power, which is not radiated, is dissipated as heat by the wire.

One way to make this antenna smaller, which is obviously often desirable, is to use the method of images to find some surface which can act as a ground plane to effectively turn a quarter wave long piece of wire into a dipole. However, unless very special materials (such as superconductors) are used, any attempt to make the overall size much less than a quarter wavelength will result in a significant reduction of efficiency.

This happens for two reasons. First of all, the radiation resistance of smaller antennas is less. Roughly, this can be explained by considering that the origin of radiation resistance is the time (or phase) delay which exists in interactions between parts of the antenna. The second reason is that smaller antennas generally provide a poorer impedance match to typical transmission lines or circuits. Antennas smaller than a quarter wave exhibit an impedance (at the feedpoint) which is not purely resistive. For

maximum power transfer, the reactive part must be canceled by an external component. The required reactance is usually inductive (for antennas smaller than a quarter wavelength), and physically small inductors made of realistic materials are lossy.

One final note on efficiency: when discussing aperture antennas (such as parabolic dishes or horn antennas), a term "aperture efficiency" is often used. This has nothing to do with energy loss, but instead refers to the amount of directivity achieved for an antenna of a given size.



(A) shows the pattern of a single element. (B) shows the pattern of the two element array with d=1/2 wavelength

Figure 14 - Pair of Isotropic Antennas

A.5 Methods for Achieving Directivity

Just as larger antennas (up to a point) tend to be more efficient, large antenna size is the essential element for achieving directivity. Consider a pair of isotropic antennas as

shown in Figure 14. If we measure the E field at a constant radius around one of these elements by itself, we would see that it would have a constant amplitude and phase around the entire spherical shell (or circle in 2D). When a pair of such antennas are spaced at a distance d apart, their E fields add at each point, with the resultant amplitude and phase dependant on the amplitude and phase of the E field at that point due to each sub-antenna. The two sub-antennas are called elements, and the combination is an array antenna.

When d is one half wavelength and the two elements are fed in phase, it is easy to see that the combination of the two E fields will have a maximum, over the range of all angles, on a circle which is perpendicular to the plane of the page. In 2D, there would be two maximum points, indicated in the figure. The resulting pattern is donut-shaped (a cross-section of which, in 2D, is a "figure eight"). The maximum D_{MAX} is achieved when the overall radiated power is minimized (since the maximum will always be some constant times twice the E field squared, and D_{MAX} is this maximum divided by the total), which occurs when the elements are spaced a half-wavelength apart, so that their E fields add destructively along the line containing the two elements.

More complex patterns can be achieved by using more elements. The amplitude and phase of each element can be adjusted as well to make certain changes to the pattern. The principal results of such changes would be the reduction of sidelobe magnitudes and altering the direction of the maximum. In all cases, changing the angle of the field measurement point (the angle of the radius between some reference point on the array and the measurement point) changes the phase of the E field contribution from each element. So, the array can be described as a "spatial filter" which sends more power in

some directions than others. The key requirement for this is that the elements be spaced far enough apart that there are significant changes in the phase difference between elements as the angle changes. Hence the statement that antenna array size is an essential element for directivity.

This idea can be extended from a countable number of individual elements to a continuous current distribution. The array of elements can be considered to be a special case of this, where the distribution is an array of Dirac delta functions, one for each element. It can be easily shown that the radiation pattern is in fact the spatial Fourier transform of this current distribution.[8]

There are three main categories of directional antennas: active arrays, passive arrays, and aperture antennas. The two-element case considered earlier (as well as its logical extension to N elements) is an example of an active array. It is termed active because power is fed to each of the elements and the direct interaction among the elements (i.e., power received in one element due to another) is negligible in determining the antenna radiation pattern. In this case, the maximum directivity which can be achieved for N identical elements is N times the individual element D_{MAX}. In addition, the overall function D_{MAX} can be factored into the product of an array factor (depending only on the configuration of the array) and an element factor (which is simply the directivity function of an individual element). No matter how many elements are in the array, the minimum beamwidth is on the order of

$$\frac{\lambda}{w} \text{ radians} \qquad \textbf{Equation 7-4}$$

where w is the total width (in the direction perpendicular to the pointing direction) of the array. Adding additional elements while not changing w (i.e., making the array less sparse) increases the directivity not by making the main beam narrower but instead by decreasing the magnitudes of the sidelobes, which represent power wasted by not being sent in the main beam.

Passive arrays (such as Yagi arrays) operate by diffraction (rather than pure interference in the case of active arrays). Only one or a small number of elements are driven (i.e., have power fed to them) and the antenna relies on interactions among the elements to excite currents in the entire array and produce the desired pattern. When compared to active arrays, passive ones usually have the elements spaced closer together (about 1/4 to 1/10 of a wavelength) and oriented so that their own main beams intersect (so that the coupling between them is maximized). Passive arrays are more difficult to design (and less tolerant of position error of the elements) because of the need to consider complex interactions among elements, but it is possible to achieve a D_{MAX} on the order of N^2 times the D_{MAX} of the individual elements.

Aperture antennas use a continuous current distribution rather than discrete elements to produce a given pattern. Examples include parabolic "dish" antennas, horn antennas, lens antennas, and patch antennas. In each of these, some surface (whether it is driven directly or is a reflector or waveguide wall) has a current distribution which produces radiation. The radiation pattern is more restricted than an array but the antenna is simpler and easier to design.

A.6 Summary

Antennas act as reciprocal impedance matching devices between guided wave structures (like transmission lines) and free space. As such, they can be fully described by two quantities: efficiency and the directivity function (or radiation pattern). For practical antennas, both are related to the antenna's size. It is difficult to make efficient antennas which are considerably smaller than ¼ wavelength in maximum dimension. To make a highly directive antenna, it must have a maximum dimension much larger than one wavelength. In order to receive or transmit a stronger signal, it is necessary to increase maximum directivity or efficiency. There is no such thing as an efficient, passive(no special signal processing) antenna which is "better" or receives/transmits a stronger signal without having to be pointed more accurately. Directivity is achieved by assembling arrays of antennas (which can be "steered" by varying the phase of the element excitation) or by spreading out the antenna current distribution over a large area with a reflector or similar structure.

APPENDIX B

AFV SCHEMATICS

Table 3: AVR to PC Packet Format

AVR to PC

0xFF	0xFF	0xAB	e0	C	e1	C	e2	C	e3	C	e4	C	e5	C	e6	C	ml	C	mm	C	r0	C	r1	C	r2	C	r3	C	xvl	C	xvm	C	yvl	C	yvm	C	zvl	C	zvm	C	phl	C	phm	C	thl	C
thm	C	psl	C	psm	C	xrl	C	xrm	C	yrl	C	yrm	C	zrl	C	zrm	C	w1l	C	w1m	C	w2l	C	w2m	C	w3l	C	w3m	C	w4l	C	w4m	C	c1l	C	c1m	C	c2l	C	c2m	C	c3l	C	c3m	C	c4l	C	c4m	C
v1	C	v2	C	v3	C	v4	C	vm	C	crl	C	crm	C

NOTE: Data is actually sent LSbit first. Each byte is preceded by a start bit (logic low) and ends with a stop bit (logic high). Idle time is filled in by sending 0x55 bytes.

Key:
e0 through e6 - escape bytes 0 through 6
C - the ones complement (XOR by 0xFF which is the same as flipping all bits) of the previous byte. Note that if C would be equal to the flag byte (0xAB) then it would be decremented by one to be 0xAA.
ml - AFV mode unsigned integer LSByte
mm - AFV mode unsigned integer MSByte
r0 through r3 - response bytes, which allow AVR to read back EEPROM contents for verification
xvl,xvm,yvl,yvm,zvl,zvm - X,Y,and Z velocity signed integers (LSByte and then MSByte) in millimeters/second
phl,phm,thl,thm,psl,psm - Phi, Theta, Psi angle signed integers (LSByte then MSByte) in centidegrees
xrl,xrm,yrl,yrm,zrl,zrm - X(Phi),Y(Theta),Z(Psi) Euler Angle Derivative signed integers (LSByte then MSByte) in centidegrees per second
w1l,w1m,w2l,w2m,w3l,w3m,w4l,w4m - prop actual RPM signed integers (LSByte then MSByte)
c1l,c1m,c2l,c2m,c3l,c3m,c4l,c4m - prop commands sent to MCBs as RPM signed integers (LSByte then MSByte)
v1,v2,v3,v4 - MCB motor supply voltage ADC code words (unsigned characters, multiply by 0.19454 to get volts)
vm - main electronics battery voltage ADC code word (unsigned character, multiply by 0.19454 to get volts)
crl,crm - LSByte and MSByte of 16bit CRC checksum (to provide error checking on RF link). CRC is computed on entire packet from ml to vm, before escaping and without the C's. Escaping and adding C's happens after CRC computation because it must apply to CRC as well.

Escape Byte Description:

Escape bytes each have the format 0XXXXXXX with the MSbit on the left and always zero. Each of the X's is an escape bit. For Escape byte 0, bit 0 (LSbit) covers ml, bit 1 covers mm, bit 2 covers r0, etc. up to bit 6 which covers xvl. The LSbit of escape byte 1 then covers xvm and so on. Escape byte 6 is not completely used, it's bits 6 and 5 do not cover anything but may be used for the future.

Escaping ensures that no flag characters can occur within a packet. If a byte would have been 0xAB, it will be zeroed and it's corresponding escape bit will be set. Escape bytes themselves cannot be 0xAB because 0xAB has its MSbit (bit7) set and all escape bytes have this cleared.

Table 4: PC to AVR Packet Format

PC to AVR

0xFF	0xFF	0xAB	e0	C	Q	C	d0	C	d1	C	d2	C	d3	C	crl	C	crm	C

NOTE: Data is actually sent LSbit first. Each byte is preceded by a start bit (logic low) and ends with a stop bit (logic high). Idle time is filled in by sending 0x55 bytes.

Key:
e0 - escape byte 0
Q - command byte
d0 through d3 - data bytes (vary depending on command)
crl,crm - LSByte and MSByte of CRC checksum

Escape Byte Description:

Escape bytes each have the format 0XXXXXXX with the MSbit on the left and always zero. Each of the X's is an escape bit. For Escape byte 0, bit 0 (LSbit) covers Q, bit 1 covers d0, bit 2 covers d1, etc. up to bit 6 which covers crm

AVR Command Description:

Value	Name	Data byte usage
0	Zero IMU	none (simply sends IMUZERO command to DSP)
1	Load Controller Data	writes bytes d0 d1 d2 d3 at current address in main AVR's EEPROM (d0 is written at current address, d1 at current address plus 1, etc.) Use command 16 to set current address before sending this command
2	Reset DSP	none (resets DSP)
3	IMU Recal	none (commands DSP to recal IMU)
4	Idle	none (makes AFV enter idle mode)
5	Test Autodescent	none (makes AFV enter Autodescent mode)
6	Test Rampdown	none(makes AFV enter Rampdown mode)
7	Keepalive	none(when not idle, must send this command at least at several Hz to keep AFV from going failsafe)
8	Manual/non-flight	none(makes AFV enter Manual/non-flight mode)
9	Manual/flight	none(makes AFV enter Manual/flight mode)
10	Controlled Flight	none(makes AFV enter Controlled Flight mode)
11	Command Motor 1	d0,d1 are LSByte, MSByte of signed RPM command to prop 1 (for use in Manual modes)
12,13,14 are same as 11 but for props 2,3,4 respectively		
15	Cancel IMU Recal	none(cancels IMU recal if in progress)
16	Set Controller Data Address	d0,d1 are LSByte,MSByte of address in AVR's EEPROM to read four bytes of controller data from. Address pointer is set to this location so next write will be to this location and also current contents are read out into the response bytes (see AVR to PC packet structure)
17	Load Controller into DSP	none(transfers contents of controller data in AVR's EEPROM into DSP)
18,19,20,21 Load new setpoint into DSP		d0,d1 are LSByte,MSByte of signed setpoint. 18,19,20 are X,Y,Z velocity setpoints in millimeters/sec. 21 is Yaw angle in centidegrees.

Table 5: AVR to DSP Packet Format

AVR to DSP

0xFF	0xAB	e0	e1	Q	d0	d1	d2	d3	d4	d5	d6	d7

Key:

e0,e1 - escape byte 0 and escape byte 1
Q - command byte
d0 through d7 - data bytes (vary depending on command)

Escape Byte Description:

Escape bytes each have the format 0XXXXXXX with the MSbit on the left and always zero. Each of the X's is an escape bit. For Escape byte 0, bit 0 (LSbit) covers Q, bit 1 covers d0, bit 2 covers d1, etc. up to bit 6 which covers d5.. The LSbit of escape byte 1 then covers d6 and so on. Escape byte 1 is not completely used

DSP Command Description:

Value	Name	Data byte usage
0	IMUZERO	none (just zeros a bunch of DSP internal variables)
1	LOAD_CONTROLLER_DATA	d0=LSByte of data address, d1=MSByte of data address, d2 through d5 are bytes that represent a single precision floating point number to be loaded into the controller data array on the DSP
2	IMURECAL	none (just recalibrates IMU)
3	START_CONTROLLER	none(just turns on controller)
4	STOP_CONTROLLER	none(just turns off controller)
5	SETPOINT	d0,d1=LSByte and MSByte of X velocity command, d2,d3 are same for Y vel, d4,d5 for Z vel, and d6,d7 for Yaw angle. Velocity commands are in millimeters/second, Yaw angle command is in centidegrees.
6	CANCEL_RECAL	none(just cancel IMU recal if it is in progress)

Table 6: DSP to AVR Packet Format

DSP to AVR

0xFF	0xAB	e0	e1	e2	e3	e4	xl	xm	yl	ym	zl	zm	xvl	xvm	yvl	yvm	zvl	zvm	phl	phm	thl	thm	psl	psm	xrl	xrm	yrl	yrm	zrl	zrm	w1l	w1m	w2l	w2m	w3l	w3m	w4l	w4m
0xFF	0xAB	e0	e1	e2	e3	e4	x l	x m	y l	y m	z l	z m	xv l	xv m	yv l	yv m	zv l	zv m	ph l	ph m	th l	th m	ps l	ps m	xr l	xr m	yr l	yr m	zr l	zr m	w1 l	w1 m	w2 l	w2 m	w3 l	w3 m	w4 l	w4 m

Key:

e0 through e4 - escape bytes 0 through 4
xl,xm,yl,ym,zl,zm - X,Y,and Z position (currently 0) signed integers (LSByte then MSByte) in centimeters
xvl,xvm,yvl,yvm,zvl,zvm - X,Y,and Z velocity signed integers (LSByte and then MSByte) in millimeters/second
phl,phm,thl,thm,psl,psm - Phi, Theta, Psi angle signed integers (LSByte then MSByte) in centidegrees
xrl,xrm,yrl,yrm,zrl,zrm - X(Phi),Y(Theta),Z(Psi) Euler Angle Derivative signed integers (LSByte then MSByte) in centidegrees per second
w1l,w1m,w2l,w2m,w3l,w3m,w4l,w4m - prop RPM command signed integers (LSByte then MSByte)

Escape Byte Description:

Escape bytes each have the format 0XXXXXXX with the MSbit on the left and always zero. Each of the X's is an escape bit. For Escape byte 0, bit 0 (LSbit) covers xl, bit 1 covers xm, bit 2 covers yl, etc. up to bit 6 which covers xvl. The LSbit of escape byte 1 then covers xvm and so on. Escape byte 4 is not completely used

Table 7: AFV Main Board 2 Connector Pinouts

AFV Main Board 2 Connector Pinouts		**RC Command Receiver Connector**	
		1	5V power
JTAG Connector		2	Ch 1 (Throttle)
Interface to Atmel JTAG ICE Programmer,		3	Ch 2 (Aileron)
refer to JTAG ICE manual for pinout		4	Ch 3 (Elevator)
SPI ICSP Connector		5	Ch 4 (Rudder)
Interface to Atmel STK500 Programmer (we do not use)		6	ground
Refer to STK500 manual for pinout.		**DSP Board Connector**	
Motor Control Board Connectors		1	12V gate supply
1	Gate Supply	2	main battery positive
2	5V power	3	5V power
3	SDA(serial data line)	4	DSP reset pin
4	SCL(serial clock line)	5	ground
5	Ground	6	Data In to AVR
Serial Debug Port Connector		7	ground
1	5V power	8	Data Out from AVR
2	Data Out from AVR	9	ground
3	ground	10	Clock to AVR
4	Data In to AVR	11	ground
5	ground	12	TFS(to AVR)
		13	ground
		14	RFS(from AVR)

Table 8: AFV Mode Table

Number	Mode Name	Failsafe Behavior	Description
0	IDLE	None	AFV sends telemetry but nothing else, motors disabled
1	IMURECAL	None	Not currently used, the AFV remains in Idle mode during IMU calibrations
2	MANUALNONFLIGHT	Motors off if no KEEPALIVE commands from base station	Motor commands are directly accepted from base station, motors active
3	CONTROLLEDFLIGHT	Goes to autodescent mode if no KEEPALIVE commands from base station, goes to rampdown if DSP stops communicating	The controller on the DSP is active and sends commands directly to motors, motors active. Controller LED is on constantly on DSP.
4	AUTODESCENT	Goes to rampdown if DSP stops communicating	DSP still generates motor commands and motors active, but a slow descent is commanded. Terminates after 20 seconds and goes to Idle
5	RAMPDOWN	None, autoterminates in 20 seconds	Ramps all four motors down at 10 RPM/second. Terminates after 20 seconds, goes to Idle
6	MANUALFLIGHT	Goes to rampdown if no KEEPALIVE commands from base station	Same as MANUALNONFLIGHT except for failsafe behavior

Table 9: Controller Data Memory Block

Variable Name	EEPROM Address Range
Nominal Thrust per propeller(N)	0-3
Thrust Coefficient (RPM/root(Newton))	4-7
X acceleration low pass filter(LPF) time constant (s)	8-11
Y acceleration LPF tc	12-15
Z acceleration LPF tc	16-19
Phi rate LPF tc	20-23
Theta rate LPF tc	24-27
Psi rate LPF tc	28-31
Gain matrix 1,1 entry (K1,1)	32-35
K1,2	36-39
……	……
K4,15	268-271

Table 10: Motor Controller Board Connector Pinouts

AFV Motor Control Board Connector Pinouts	
JTAG Connector	
Interface to Atmel JTAG ICE Programmer,	
refer to JTAG ICE manual for pinout	
Hall Effect Sensor Connector	
1	5V power
2	Ground
3	Sensor 1
4	Sensor 2
5	Sensor 3
Encoder Connector	
1	Ground
2	N/C
3	Ch A
4	5V power
5	Ch B
Main Board Connector	
1	Gate Supply
2	5V power
3	SDA(serial data line)
4	SCL(serial clock line)
5	Ground

Table 11: Motor Pinouts and Commutation Table

MaxCim MAXN32-13D Motor Pinout and Commutation Diagrams

Motor Drive Phases	
White	P1(A)
Red	P2(B)
Black	P3(C)
Hall Effect Cable	
Black	Ground
Red	5V power
Blue	Sensor 1
Yellow	Sensor 2
Green	Sensor 3

Commutation Diagrams

State	I	II	III	IV	V	VI
Sensor 1	L	H	H	H	L	L
Sensor 2	H	H	L	L	L	H
Sensor 3	L	L	L	H	H	H
For Clockwise rotation as viewed from back of the motor						
Phase 1	L	Z	H	H	Z	L
Phase 2	H	H	Z	L	L	Z
Phase 3	Z	L	L	Z	H	H
For Counter-Clockwise rotation						
Phase 1	H	H	Z	L	L	Z
Phase 2	Z	L	L	Z	H	H
Phase 3	L	Z	H	H	Z	L

For sensors: H=high(5V nominal), L=low(0V nominal)
For phases: H=positive,L=negative,Z=open

Table 12: AFV Navigation Performance

AFV Navigation Performance	(note: approximate)
Quantity	**Value**
On Ground	
Roll, Pitch Angle error	0.1 degree (from accel offsets)
Yaw Drift	3 deg/hour
Velocity Error	0.2 meter/second
In Flight	
Roll, Pitch Angle Error	Untested (similar to above)
Yaw Drift	Untested (similar to above)
Velocity Error	About 0.5 meter/second

Table 13: Motor Control Board Specifications

AFV Motor Control Board Specifications			
Assumptions: code version 2 (last modified by SHB), MaxCim MaxN32-13D motor,			
100:15 belt drive, APC sport 18x6 propeller, forced air cooling from propeller			
Specification name	**Value**	**Tolerance**	**Notes**
Logic supply voltage	5V	+/- 5%	
Gate safety supply voltage	12V	+/-10%	
Nominal logic supply current	50mA		
Nominal gate supply current	30mA		
Max. instantaneous battery current	300A	guess	LiPoly 7cellsx2parallel short circuit current=abt 100A
Max. sustained battery current	40A		At 100% duty, would be less at lower duty
Max. instantaneous motor current	300A	guess	Could be encountered in heavy breaking
Max. sustained motor current	40A		MCB limitation, motor limitation is 70A
Motor supply voltage range	0-40V		MCB limitation
Software failsafe timeout	0.9 sec		Setpoint is set to 0 rpm if this time passes without command
Max. command update rate	300 Hz		Recommended is 100Hz
Command update wait time	400 usec		After transmitting command, TWI master must wait
			this time before initiating receive from MCB
Local control loop update frequency	244 Hz		
Maximum command latency	1 cycle		cycle means one 244Hz control loop cycle
Measured system time constant	40 msec		
Effective P gain	0.33 V/rad/sec		assuming 7 cells,3.8 V per cell,linear PWM/V map
Effective I gain	1.7 V/rad		same assumption
Effective D gain	0		
Immediate encoder resolution	14 RPM		Affects P term but not I term, so that average RPM has
			1 RPM resolution but instantaneous is 14 RPM.
Max/min RPM command	-32768/+32767		
Encoder RPM limitation	+/- 11800		
Motor max allowable RPM at propeller	+/- 7500		
Max/min system achievable RPM	+5200/-4900	(approx)	assuming 7 cells, 3.8V per cell, 0.07 ohms per cell
Motor current limit (from -1000 RPM to +1000 RPM)	10A	(approx)	Enforced by PWM limits
Motor current limit (outside of -1000 to +1000 RPM)	50A	(approx)	Enforced by PWM limits
Integral control anti-windup limit	+/- 100% duty		Windup limited to one full +/- 100% duty cycle quantity
PWM frequency	31.25 kHz		8 bit resolution, 0 to 90% duty range
			90% maximum to ensure that IR2133S charge pumps
			always remain charged (they charge during a PWM low)
Max positive thrust production	24.88 N		at 5200 RPM

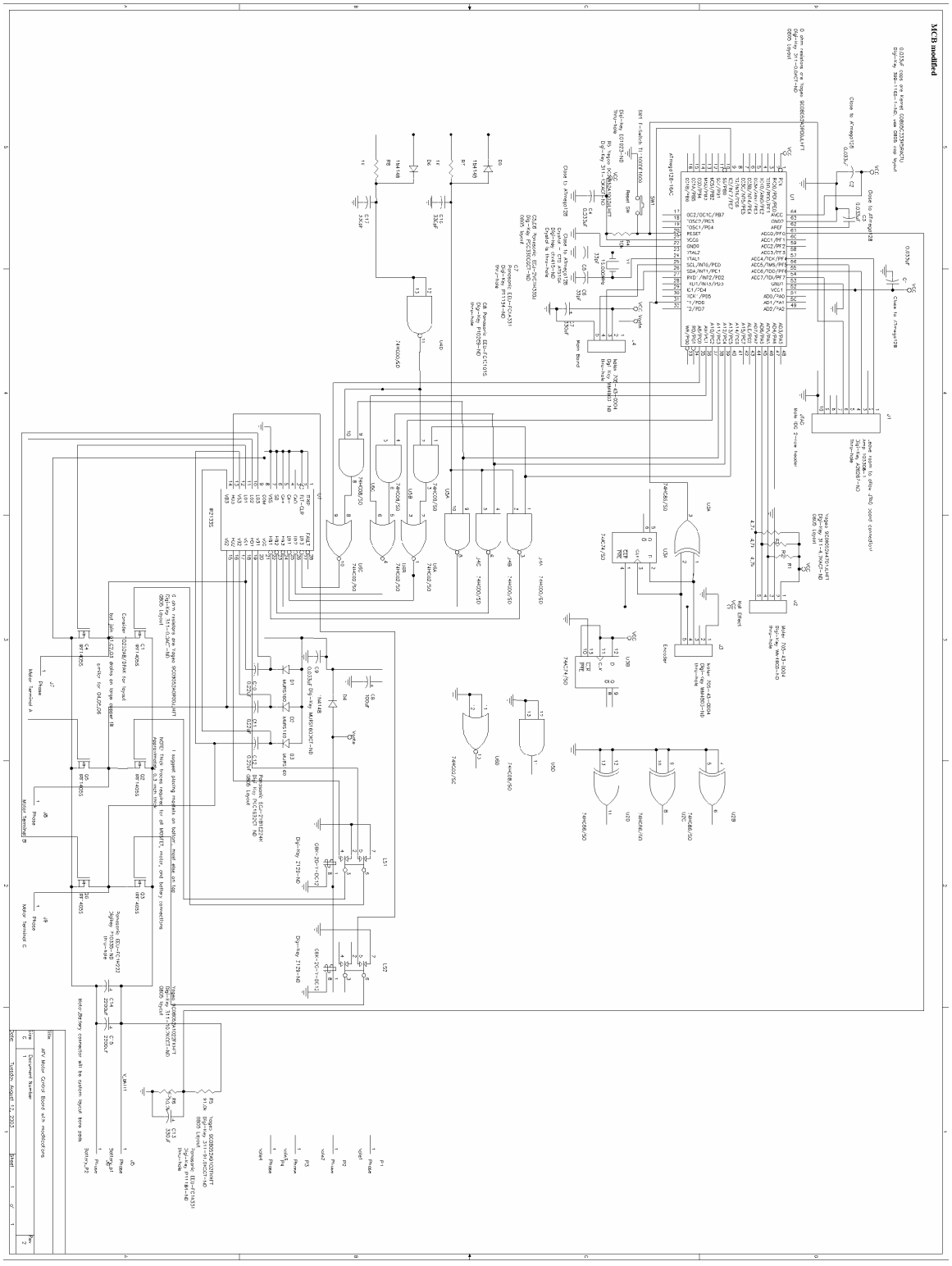


Figure 15 - Motor Control Board Schematic

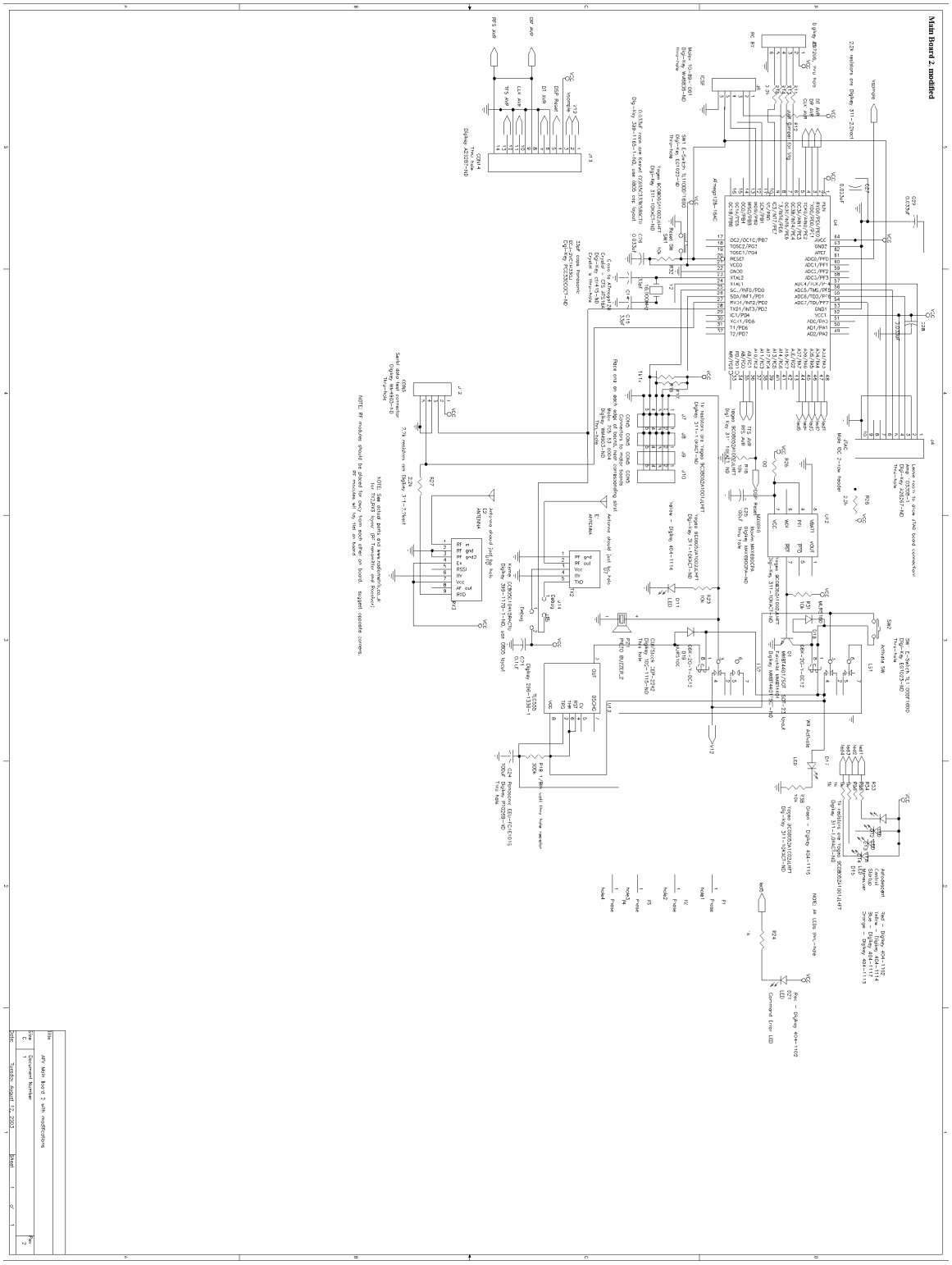


Figure 16 - Main PCB 2 (AVR) Schematic

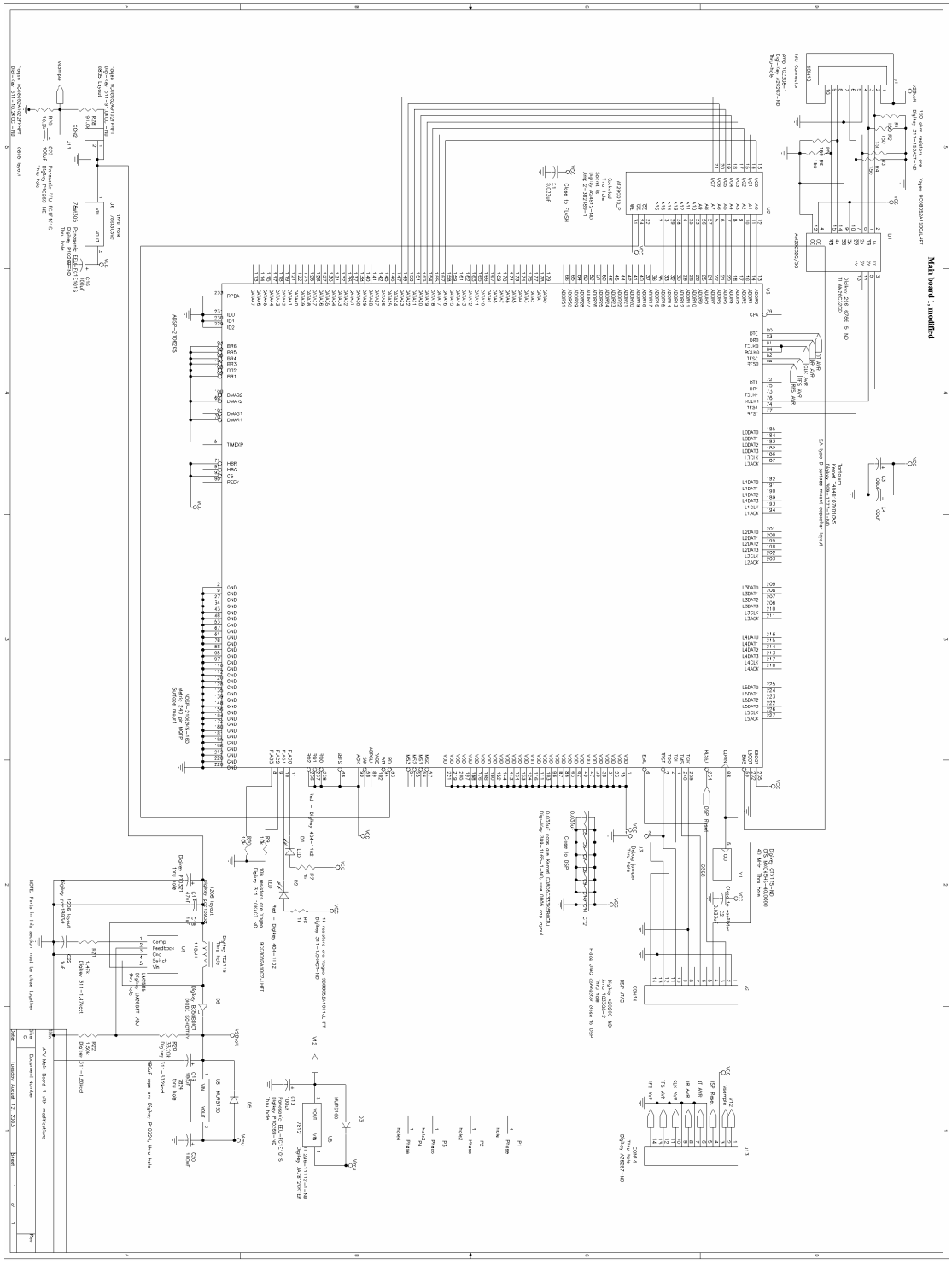


Figure 17 - Main PCB 1 (DSP) Schematic

NOTE: For higher resolution versions of these schematics see APPENDIX C

APPENDIX C

LOCATION OF RESOURCES

In addition to this document, there are two CDs with relevant information for this project. Their contents are detailed below.

Eryk Nice's MS Thesis CD Contents Summary (AFVMechECD)	
X:\Documentation\2003-2004\Designof4RotHoverVehicle\ Designof4RotHoverVehicle.doc	Eryk Nice's MS thesis
X:\Documentation\2001*.*	MechE, EE, and vision system documentation from the 2001 system
X:\Analysis&Simulation*.*	2D and 3D AFV simulations done by Eryk Nice in 2003-2004
X:\Analysis&Simulation\Structure*.*	2003-2004 AFV structural analysis
X:\ProE*.*	2003-2004 AFV mechanical models

Sean Breheny's MS Thesis CD Contents Summary (AFVEECD)	
X:\ 2003 Electronics CD\AFV Electronics Documentation EARLY DRAFT.doc	EE documentation from the 2003-2004 AFV project.
X:\AFVPAA.m	AFV linearized model and H_∞ control design
X:\WASP\Final Release*.*	Final release of the WASP antenna simulation GUI code
X:\WASP*.*	Word DOC and PowerPoint presentation on the Cornell participation in the DARPA WASP Program 2002-2003
X:\Unpublished Docs*.*	Two presentations and one PDF paper on antenna theory and the concept of this research project
X:\Wind Models*.*	Two papers on wind modeling from NASA and the AIAA
X:\2003 Electronics CD\datasheets*.*	Datasheets for various AFV components
X:\2003 Electronics CD\AFV Source Code\latest code*.*	Latest AFV (2003-2004) onboard systems source code
X:\2003 Electronics CD\AFV Source Code\For use with GPS*.*	An unfinished version of the AFV code to interface with a GPS receiver.
X:\2003 Electronics CD\PC-side source code*.*	Source code for the DOS-based PC telemetry program for the 2003-2004 AFV system
X:\2003 Electronics CD\DOS C compiler*.*	Borland Turbo C++ DOS C compiler from around 1993 – used to compile PC-side source code
X:\2003 Electronics CD\matlab code\genl_opt2.m	MATLAB code to find the optimal combination of batteries, motor, gear ratio, and propeller for AFV.
X:\2003 Electronics CD\Schematics*.*	DXFs and high-resolution BMPs of the AFV schematics.

APPENDIX D

AFV OPERATION NOTES

NOTE! Because each motor control unit (MCU) has a unique address and this is coded into the source code, the *MCB_sens2.c* source code must be manually modified for each MCU, inserting the proper address, recompiled, and programmed into that particular MCU. This must be done once for each MCU any time there is an MCU code change.

WARNING: The user should ensure that the AFV is in the MANUALNONFLIGHT mode and that all four motor setpoints are zero before pressing the button to enable the motors. Otherwise, it would be possible for the motors to turn on when the setpoints were not zero and they would follow that command and immediately spin up. When the motors are activated, they may cause the props to move slightly since the controllers onboard the MCUs are always active and may have built up a slight amount of integral windup before the motors were activated.

WARNING: When the motors are active, the props may either be still (commanded 0 RPM) or moving (commanded something else) but the motor warning beeper will be sounding and the yellow “motors active” LED will remain on. NO ONE should approach the AFV while it is in this state. It should be considered possible that at any moment the props could spin up.

WARNING: During software debugging, there were a couple of instances where the main microcontroller did not respond to the command to turn off the motors and continued to supply the “keepalive” signal to the motor watchdog timer even when no

commands were being sent to it. The code was designed to prevent this from happening. The problem that was causing the main microcontroller to lock up in this manner has been fixed, but it still is not known exactly how the code was able to continue to generate the "keepalive" signal while in this state. If this ever happens, the motors should be commanded zero if possible and if it is possible to do so without endangering anyone, someone should disconnect the power to the motors. If this is not possible, the AFV should be left as is and everyone should wait until the main electronics battery has drained and the main electronics assembly can no longer generate the 12 Volt gate drive signal, at which point the beeper will turn off. It would then be safe to approach the AFV.

NOTE: The controller should not be changed in-flight. After the controller gets reloaded into the DSP, the DSP gets placed into "controller off" mode so that this would likely cause a physical crash if done in-flight. The PC-side software currently prevents the controller from being reloaded in flight and this protection should be preserved in any modifications.

NOTE: There is currently a known problem with the software (might be either PC side or onboard) that a new controller must be loaded twice before loading correctly. We did not have time to finish debugging this. It is recommended that controllers always be loaded twice when loading a new one and that the controller be activated and tested on the ground first (without the motors on) to check that it is working properly before flight.

REFERENCES

[1] S. Breheny, AFV Electronics Documentation, AFVEECD:\ 2003 Electronics CD\AFV Electronics Documentation EARLY DRAFT.doc, 2003.

[2] S. Breheny and R. D'Andrea, Using Airborne Vehicle-Based Antenna Arrays to Improve Communications with UAV Clusters, Conference on Decision and Control, 4158-4162, 2003. © 2003 IEEE

[3] R. Chandra, S. Breheny, R. D'Andrea, Antenna Array Synthesis with Clusters of Unmanned Aerial Vehicles. *Automatica* to appear.

[4] J. Miller, R. D'Andrea, S. Breheny, Final Documentation for the Cornell Phased Array Antenna Contribution to the DARPA Wonderfully Agile Sensor Platforms (WASP) Project, AFVEECD:\WASP\Cornell WASP PAA Final Documentation.doc, September 2003.

[5] Several examples:

- P. Castillo, A. Dzul and R. Lozano, Real-time Stabilization and Tracking of a Four Rotor Mini-Rotorcraft, http://www.hds.utc.fr/~castillo/publications/4rotor.pdf
- O. Fong and S. L. Huggins IV, Mechanical and Aerodynamic System Design, AFVMechECD:\Documentation\2001\Mechanical and Aerodynamic System Design.doc, 2001.
- Association for Unmanned Vehicle Systems, International, Aerial Robotics Competition homepage, http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html
- Draganfly, "Draganflyer III", web site http://www.rctoys.com/draganflyer3.php, accessed on March 8, 2003.
- Draganfly Innovations Inc., Draganflyer IV homepage, http://www.rctoys.com/draganflyer4.php
- Kroo and F. Prinz, The Mesicopter: A Meso-Scale Flight Vehicle, Mesicopter homepage, http://aero.stanford.edu/mesicopter/
- Experimental Rocket Propulsion Society, Inc., The GizmoCopter Project homepage, http://gizmocopter.org/
- D. Fitzgerald, XUV-5 PipeDream Quad Project homepage, http://www.eese.bee.qut.edu.au/QUAV/index2.html , 2002
- J. Borenstein, The Hoverbot – An Electrically Powered Flying Robot, http://www-personal.engin.umich.edu/~johannb/hoverbot.htm

- S. Breheny, Design and Implementation of Control and Sensing Electronics for the Cornell Autonomous Flying Vehicle, AFVMechECD:\Documentation\2001\Design&ImpofControl&SensingforCU AFV.doc , 2001
- S. L. Venneri and A. K. Noor, "Plenty of Room in the Air," Mechanical Engineering Magazine, Nov. 2002
- P. Pounds, R. Mahony, P. Hynes, and J. Roberts, Design of a Four-Rotor Aerial Robot, Proc. 2002 Australasian Conference on Robotics and Automation, http://www.araa.asn.au/acra/acra2002/Papers/Pounds-Mahony-Hynes-Roberts.pdf , 2002
- P. Ferguson, F. Busse, and J. How, "Navigation Performance Predictions for the Orion Formation Flying Mission," International Symposium on Formation Flying: Missions and Technologies, Toulouse, France, October 2002

[6] J. Stiles, N. Goodman, and SiChung Lin, "Performance and processing of SAR satellite clusters," Proc. of the 2000 IEEE Geoscience and Remote Sensing Symposium,Honolulu, HI, vol. 2, pp. 883-885, July 2000

[7] S. Breheny, Phased Array Antennas for Autonomous Flying Vehicle Communication, AFVEECD:\Unpublished Docs\antennas.pdf, internal research summary for Prof. Raffaello D'Andrea, Cornell University, March 2002

[8] D. K. Cheng, Field and Wave Electromagnetics, Addison-Wesley, 2nd edition,1992 and
Skolnik, Merrill I., editor, Radar Handbook, McGraw-Hill Book Company, 1970

[9]T. Kailath, Linear Systems, Prentice Hall, 1980

[10] Systron Donner Inc., DQI Digital Quartz IMU User's Guide, 08 April 2002

[11] Bishop Power Products, Online Catalog, http://www.b-p-p.com/Etec1200.htm

[12] J. D. Anderson, Introduction to Flight, McGraw Hill,2nd edition,1985

[13] T. Cimato, MaxCim Motors Technical Specs,
http://www.maxcim.com/technical.html

[14] P. Horowitz, W. Hill, The Art of Electronics, Cambridge University Press,2nd edition, 1989

[15] K. Ogata, Modern Control Engineering, Prentice Hall, 3rd edition,1997

[16] G. Welch, G. Bishop, An Introduction to the Kalman Filter, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, May 2003, http://www.cs.unc.edu/~welch/kalman/ and

Y. Bar-Shalom, X. Rong Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, John Wiley & Sons, 2001

[17] E. B. Nice, M.S. Thesis: Design of a Four Rotor Hovering Vehicle, Cornell University, School of Mechanical and Aerospace Engineering, May 2004 (see also AFVMechECD:\Documentation\2003-2004\Designof4RotHoverVehicle\ Designof4RotHoverVehicle.doc)

[18] http://www.defense-update.com/features/du-2-04/mav-oav.htm

[19] M. I. Skolnik, Introduction to Radar Systems, McGraw-Hill, 2nd edition, 1980

[20] H. S. C. Wang, "Performance of phased-array antennas with mechanical errors," IEEE Transactions on Aerospace and Electronic Systems, vol. 28, no. 2, pp. 535-545, April 1992

[21] W. W. Hansen and J. R. Woodyard, "A new principal in directional antenna design," Proc. IRE, vol. 26, pp. 333-345, Mar. 1938

[22] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, vol. 9, no. 1, pp. 112-147, 1998.

[23] G. E. Dullerud, F. Paganini, A Course in Robust Control Theory, Springer-Verlag, 2000

[24] Information on wind modeling was obtained from:

- Military Specification MIL-F-8785C, Nov 5, 1980, available from www.aoe.vt.edu/~durham/AOE5214/MILSPEC8785C.pdf
- M. Perlmutter, W. Frost, G. H. Fichtl, Three Velocity Component, Nonhomogeneous Atmospheric Boundary-Layer Turbulence Modeling, AIAA Journal, Vol. 15, No. 10, Oct. 1977, 1144-1154
- Zhao Z., Xiao Y., Shi Y., Digital Simulation Technique for the Dryden Atmospheric Model, Acta Aeronautica et Astronautica Sinica (Peoples Republic of China), V. 7, No. 5, Oct. 1986 p 433-443. Trans. by Scitran, Inc., Santa Barbara, Calif., Aug 1988.
- C. G. Justus, W. R. Jeffries III, S. P. Yung, D. L. Johnson, The NASA/MSFC Global Reference Atmospheric Model – 1995 Version (GRAM-95), NASA Technical Memorandum 4715, August 1995

[25] R. D'Andrea, "A Linear Matrix Inequality Approach to Decentralized Control of Distributed Parameter Systems," Proc. 1998 American Control Conference, vol. 3 pp. 1350-1354

[26] R. D'Andrea, C. Langbort, and R. Chandra, "A State Space Approach to Control of Interconnected Systems," Mathematical Systems Theory in Biology, Communication, Computation and Finance (J. Rosenthal, ed.), Springer, IMA Book Series, 2003. To appear

[27] http://www.ntp.org/

[28] B. Sklar, Digital Communications: Fundamentals and Applications, Prentice Hall, 2nd edition, 2001

[29] J. Shau-Shiun Jan, P. Enge, "Using GPS to synthesize a large antenna aperture when the elements are mobile," Institute of Navigation National Technical Meeting, Anaheim, CA, Jan 2000

[30] S. Boyd and R. Lorenz, "Robust minimum variance beamforming," web site http://www.stanford.edu/~boyd/rmvb.html

[31] the Constellation3Di by ArcSecond Inc., see http://www.indoorgps.com/