AIRBORNE INFRARED TARGET TRACKING WITH THE NINTENDO WII REMOTE SENSOR

A Thesis

by

ANDREW WILSON BECKETT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	John Valasek
Committee Members,	Rabi N. Mahapatra
	Tom Pollock
	Thomas W. Strganac
Department Head,	Rodney D. W. Bowersox

December 2012

Major Subject: Aerospace Engineering

ABSTRACT

Intelligence, surveillance, and reconnaissance unmanned aircraft systems (UAS) are the most common variety of UAS in use today and provide invaluable capabilities to both the military and civil services. Keeping the sensors centered on a point of interest for an extended period of time is a demanding task requiring the full attention and cooperation of the UAS pilot and sensor operator. There is great interest in developing technologies which allow an operator to designate a target and allow the aircraft to automatically maneuver and track the designated target without operator intervention. Presently, the barriers to entry for developing these technologies are high: expertise in aircraft dynamics and control as well as in real-time motion video analysis is required and the cost of the systems required to flight test these technologies is prohibitive. However, if the research intent is purely to develop a vehicle maneuvering controller then it is possible to obviate the video analysis problem entirely. This research presents a solution to the target tracking problem which reliably provides automatic target detection and tracking with low expense and computational overhead by making use of the infrared sensor from a Nintendo Wii Remote Controller.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Intelligence, surveillance, and reconnaissance unmanned aircraft systems (UAS) are the most common variety of UAS in use today and provide invaluable capabilities to both the military and civil services. UAS are well-suited to the ISR role: large UAS can remain airborne for far longer than the limits of human endurance without needing to be large enough to hold multiple crews or expose personnel to enemy fire; small UAS can be man-portable and field-deployable to provide instant intelligence to small units. UAS operational requirements vary from one system to another, but generally many personnel are required to operate large UAS: from the pilots and sensor operators working in shifts at the ground control station to the teams of analysts poring over the data take.[1] Small UAS are typically operated from a portable computer and run a few predetermined flight profiles while relying on a human operator to point the sensors. As such, there is significant interest in reducing the workload and personnel requirements of these systems.[2]

Keeping the sensors centered on a target for any extended period of time is a demanding task requiring the full attention and cooperation of the pilot and sensor operator on large UAS. The sensor operator must manually keep the sensor package pointed at the target and the pilot must ensure that the vehicle maintains a trajectory and attitude in which it is possible for the sensors to see the target. This task is even more challenging on small UAS because the operator must steer the sensors without pilot cooperation in an environment where the target may be traversing rapidly with respect to the aircraft. Sensors must be installed on a vibration and vehicle motion-compensated gimbal for this task to be possible. *As such, there is great interest in developing technologies which allow an operator to designate a target and allow the*

vehicle to automatically maneuver and track the designated target without operator intervention.[3]

1.1 Previous Work

Presently, the barriers to entry for developing these technologies are high. Engineers developing solutions require expertise in motion video analysis algorithms as well as in vehicle dynamics and control techniques. Video analysis algorithms are computationally-intensive and the resulting UAS architectures broadly fall into one of two categories: carry a high-performance computer onboard the aircraft to process the motion video or transmit the video to a ground control station for processing and transmit the results back to the aircraft (Figure 1.1).[4][5] The video analysis algorithms themselves are an ongoing research topic which require computational capabilities not found in small embedded systems. [6][7][8]

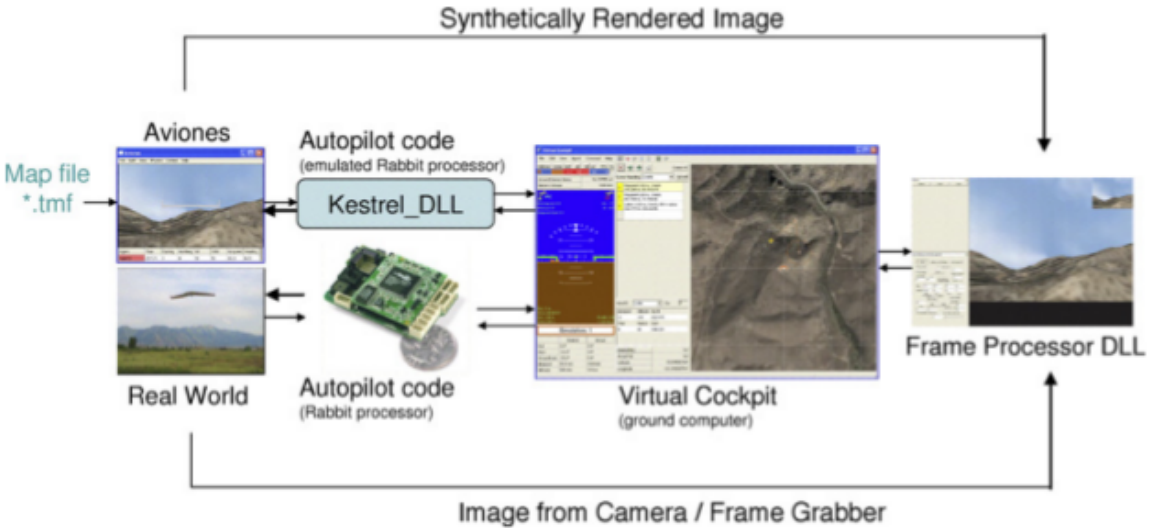

Figure 1.1: Data Flow in a Ground Station-in-Loop Tracking Architecture [5]

Each of these approaches have their disadvantages. Carrying a capable computer onboard restricts the minimum size and complexity of the UAS to a point beyond the capability of many researchers. Transmitting video to the ground for real-time

processing includes the ground station and communications datalink in the automatic control loop which reduces the reliability of the system to that of the communications datalink. In either case, porting the video analysis algorithms to avionics hardware, tuning the algorithms to track the expected types of targets, and providing an easy and intuitive operator interface are ongoing challenges.

However, if the research intent is purely to develop a vehicle maneuvering controller for maintaining the sensors on target then it is possible to obviate the video analysis problem entirely. For this purpose, live motion video and processing are not required if the output from those algorithms can be provided by some other means. Technologies developed in the consumer electronics space for motion tracking are eminently suitable in this regard because they provide a target location with drastically less expense and complexity than full motion video processing solutions.

1.2 Nintendo Wii Remote Sensor

Nintendo manufactures the Wii video game console (Figure 1.2) which is the first video game console to use motion gestures for player control input. Primary user input is through the Wii Remote Control: a handheld wireless controller which features a Bluetooth™ wireless system-on-chip, multiple buttons, three axes of accelerometers and rate gyroscopes, and an infrared camera with integrated signal processing. The Wii uses this camera to detect and track a pair of infrared LED arrays positioned above or below a television display for determining the pose of the Wiimote relative to the screen. Though it is a proprietary piece of consumer electronics, the Wiimote has been thoroughly reverse-engineered by online communities for a variety of machine vision applications.[9]

Figure 1.2: Nintendo Wii Video Game Console [10]

1.3 Research Objectives

The goal of this research is to develop a sensor with low computational overhead which reliably provides all the required features for the vehicle maneuvering controller discussed above to function. Specifically, these required features are:

1. Automatic target detection.
2. Automatic target tracking with at least 1 Hz update rate.
3. Device driver class fits in ArduPilot Mega available free program memory with ArduPlane and experimental control law.
4. Known relationship between sensor frame coordinates and bearing from sensor installation.
5. Targets detectable at a practical orbit radius.

This thesis presents a solution which consists of a fully-characterized sensor based on the Nintendo Wiimote infrared sensor, a device driver class written in C++ which targets the Arduino open-source hardware, open-source software electronics

prototyping platform; and a target beacon which is detectable while aloft at useful distances.

This solution targets Arduino because future work will use this solution to develop and verify vehicle maneuvering controllers for the DIYDrones ArduPilot Mega autopilot. One of the tenets of the Arduino platform is software interoperability: software written for the platform typically requires little or no modification to port the software to different types of Arduino hardware. ArduPilot Mega adheres to the Arduino platform standards and as such can make use of software written for the Arduino platform.

2. SENSOR

This chapter discusses the development and characterization of the sensor hardware and software. The infrared sensor used in the Nintendo Wiimote is a proprietary unit manufactured by PixArt Imaging Systems and as such no information is available from the manufacturer. Though the sensor interfaces are known through reverse-engineering, sensor performance data is provided without reference to how it was obtained. Therefore, all sensor specifications critical to this application are determined experimentally.

2.1 Hardware

The Wiimote infrared camera consists of a monochrome CMOS imager with a physical resolution of 128 by 96 pixels, small-aperture optics, and a filter stack tuned for sensitivity to 940 nm infrared light. The sensor also includes an integrated digital signal processor using PixArt's Multi-Object Tracking™ engine.[11] This DSP performs subpixel sampling to provide an effective imaging frame measuring 1024 by 768 pixels.

Though it is possible to interface with the Wiimote without modification using the Bluetooth™ wireless communication protocol, in practice this is not desirable in a UAS because electromagnetic interference (EMI) from these radios may degrade performance of the aircraft telemetry and control radios. Since the infrared sensor electrical specification is reverse-engineered, there is no issue with dismounting the infrared sensor from the Wiimote printed circuit board (PCB) and installing it on a custom carrier board (Figure 2.1).

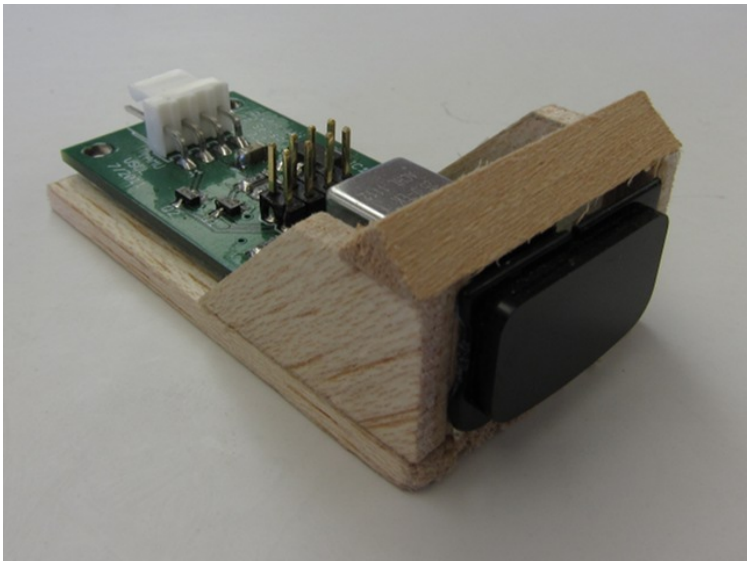

Figure 2.1: Wiimote IR Sensor as Installed on Carrier Board with IR Filter

2.1.1 Carrier Board

The sensor requires a supply voltage of 3.3 VDC and interfaces with other devices using the Philips/NXP Inter-Integrated Circuit (I^2C) bus protocol in fast mode (400 kbit/s). Since both standard Arduino hardware[12] and ArduPilot Mega[13] use 5 VDC for supply voltage and logic signal voltage, the carrier board includes a 3.3 VDC output low-dropout linear voltage regulator and logic signal level translation circuitry to bidirectionally convert between 3.3 VDC and 5 VDC logic levels. The carrier board includes a 25 MHz quartz crystal oscillator to supply the 24–26 MHz clock signal necessary for sensor operation. Figure 2.2 shows the carrier board schematic and printed circuit board layout.

2.2 Software

The sensor is intended to be used with the DIYDrones ArduPilot Mega autopilot and to that end the sensor device class is written in C++ using object-oriented programing paradigms. Arduino software is written in the Wiring programming lan-

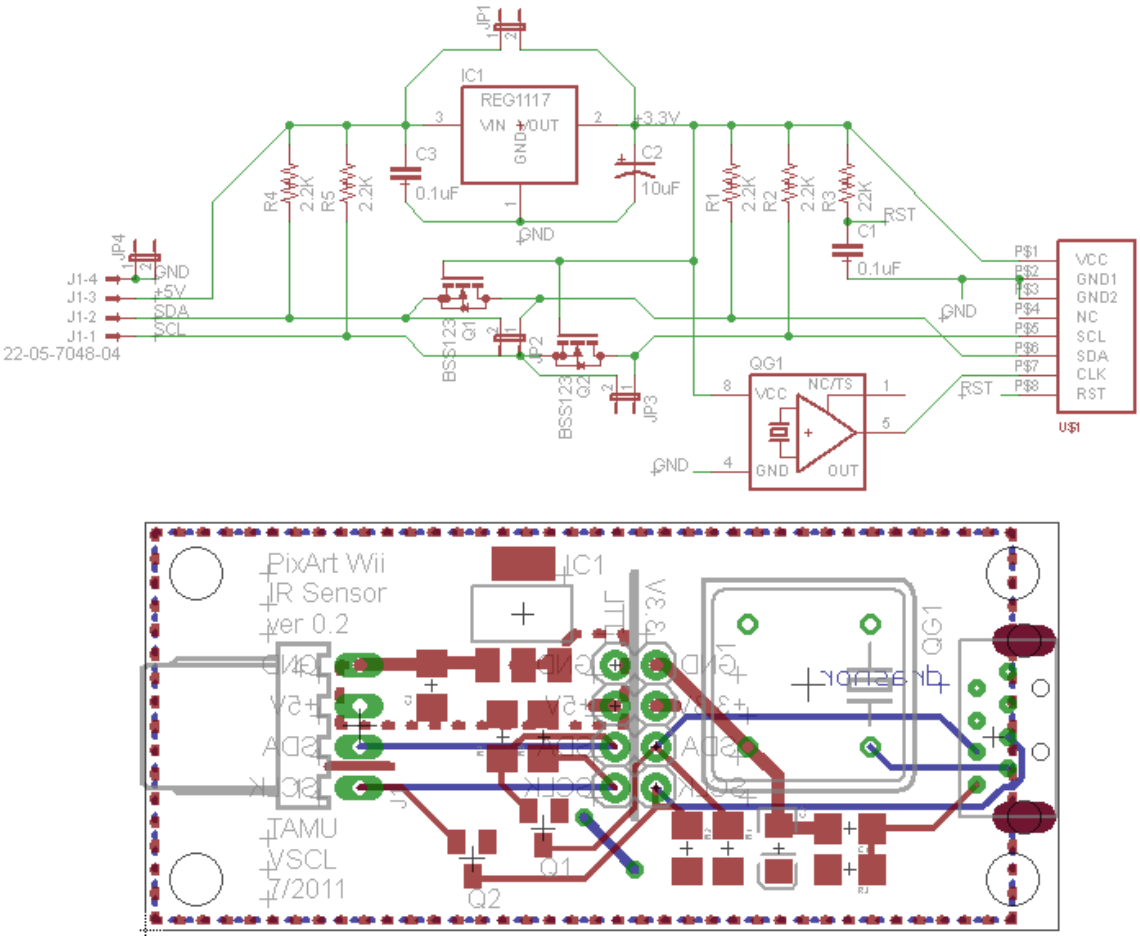

Figure 2.2: Carrier Board Schematic and Printed Circuit Board Layout

guage which is a dialect of the C++ programming language. Arduino heavily favors object-oriented programming paradigms and implements the bulk of the hardware device drivers and Wiring language features as C++ classes. Hardware abstraction is accomplished using class inheritance: each hardware board provides hardware-specific implementations of virtual methods inherited from the Arduino base classes.

The infrared sensor features three data reporting modes: Basic, Extended, and Full. In Basic mode, the sensor reports only the X-Y coordinates of the center of each tracked blob. In Extended mode, the sensor reports the approximate pixel size of each blob in addition to the basic mode information. In Full mode, the sensor reports the X-Y coordinates of the corners of the bounding box containing each tracked blob in addition to the extended mode information. Since only the target X-Y image frame coordinates are needed for this application, the sensor device class only implements Basic mode functionality. Up to four blobs can be tracked simultaneously and the blob tracks are refreshed at 100 Hz. It is important to note that it is not possible to obtain the image seen by the camera because the sensor is only capable of reporting processed data regarding the tracked blobs. The sensor device driver class is based on the PVision device driver class by Stephen Hobley[14] but differs in that it targets Basic mode operation and implements online sensitivity setting change, tracks the number and ID of each active blob, and is compliant with the ArduPilot Mega recommended code style.

Four single-byte sensitivity parameters control sensor detection and tracking criteria:

1. **MAXSIZE (p0)**: Maximum blob size. The Wii uses values from 0x62 to 0xC8.

2. **GAIN (p1)**: Sensor gain. Smaller values permit detection and tracking of

dimmer targets.

3. **GAINLIMIT (p2)**: Sensor gain limit. Unknown effect but must be less than GAIN for the sensor to function.

4. **MINSIZE (p3)**: Minimum blob size. The Wii uses values from 0x03 to 0x05.

Empirical testing was performed to verify proper sensor operation and to debug sensor driver class methods. This testing consisted of detection and tracking attempts with default sensitivity settings tracking a lit candle at distances of <1 m. All further testing and characterization discussed below was performed using a beacon made from these components:

1. **Lamp**: Osram Semiconductor SFH-4751 wide-angle high-output 940 nm infrared emitter

2. **Power Supply**: LEDdynamics LUXdrive BuckPlus 7023-D-E-1000 1 A constant-current LED drive

The sensitivity settings used for the sensor characterization discussed below result from indoor empirical testing to determine the highest sensor gain attainable without spurious target detection. However, the default sensitivity settings were used for the outdoor flight testing because the sensor experienced spurious detection events outdoors when configured with the indoor sensitivity settings. The characterization testing was not repeated using the outdoor settings because the default sensitivity settings do not differ in GAIN or GAINLIMIT by more than 0x18 or 9.4% of range.

2.3 Characterization

The most critical parameters to know in determining the suitability of this sensor for airborne target tracking are angle of view, minimum irradiance to detect and track a blob, and optical filter pass band.

2.3.1 Angle of View

The sensor angle of view is estimated to be 40° horizontal by 30° vertical. The experimental setup used to determine these angles is shown in Figure 2.3:

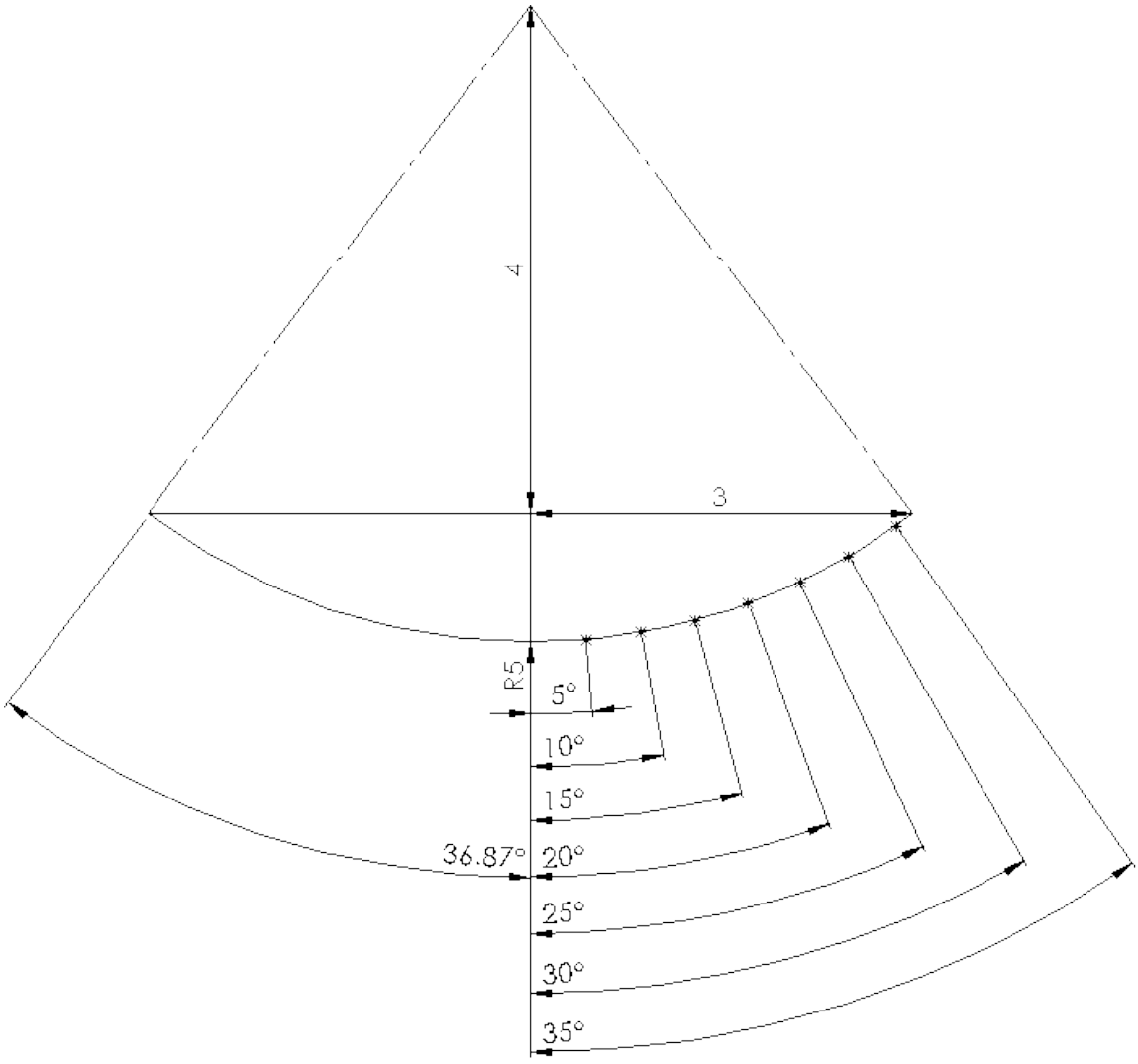

Figure 2.3: Angle of View Measurement Experimental Setup

The test beacon was placed at 5° increments along a 5 m radius arc facing the origin and the corresponding X-Y coordinates reported by the sensor were recorded. The experiment was performed once with the sensor horizontal axis parallel to the

test pattern to determine horizontal angle of view and once with the vertical axis parallel to the test pattern to determine vertical angle of view. The angle of view was determined by averaging the pixel displacements between each 5° point to determine °/pixel, then multiplying by the X and Y resolution to determine angle of view. The resulting angle of view is 40° horizontal by 30° vertical. Since the coordinates of the 5° marks vary linearly, the resulting pixel width is a constant 141 arcseconds/pixel on both axes.

The test pattern was constructed using a tape measure in United States customary units with a precision of 1/8 in. and marked out using a lump of mortar leaving marks roughly 1/2 in. wide. Each successive 5° mark is measured with respect to the previous mark and as such these errors are cumulative across the full range. The sigma uncertainty of the angle of view measurement is 7.80e-03.

2.3.2 Detection Sensitivity

The sensor detection sensitivity was determined by finding the maximum distance at which the test beacon could be reliably detected and determining the irradiance at that distance. According to the manufacturer, the radiant intensity of this LED varies between 630 mW/sr and 1000 mW/sr between batches. Since there is a large uncertainty in this parameter, the actual irradiance is determined experimentally rather than calculating from radiant intensity. The spectral irradiance of the test beacon as measured by a StellarNet Black CXR-SR-50 spectrometer with a CR2 180° cosine sensor head is shown in Figure 2.4. The spectral resolution of the CXR-SR-50 is 0.5 nm.

The Wiimote infrared sensor detects the test beacon reliably at a distance of 5 m. At this distance, the test beacon irradiance E_e is 0.97 µW/cm^2. This value is obtained by extrapolating the fitted curve shown in Figure 2.5. The data points

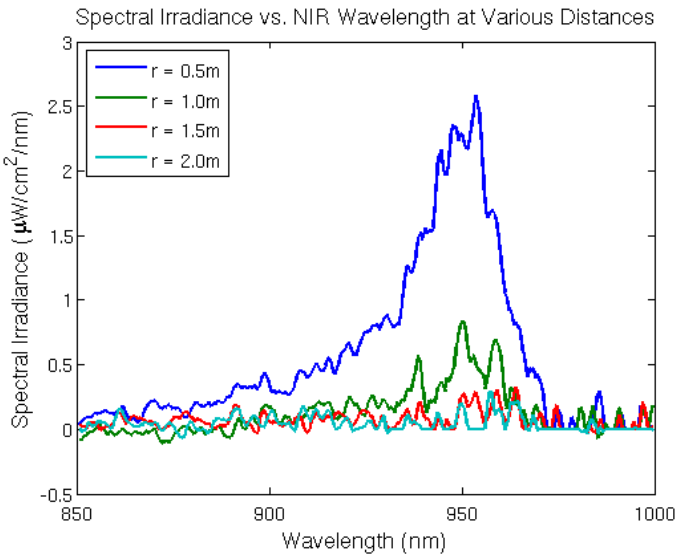

Figure 2.4: Spectral Irradiance of Test Beacon at Various Distances

shown in Figure 2.5 are the area under the curve at each distance shown in Figure 2.4 and are found using the trapezoidal rule with 0.5 nm spacing.

Target detection was only attempted at 0.5 m increments. The sigma uncertainty of the minimum detectable irradiance calculation is 3.36e-02.

2.3.3 Optical Filter Performance

The Wiimote infrared sensor filter is a colored plastic filter exhibiting the attenuation curve shown in Figure 2.6. This attenuation curve was found by comparing direct sunlight to filtered direct sunlight using a StellarNet Black CXR-SR-50. The -3 dB attenuation point occurs at a wavelength of 866 nm. The gaps in Figure 2.6 near 930 nm and above are due to gaps in the solar spectrum. The sensor is not sensitive at wavelengths longer than 1500 nm due to limitations in silicon CMOS imaging sensors.

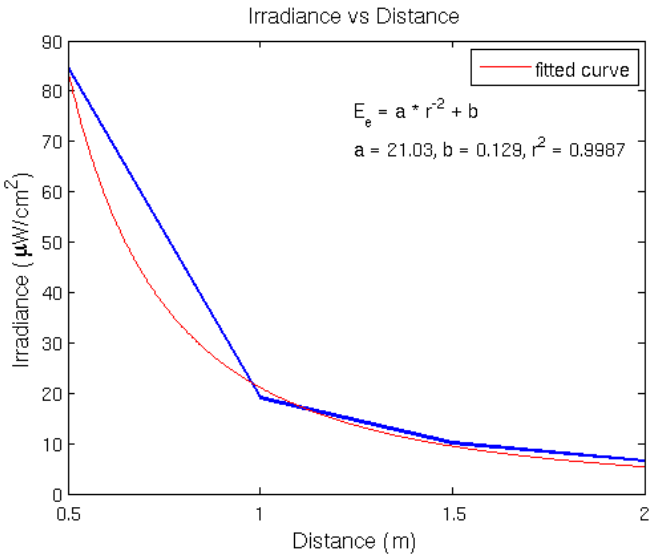

Figure 2.5: Irradiance of Test Beacon vs Distance

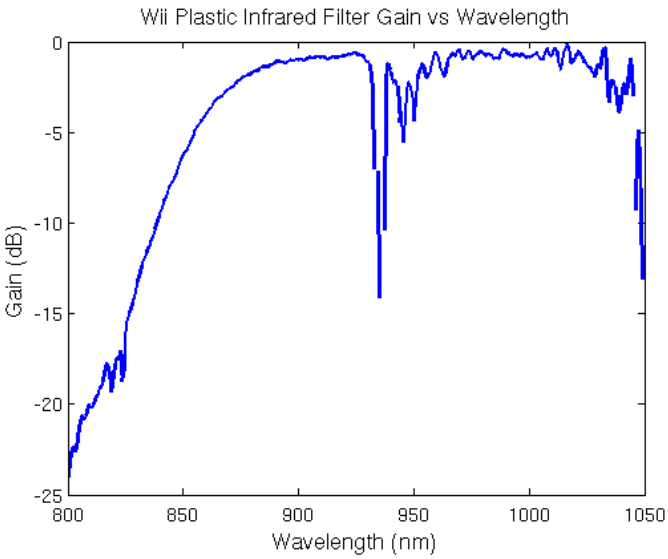

Figure 2.6: Wii Plastic Infrared Filter Gain vs Wavelength

3. BEACON

This chapter discusses the requirements, design, and construction of the target beacon for use in sensor verification flight testing.

3.1 Performance Requirements

The Nintendo Wiimote tracks a pair of infrared LED arrays contained within the Wii Sensor Bar to determine the pose of the Wiimote relative to the user's television screen (Figure 1.2). These LEDs are adequate for this purpose in mid to low-light conditions and distances of a few meters but a much brighter beacon is necessary for detection at greater distances and in brighter lighting conditions. This is because irradiance varies with distance as shown in Equation 3.1 for an isotropic point source:[15]

$$E_e = \frac{\Phi_{e,source}}{4\pi r^2} \tag{3.1}$$

Equation 3.1 describes irradiance for a perfectly spherical radiation pattern, but in practice the beam should be hemispherical as described by Equation 3.2. A hemispherical beam is desirable because the best relative pose of the aircraft and beacon for the vehicle maneuvering and tracking application is unknown. A hemispherical beam is achieved by the simple expedient of placing a flat reflector below the beacon lamp.

$$E_e = \frac{\Phi_{e,source}}{2\pi r^2} \tag{3.2}$$

Equation 3.3 is a straightforward manipulation of Equation 3.2 and gives the required radiant power for sensor detection as a function of range for $E_e = 0.01$ W/m^2

(rounded up from 0.097 W/m^2 found in Chapter 2) and a hemispherical beam. Figure 3.1 shows this relationship over the range of distances considered for the tracking controller application.

$$\Phi = \frac{\pi r^2}{50} \tag{3.3}$$

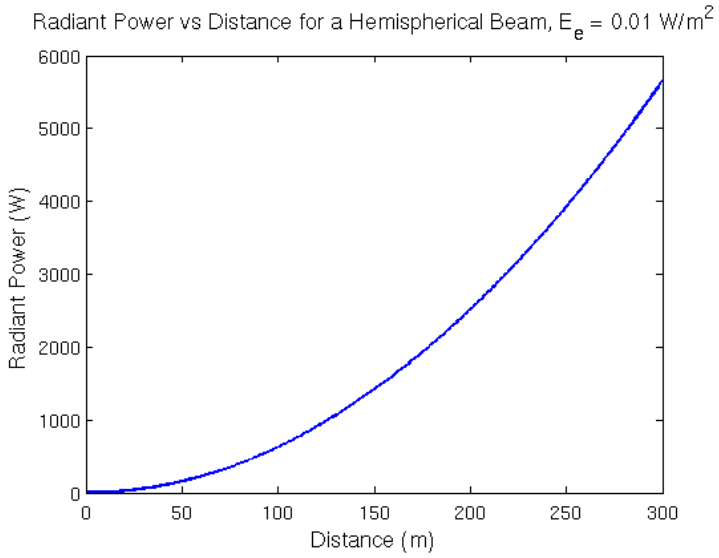

Figure 3.1: Radiant Power versus Distance for a Hemispherical Beam, E_e = 0.01 W/m^2

Figure 3.1 is somewhat misleading because only the radiant power in the spectrum between 868 nm and 1500 nm is passed by the optical filter discussed in Chapter 2. Any beacon chosen must output radiant power in this spectrum adequate to satisfy Equation 3.3 for the desired distance from aircraft to target.

3.2 Design

Rather than design for a particular distance, the highest-output available infrared source was selected for the beacon: a 500 W quartz tungsten halogen incandescent lamp. Determining the radiant power in the detectable spectrum requires a spectral

power distribution curve that is not normally provided by the manufacturer. However, a spectral power distribution curve can be found by modeling the lamp as a black body radiation source at the lamp color temperature of 3200 K. The spectral radiance curve for the selected lamp, a GE Lighting 23735–FDF-Q500T3/CL, is shown in Figure 3.2.

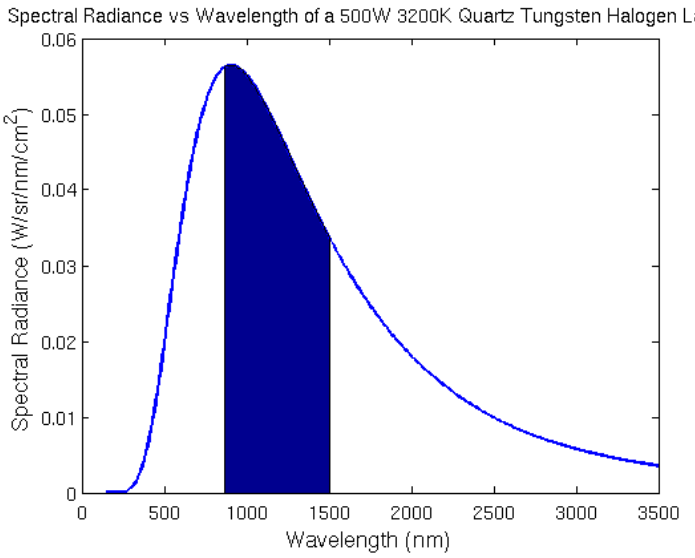

Figure 3.2: Spectral Radiance of a GE Lighting 23735–FDF-Q500T3/CL[16][17]

The shaded region in Figure 3.2 corresponds to the spectrum passed by the sensor infrared filter and is 41.6% of the total beacon output. The useful output power is 208 W and is detectable at a range of 58 m. GE does not provide error bounds for color temperature or radiant power but ±10% is typical for other lighting products.

3.3 Construction

The beacon is constructed from a cannibalized portable halogen fixture. A custom-machined aluminum mount provides attachments for the bulb holder, power cord fuse holder, aluminum reflector, and a standard 1/4-20 NC tripod mount tapped hole. The reflector is made from a 12-inch disposable aluminum pie pan. In use, the

beacon is mounted to a level photographic tripod and powered by a 3 kW generator. Figure 3.3 shows the beacon in detail and Figure 3.4 shows the beacon in operation.

Figure 3.3: Beacon Head

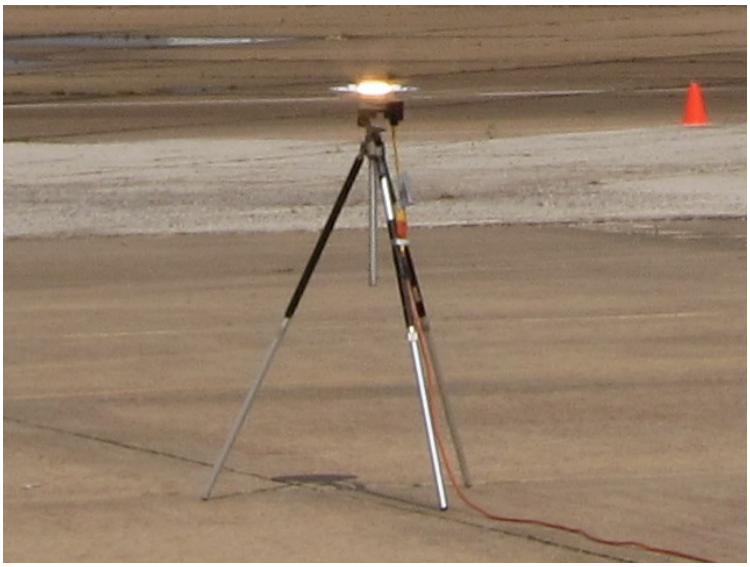

Figure 3.4: Target Beacon in Operation

4. FLIGHT TEST

This chapter discusses the flight text experiment to verify sensor performance. Since it is impossible to directly obtain the image seen by the infrared sensor as discussed in Chapter 2, the flight test verifies correct sensor operation by comparing the detected blob X-Y coordinates with video imagery taken from a small video camcorder mounted nearby the infrared sensor. The pilot flies the test aircraft in an orbit around the target beacon at an altitude and distance of his discretion in an attempt to keep the target in the sensor field of view. The experimental payload provides a real-time yes/no indication of whether or not the sensor is tracking a target to a ground station notebook computer.

As discussed in Chapter 1, this is a challenging task even when the pilot has visual feedback from the sensors. This research is intended as a proof-of-concept leading up to integration with a vehicle maneuvering controller and the success criteria reflects the difficulty of the pilot's task. The flight test success criteria are:

1. **Automatic detection and tracking**: The sensor must acquire the target beacon correctly during a video intra-coded image frame (I-frame) at least once.

2. **Relate target coordinates to bearing**: Error between the video image of the target beacon and the coordinates reported by the infrared sensor must be less than 3°.

4.1 Aircraft

The aircraft selected to carry the sensor experiment is a modified Multiplex Easy Glider Pro which includes a slightly more powerful powerplant than stock (Figure

4.1). This aircraft was selected because it features adequate payload capacity to carry the experiment, has lower personnel and infrastructure requirements than the other available aircraft in TAMU inventory, and was available at the time of testing. The endurance of the Easy Glider Pro in this configuration is roughly 45 minutes.

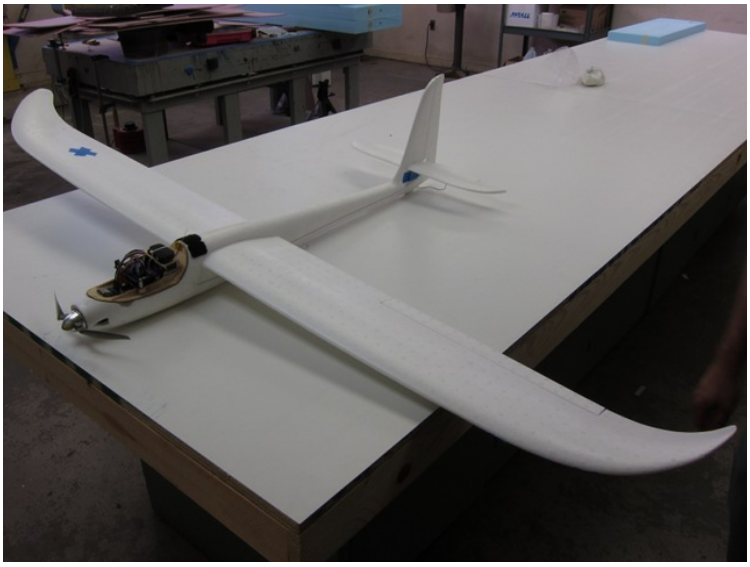

Figure 4.1: Multiplex Easy Glider Pro with Experimental Test Hardware Integrated

The autopilot selected for hosting the vehicle maneuvering controller is a DIY-Drones ArduPilot Mega (shown in Figure 4.2 with the experimental infrared sensor). ArduPilot Mega is based on the Arduino open-source software, open-source hardware electronics prototyping platform. Arduino is widely available and widely used by the electronics prototyping community because of its low price point and ease of use. One of the basic tenets of Arduino is software interoperability: source code written for the Arduino platform can be included in projects using any Arduino-derived hardware as long as object-oriented programming paradigms are observed.

ArduPilot Mega was chosen because it was the only generally-available autopilot at the time of selection with freely-available source code. However, ArduPilot

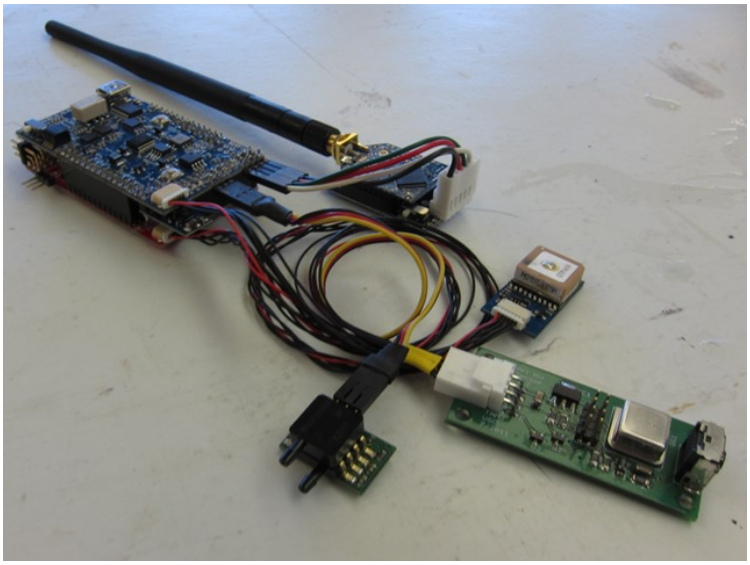

Figure 4.2: DIYDrones ArduPilot Mega 1.4 with XBee Radio, MediaTek GPS, Freescale Dynamic Pressure Sensor, and Wiimote IR Sensor on Custom Carrier Board

Mega has very limited space in EEPROM for logging flight data and relies largely on wireless telemetry for transmitting vehicle state information at up to 10 Hz for display and logging on the ground. For this experiment, the sensor data is collected at 30 Hz to match the camcorder frame rate. This rate is much faster than the 1 Hz rate required for the maneuvering controller and several integrated peripherals on ArduPilot Mega must be worked around to install additional log storage. Accordingly, though ArduPilot Mega is suitable for flight test of the maneuvering controller, a different platform is required for sensor verification and validation test flights.

4.1.1 Payload

The infrared sensor and camcorder are mounted on a variable-tilt common mount in over-and-under configuration looking out the left side of the aircraft. A micro servo actuates mount tilt so that different look-down angles can be selected on-line. This feature is included to allow tailoring the look-down angle as required for use later

with the vehicle maneuvering controller and has the side benefit of attenuating sensor vibration.

The experimental payload is composed of:

1. **Microcontroller**: ITead Studio ITeaduino 2.0 Arduino-compatible development board
2. **Storage**: Adafruit Industries MicroSD card breakout board+ with Patriot 8 GB MicroSDHC card
3. **Telemetry**: Digi XBee 2.4 GHz 802.15.4 wireless radio on a DIYDrones XtreamBee XBee adapter board
4. **Video Camcorder**: 808 Type 11 1280x720 camcorder
5. **Sound**: Generic piezoelectric loudspeaker
6. **Camera Mount Servo**: Hitec HS-55 sub-micro servo
7. **Power**: Liquidware Medium-Capacity Lithium Backpack

The experimental payload hardware architecture is shown in Figure 4.3 and software architecture is shown in Figure 4.4.

The payload is installed in a custom-made payload mount fitted to the Easy Glider Pro. The mount is made from balsa wood with a carbon fiber backing which helps shield the experiment from motor and electronic speed control EMI as well as shield the 72 MHz R/C radio receiver from payload EMI. The complete experimental payload package including mount is 193 mm long by 83 mm wide by 59 mm tall and weighs 160 g (Figure 4.5).

It is necessary to synchronize the infrared sensor data with the video imagery to correlate the target coordinates in both frames. However, the 808 camcorder does not

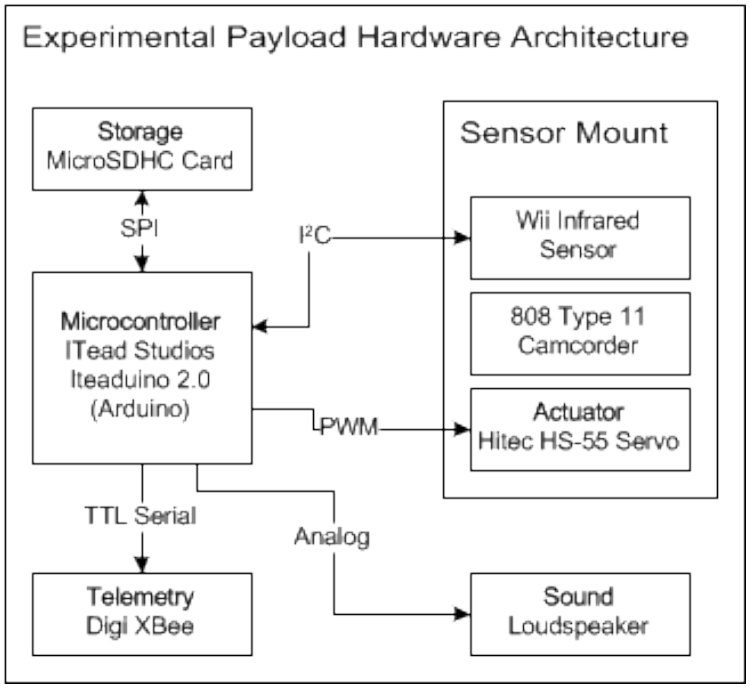

Figure 4.3: Experimental Payload Hardware Architecture

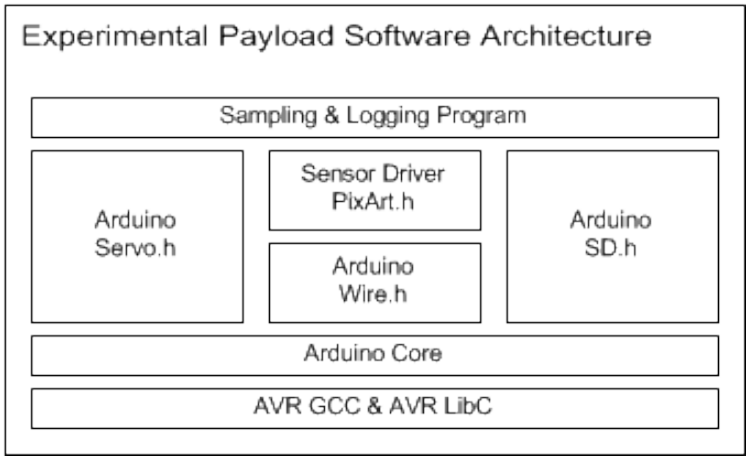

Figure 4.4: Experimental Payload Software Architecture

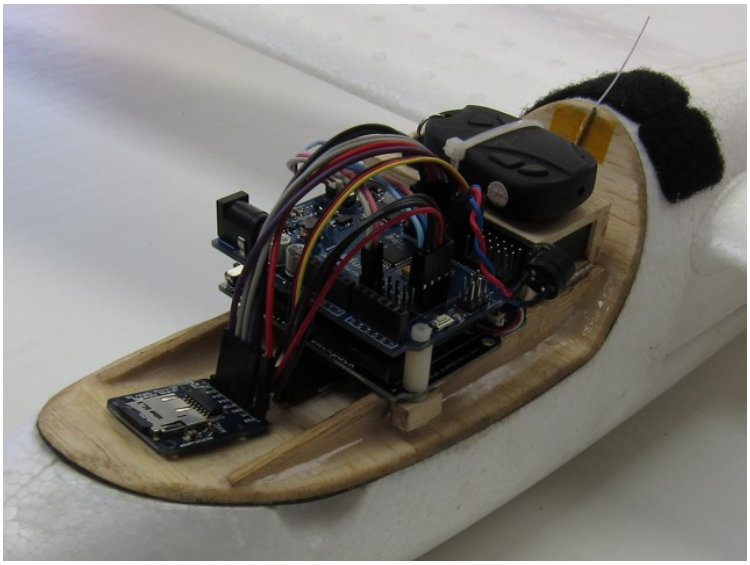

Figure 4.5: Experimental Payload Installed on Multiplex Easy Glider Pro Aircraft

support external triggering for frame synchronization. Instead, the data collection software generates an audible chirp as each sample is taken which can be identified in the camcorder audio stream. The corresponding video frames can be found because the camcorder audio and video streams are internally synchronized.

Upon blob detection, the sensor test program logs 15 samples containing a timestamp and X-Y coordinates of all active blobs at a rate of 30 Hz and reports a detection event over wireless telemetry. These 15-sample runs are 500 ms long to matches the 500 ms key frame interval and 30 Hz frame rate of the h.264 video codec used in the camcorder. Key frames or intra-coded image frames (I frames) are frames at which a fully-specified image is recorded. The remaining 14 frames in the sample interval are forward-predicted image frames (P frames) which are stored as a series of pixel changes from the previous I frame. Only I frames may be used for this analysis because is not possible to guarantee image accuracy during P frames.

A laptop PC running a serial terminal emulator is used to view detection events

transmitted over wireless telemetry.

4.2 Verification

The target coordinates in the sensor image frame are correlated with I frames from the video by matching the times at which synchronization tones in the camcorder audio stream occur with the corresponding timestamped coordinates in the sensor log. Detecting sample synchronization tones while airborne proved impossible due to wind and propeller noise. However, correlation on certain flights was possible because the payload was allowed to view the target beacon after system initialization but before takeoff to establish a time offset between the payload sample timer and the camcorder video timer. Both the payload and the camcorder use standard quartz crystal oscillators which experience clock jitter on the order of 10 parts per million on average. The maximum error bound in this case is 72 ms at the end of a 10 minute experiment run which is an error of roughly two image frames.

The angle of view and pixels per solid angle for both the infrared sensor and the camcorder are needed to correlate target locations between the fields of view of each sensor. Fortunately, although the infrared sensor and camcorder have differing resolution and angle of view, their combination of imager size, focal length, aperture size, and resolution are such that their pixel width matches to within experimental error bounds discussed in Chapter 2. Both devices share a pixel width of 141 arc-seconds/pixel. This trait allows the infrared sensor field of view to be overlaid on the camcorder field of view using a simple pixel coordinate offset without requiring a scaling factor.

The angle of view experiment discussed in Chapter 2 also determined that sensor frame origin is in the top-right corner. Conventionally, the image frame origin is in the top-left corner. It is important to note that the sensor frame coordinates presented

here have had their X value inverted so that comparison with the video imagery can be made. The offset between sensor frame and video frame was found by pointing the payload package at a beacon placed 5 m away and finding the coordinates of the beacon in both frames. The conversion from sensor frame to video frame is X + 122 pixels, Y + 27 pixels as-installed as shown in Figure 4.6. At 5 m, the parallax error due to the 1 cm distance in-plane between the infrared sensor and camcorder is 0.11°or 3 pixels and as such is considered negligible and not accounted for in the frame offset.

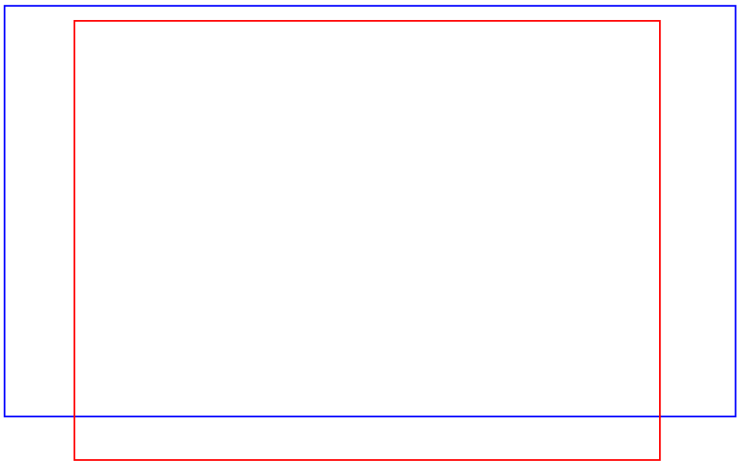

Figure 4.6: Relative Position of Fields of View (Infrared: Red, Camcorder: Blue)

4.3 Results

All testing occurred at Texas A&M Flight Test Station (ICAO code 83TX) in the center of runway 17R/35L. Table 4.1 shows the date, time, and duration of the test flights:

Flight #	Date (YYYY/MM/DD)	Time (hh:mm)	Duration (hh:mm)
1	2012/07/15	14:48 UTC	00:06
2	2012/07/15	16:05 UTC	00:16
3	2012/07/17	00:26 UTC	00:07
4	2012/07/17	00:39 UTC	00:03

Table 4.1: Flight Test Date, Time, and Duration

Data collection began at power-up prior to takeoff and continued until power-down after landing. Only tests 2 and 3 contained a discernible audible sample tone prior to takeoff. However, test 2 contained evidence of spurious target detection events prior to takeoff and as such only test 3 was examined. Test 4 terminated early due to inclement weather conditions. An instance in which two successive I-frames containing valid sample data was identified in test 3 and shown in Figures 4.7 and 4.8 below. The beacon location reported by the infrared sensor is highlighted in magenta.

Figure 4.7: Target Beacon Location vs. Detected Location, Sample 3915000

The coordinate data for these two samples is shown below in Table 4.2.

Figure 4.8: Target Beacon Location vs. Detected Location, Sample 3920000

	Sample 3915000	Sample 3920000
X coordinate, infrared	768	322
X coordinate, camcorder	770	343
X coordinate error	2	21
Y coordinate, infrared	220	328
Y coordinate, camcorder	230	324
Y coordinate error	10	4
Error, pixels	10	22
Error, degrees	0.4	0.9

Table 4.2: Target Coordinates and Error, Infrared Sensor vs. Camcorder

The sensor performs within expectations and easily achieves the 3° error success criteria. Interestingly, the error between the coordinates reported by the sensor and the location seen in the video does not maintain constant orientation: in the first sample the error occurs mostly on the Y axis while in the second sample the error occurs mostly on the X axis. This effect may be due to limitations in the proprietary subpixel sampling techniques in the sensor DSP since the transverse

speed is fairly high at 36°/s between these two samples and the error is on the order of the 0.3125°/px pixel width of the physical sensor. Error due to inaccuracy in the pixel offset between the infrared sensor frame and the video image frame would have appeared as a constant offset on both the X and Y axes.

5. CONCLUSIONS AND RECOMMENDATIONS

The sensor system developed in this research satisfies the requirements for the vehicle tracking and maneuvering controller application. Those requirements are given in Chapter 1 and are repeated below:

1. Automatic target detection.
2. Automatic target tracking with at least 1 Hz update rate.
3. Device driver class fits in ArduPilot Mega available free program memory with ArduPlane and experimental control law.
4. Known relationship between sensor frame coordinates and bearing from sensor installation.
5. Targets detectable at a practical orbit radius.

The automatic target detection and tracking requirements are satisfied by the intrinsic features of the Wiimote infrared sensor: up to four infrared signatures matching the criteria described by the sensor sensitivity settings are tracked at a rate of 100 Hz. The device driver class written to interface the sensor with Arduino easily fits in the available program space.

The resolution of the sensor image frame is 1024 px horizontal by 768 px vertical and the bench testing discussed in Chapter 2 establishes that the sensor field of view measures 40° horizontal by 30° vertical. The resulting pixel width is 141 arcseconds/pixel. The flight test results discussed in Chapter 4 show that the target X-Y coordinates are reported to within 1° of the position shown in video imagery. This represents a high degree of confidence that the location of the target as reported

by the sensor matches the actual position of the target in the real world. Spectral analysis of the sensor discussed in Chapter 2 established the spectrum within which targets are detected and Chapter 3 developed a relationship between beacon output power and maximum distance from beacon to sensor for detection.

Additional investigation into the effects of the sensor sensitivity parameters should be performed. The flight testing was performed using the default sensor sensitivity settings from the Wii, but it is possible that better range and target discrimination may be achieved by using different settings. Unfortunately, the full parameter space is 256^4 so investigating the full space is impractical. A reduced space encompassing the range of sensitivity settings the Wii uses might be more practical.

The maximum detectable target distance might be improved by flight testing at night. Position lights suitable for night flying are a straightforward modification to the aircraft. Pilot workload from visually tracking the aircraft under these conditions is not as high as might be expected because the aircraft will be under autonomous control. However, personnel lack of night vision from close proximity to the target beacon is a concern that must be addressed.

A higher-output target beacon would also increase range, but additional measures are required to ensure personnel safety. The lamp will need to be enclosed in a fixture which can contain fragments from a bulb explosion. Personnel must be trained to avoid staring directly at the target beacon to prevent permanent vision loss.

REFERENCES

[1] Peck, M., "Man Out of the Loop," *DefenseNews*, Nov. 2010.

[2] Wilson, J., "UAVs and the Human Factor," *Aerospace America*, July 2002.

[3] Valasek, J., Kirkpatrick, K., and May, J., "Intelligent Motion Video Guidance for Unmanned Air System Ground Target Surveillance," *Infotech@Aerospace 2012*, AIAA, Garden Grove, CA, 2012, AIAA 2012-2587.

[4] Kumar, R., Sawhney, H., Samarasekera, S., Hsu, S., Tao, H., Guo, Y., Hanna, K., Pope, A., Wildes, R., Hirvonen, D., Hansen, M., and Burt, P., "Aerial Video Surveillance and Exploitation," *Proceedings of the IEEE*, Vol. 89, No. 10, 2001, pp. 1518–1539.

[5] Egbert, J. and Beard, R., "Low-Altitude Road Following Using Strap-Down Cameras on Miniature Air Vehicles," *Mechatronics*, Vol. 21, No. 5, 2010, pp. 831–843.

[6] Yue, Z., Guarino, D., and Chellappa, R., "Moving Object Verification in Airborne Video Sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 1, 2009, pp. 77–89.

[7] Comaniciu, D., Ramesh, V., and Meer, P., "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, 2003, pp. 564–577.

[8] Lipton, A., Fujiyoshi, H., and Patil, R., "Moving Target Classification and Tracking from Real-Time Video," *Proceedings, IEEE Workshop on Applications of Computer Vision*, IEEE, 1998, pp. 8–14.

[9] Lee, J., "Hacking the Nintendo Wii Remote," *Pervasive Computing*, Vol. 7, No. 3, 2008, pp. 39–45.

[10] Plunkett, L., "WiiTV.jpg," Kotaku, March 2008, http://cache.kotaku.com/assets/images/9/2008/03/wiitv.jpg.

[11] Lee, J., "Interaction Techniques Using the Wii Remote," Presentation, UX Week, Aug. 2008.

[12] "Arduino," Web site, Oct. 2012, http://arduino.cc/.

[13] "DIYDrones ArduPilot Mega Official Repository," Web site, Oct. 2012, http://code.google.com/p/ardupilot-mega/.

[14] Hobley, S., "PixArt Wiimote Sensor Library for Arduino," Blog post, March 2009, http://www.stephenhobley.com/blog/2009/03/01/pixartwiimote-sensor-library-for-arduino/.

[15] Yoshizawa, T., *Handbook of Optical Metrology: Principles and Applications*, CRC Press, 2009.

[16] O'Haver, T., "Color Temperature of a Blackbody Source," Software program, University of Maryland, 2000, http://terpconnect.umd.edu/ toh/models/Blackbody.html.

[17] Forsythe, W. and Worthing, A., "The Properties of Tungsten and the Characteristics of Tungsten Lamps," *Astrophysics Journal*, Vol. 61, April 1925, pp. 146–185.