OughtToPilot

Project Report of Submission PC128 to 2008 Propeller Design Contest

Jason Edelberg

Table of Contents

Project Number

PC128

Project Description: OughtToPilot

A Propeller-based attitude estimation device embeddable into low cost remote control (RC) airplanes for hobby-grade UAV application is presented. The system senses and controls the aircraft's attitude (pitch and roll) to maintain level flight and guides the aircraft toward a pre-loaded GPS waypoint. An onboard Propeller-controlled digital camera images the flight in still frame or video mode. Flight data is continuously logged to a Secure Digital (SD) memory card.

The OughtToPilot employs a non-linear fuzzy-logic algorithm to blend two different types of inertial sensors (accelerometers and gyros) in a computationally efficient manner. This model free blending and estimation method was chosen over Kalman filtering to reduce the design-side system modeling effort and the real-time computational burden. All computations are conducted upon integer based pseudo float numbers with most values corresponding to a fractional number with three digits after the decimal point, multiplied by a thousand. (For example, the value 2008 in a variable named *pitch_angle* would correspond to an angle of 2.008 degrees.) It is believed that the inaccuracies of the low end (but cost effective) sensors far overshadow the inaccuracies introduced by the pseudo float computations.

The physical system (not including the airplane) is composed of the computer-board, a SD card reader, a GPS module, and a small digital camera. The computer board primarily houses the Propeller chip, its associated components (EEPROM, power conditioning, etc), connection ports, an analog to digital converter, and the inertial sensor. The SD cards stores the flight data which is continuously collected (regardless of mode). The recorded values are stored in a binary format so that they can be logged at a high data rate. The default parameters to be recorded are the raw sensor readings (gyros, accelerometers, and barometric pressure sensor), GPS readings, and internally computed values of interest to the programmer. The digital camera is a small and inexpensive 1.0 megapixel digital camera that has been modified to interface directly with the Propeller chip.

Concept of Operation: The OughtToPilot is pre-programmed with the flight plan (waypoint) before takeoff. Once airborne (take off and landings are conducted manually), the mode control switch is toggled from manual to automatic mode. At this point the system assumes control over the throttle (motor speed) and control surfaces (elevator and rudder for this test platform) and enters Automatic mode, where it regulates itself to fly straight and level. Pre-programmed turns can be ordered by pushing the transmitter control stick fully left or fully right for at least 1/10 of a second. Once ordered, the OughtToPilot executes a 3 second turn routine. If the operator moves the stick fully up or fully down for 1/10 of a second, the system switches over to Navigation mode where the system guides the plane toward the waypoint . (Further refinement of turns – steep vs. gentle - need to be implemented for more reliable navigation performance. Currently the system will reach the desired waypoint in about one out of five tries.) Attitude control and navigation continue to operate until the mode switch is toggled back to manual mode. Once the plane has landed and has been recovered, the SD data cards are pulled from the camera and SD card reader for download and analysis. This simple but straightforward concept of operations allows for reasonable functionality, while keeping the system simple enough to use the stock transmitter and receiver.

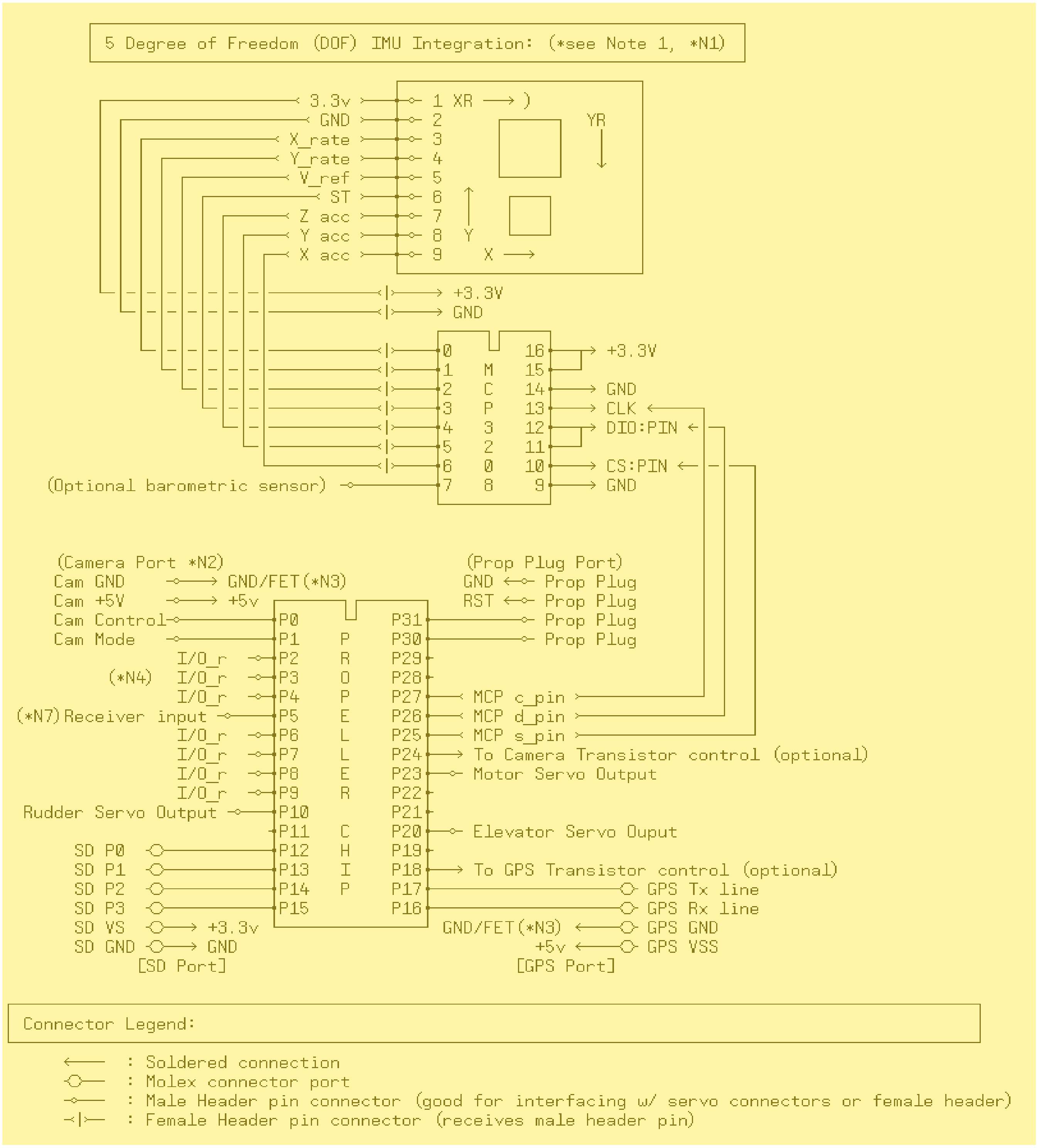
This project is made possible only by the Propeller chip – with so many control, estimation, and communication loops running all at once (each at different frequency) the only feasible approach is that of a multi-core parallel architecture. The power of the Propeller chip allows this project to fit in the size, weight, power, and cost requirements of a small introductory level RC airplane.

Parallel Operations

1. Main loop – management and Navigation 2. Receiver signal interpretation 3. Servo output commands 4. GPS I/O	5. SD card I/O 6. Analog data acquisition 7. Attitude Estimation 8. Flight Control

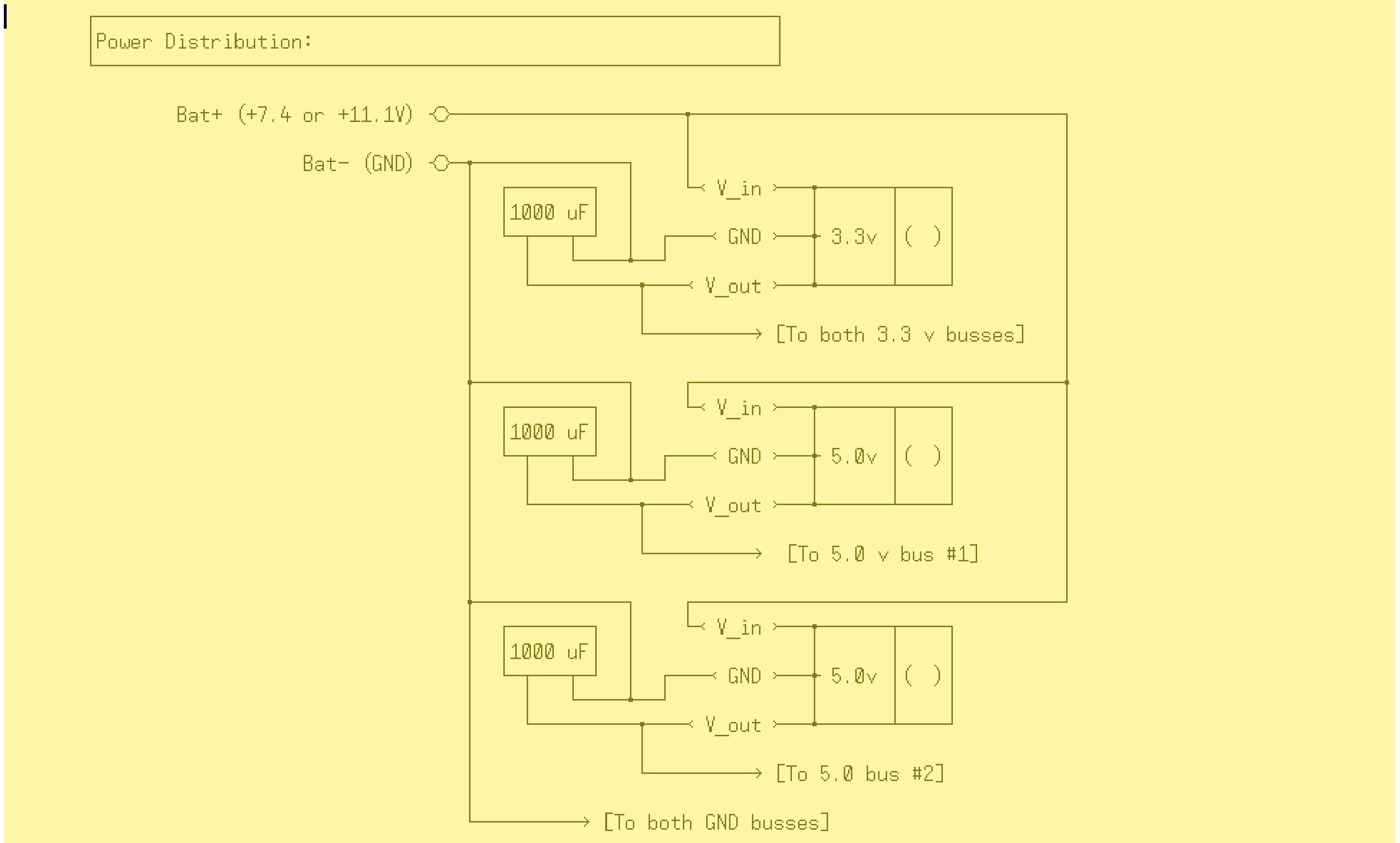
Schematic

Schematic A) Core architecture



Schematic B) Power distribution

```
Schematics:  The basic Propeller chip setup circuit (EEPROM/ oscillator, power and GND)
              is omitted from this summary diagram.
```



```
Two 5.0v regulators are used as a precaution due to the high loads servos can draw for short periods.
The rudder servo and the elevator servo are split, one on each of the two 5.0 v busses.
```

Schematic

Schematic C) Voltage layout and notes

```
 Notes:   *N1 - *N7

 Note 1: the only IMU chip inputs in this configuration are the +3.3V and GND.  The Vref and
    ST (self test) are left floating (inputs to the MCP3208 A/D converter).  The unit was operated
    both in this configuration and with Vref set to 3.3v, no performance difference was noted
    from the standard configuration to this space-saving configuration.   (The IMU - with header
    pins soldered to the bottom - simply plugs into the female pin header connected the A/D
    converter(MCP3208).

    (If more A/D channels are needed, the IMU pins from Vref and ST can be clipped off,
    opening up the inputs 2 and 3 on the MCP3208.)

 Note 2: GND/FET indicates that an optional MOSFET circuit can be installed so that
     the propeller chip can turn on or off high power electronics to conserve power.  This
     is useful for the GPS which can draw 50 to 100 mA and the camera which draws around
     80 - 100 mA.

 Note 3: As currently indicated the camera connector is of the pin style, molex style might be better.

 Note 4: I/O_r are propeller ports that are ready to interface with servo connectors (read or write)
         See notes 5 and 6.

 Note 5: Note that the board (BR1) design has 3 voltage busses running along the side.
         They are configured such that the outermost bus is +3.3v, the middle bus is
         GND, and the inside bus is +5v.  This allows for easy servo connection
         as the next note (Note 6) indicates.

   Voltage bus schematic:

         Board     +3.3v    GND     +5v
         Edge       bus     bus     bus
           │         ┼       ┼       ┼
           │         ┼       ┼       ┼
           │         ┼       ┼       ┼
           │         ┼       ┼       ┼
           │         ┼       ┼       ┼


 Note 6: The voltage bus description(Note 5) and images of motherboard.  Given
         the voltage buss configuration, the 3 pin servo connectors are soldered onto
         the breadboard such that the innermost pin electrically connects to the propeller
         I/O pin, the middle pins are all bussed to +5v, and the outer pins are all bussed to GND yielding
         the standard RC servo connector format as follows:
                                                 .
            GND      +5v       I/O               .
             ┼        ┼        -o────────┤P2    P    P29├
             ┼        ┼        -o────────┤P3    R    P28├
             ┼        ┼        -o────────┤P4    O    P27├
             ┼        ┼        -o────────┤P5    P    P26├
           outer    middle    inner              .
           pins      pins      pins              .

       Corresponding to RC servo wires:

           (black)    (red)     (white)   RC servo wires Futaba style
           (brown)    (red)     (yellow)  RC servo wires Hitec / JR style

         Also note that there is no resistor between the I/O and the propeller even though this is
         interfacing with a +5v device.  This configuration has been chosen due to the great amount
         of space and soldering-effort savings that it affords.  While the author has never
         experienced a problem with this setup, there is however some implicit risk in this configuration.

 Notes 7: With the stock reciever, all three channels and the mode (motor, rudder, elevator, and mode) are pulse
          modulated on one single output line which are intrepreted via RC_RCVR_OBJ.spin.  This conveniently
          saves wiring and therefore space which is helpful.  However, if the desired platform uses a more
          standard reciever, ample servo I/O pins have been setup on the motherboard so that each individual channel
          can be interpreted via RC_receiver.spin.
```

Bill of Materials

Description		Part #	Cost	Qty	Vendor	Total
Basline Project (not including airplane , airplane upgrades, consumables, or tools)						
Propeller Chip		P8X32A-D40	$12.99	1	Parallax	$12.99
EEPROM		602-00032	$1.95	1	Parallax	$1.95
MCP3208		MCP3208-CI/P-ND	$4.58	1	DigiKey	$4.58
Resonator Crystal (5Mhz)		251-05000	$1.10	1	Parallax	$1.10
3.3 V Regulator		601-00513	$1.95	1	Parallax	$1.95
5.0 V regulator		601-00506	$1.95	2	Parallax	$3.90
-Alternate 3.3V Switching Regulator		DE_SW033	$15.00	1	Dimension Engineering	$0.00
-Alternate 5.0V Switching Regulator		DE_SW050	$15.00	2	Dimension Engineering	$0.00
1000 uF Capacitor (very small)		493-1007-ND	$0.28	3	DigiKey	$0.84
Solderable Breadboard		BR1	$5.81	1	RP Electronics	$5.81
IMU 5 DOF		SEN-00741	$109.95	1	Spark Fun Electronics	$109.95
GPS Receiver		EM-406A	$59.95	1	Spark Fun Electronics	$59.95
GPS wires (for pigtails)		GPS-00574	$1.95	3	Spark Fun Electronics	$5.85
Digital Camera (PenCam SD 1.3)		R-PCSD13	$14.99	1	Aiptek	$14.99
SD card reader		SD Card Adapter	$12.99	1	uController.com	$12.99
SD cards (500Mb+)			$12.99	2	All over	$25.98
Optional MOSFETs (IRL520N)		IRL520NPBF-ND	$1.73	2	DigiKey	$3.46
Molex 2 pin male (to board) Power		WM4200-ND	0.21	1	DigiKey	$0.21
Molex 4 pin male (to board) GPS		WM4202-ND	0.38	1	DigiKey	$0.38
Molex 6 pin male (to board) SD Card		WM4204-ND	0.51	1	DigiKey	$0.51
Molex 2 pin female (to cable)		WM2000-ND	0.13	1	DigiKey	$0.13
Molex 4 pin female (to cable)		WM2002-ND	0.26	1	DigiKey	$0.26
Molex 6 pin female (to cable)		WM2004-ND	0.386	1	DigiKey	$0.39
Molex crimp pins (units of 10)		WM1114-ND	0.95	5	DigiKey	$4.75
Male header 16 pin		451-04001	0.99	6	Parallax	$5.94
Female header pin (cut to 9 hole)		PRT-00115	1.5	1	Sparkfun	$1.50
					subtotal:	$280.36
					w/ Extras:	$325.36
Airplane Suggestion (cheaper, but less stable than larger models)						
RC Plane kit - Supercub		HBZ7100	$159.99	1	HobbyZone	$159.99
3 wire servos		HBZ7242	$12.99	2	HobbyZone	$25.98
Brushless outrunner motor Park 450		EFLM1300	$69.99	1	HobbyZone	$69.99
ESC (w/ BEC) Electrifly Silver Series SS-25		GPMM1820	$39.99	1	HobbyZone	$39.99
					subtotal:	$295.95

Recommended tools: Soldering Iron, Molex crimper, servo pin crimper, drill, USB oscilloscope helpful but not necessary.

Required Comments and References

A small inexpensive RC airplane (HobbyZone Supercub) is used as the project's test platform with the hobbyist's budget in mind. It should be noted that while inexpensive, small foam airplanes are typically much less stable than larger, heavier, and more rigid RC airplanes. Therefore system integration and control will likely be more difficult than on a larger platform. It must be emphasized that all set-points, parameters, and flight control methods presented here are tuned specifically to the author's hardware. Other hardware WILL differ and will require testing, tuning, and investigation of flight controls. The core attitude estimation code, however, should vary only minimally due to slight inertial sensor calibration and mounting differences.

The author assumes no liability for the material presented, nor is any there any implied warranty. Remote control airplanes and electronics can be dangerous.

Reference:

Hong, Sung, "Fuzzy logic based closed-loop strapdown attitude system for unmanned aerial vehicle (UAV)",
Sensors and Actuators, 2003, 109-118

Appendix A: Fuzzy Attitude Reference Computational Estimator (FARCE)

The heart of this project is its ability to determine its attitude (pitch and roll) from a low cost inertial sensor which has three accelerometers and two angular rate sensors (referred to as gyroscopes even though a true gyro has spinning mechanical parts). The OughtToPilot uses what is essentially a Proportional plus Integral (PI) controller to estimate the gyro bias and determine its attitude. Though the system estimates and corrects both pitch and roll, the following discussion will show only a single axis.

Before the algorithmic discussion begins, a refresher on measuring tilt from a gyro and an accelerometer may be in order:

Accelerometer: An accelerometer measures just that, acceleration. If an object is at rest or at a constant velocity (no forces acting upon it), the sensor will in fact measure the amount of gravity-induced acceleration acting on its measurement axis. The tilt angle is computed as a function of the amount of gravity measured on that axis. To summarize so far - if the sensor experiences low dynamic motion, then its attitude estimate is based solely on gravity and is a very good measurement.

If however the accelerometer does have forces acting on it such as it is accelerating faster, slowing down, or turning (angular acceleration) then it will read gravity plus these other accelerations which will introduce error into the angle calculation. The result is that only for relatively gentle movement, an accelerometer can provide a good estimate of tilt, but its short term readings are not as reliable. In other words, its high frequency response is unreliable, but its low frequency response is, in fact, very reliable.

Angular rate sensor: This sensor measures the rate of rotation it experiences. To compute the angular change over a period of time, one only needs to integrate the signal from this sensor. The first problem with this sensor is that we don't know what the initial angle at time zero is, we only know the change in angle that has been experienced since time since integration began. This problem is minor in comparison to the notorious bias issue: angular rate sensors typically have a bias that varies dramatically in an unpredictable way. This means that the integration process is repeatedly integrating an offset, a ghost value that does not indicate a real phenomenon - yielding angular measurements that don't make sense. While these problems seem daunting, there is a good side to gyros – their moment to moment estimate of the angular rate is very, very good. Once the bias is known, integrating the gyro measurements over a short period of time is a very accurate way to determine the angular change. In other words, its high frequency response is very reliable, but its low frequency response is poor – just the opposite of the accelerometer.

This is why both sensors are used in inertial measurement, they are complementary. The high frequency content of the gyro measurements can be fused with the low frequency content of the accelerometers measurements to yield a good estimate.

We begin OughtToPilot's algorithmic discussion with a baseline PI estimator as a reference:

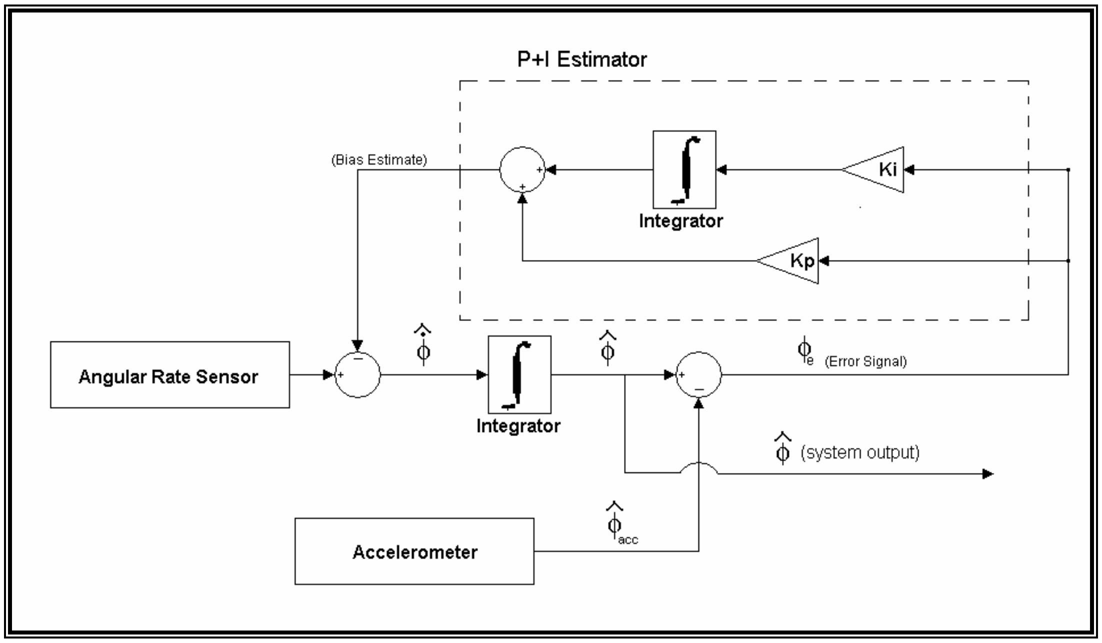


Figure 1, Baseline PI Estimator

Figure 1 shows a block diagram of a basic PI estimator configured for gyro bias estimation. Here is how it works:

1. The gyro (angular rate sensor) yields a reading of the angular rate, how fast that axis is spinning. It is integrated producing an angular measurement.
2. An angular error signal is generated by subtracting the accelerometer's attitude estimate from the gyro's attitude estimate.
3. The bias estimate then adjusted based directly upon the error signal (proportional action) and the integration (accumulation) of the error signal (integral action).

Assuming that Kp and Ki are tuned properly, this could yield a reasonable estimate of the gyro's bias. Once the bias is known, it is subtracted from the raw gyro reading so that the integration step produces the most accurate angle measurement possible. The problem with the method above is that the bias wanders in a completely unknowable and stochastic way, making appropriate selections of Kp and Ki apriori very difficult.

<u>Here is where the fuzzy logic becomes useful</u>: Based upon system knowledge, the values of Kp and Ki will be adjusted in real time to make the best use of each type of sensor. The rules are: if the system is experiencing low dynamic motion, then the system can rely more on the accelerometer readings. Therefore the values of Kp and Ki should increase as most of the error signal is due to the gyro's bias. Otherwise if the system is experiencing severe dynamic motion, then the accelerometer's measurements should be taken less seriously. In that case Ki and Kp are reduced owing to the fact that the error signal is heavily affected by inaccurate accelerometer readings and does not convey much information about the gyro's actual bias.

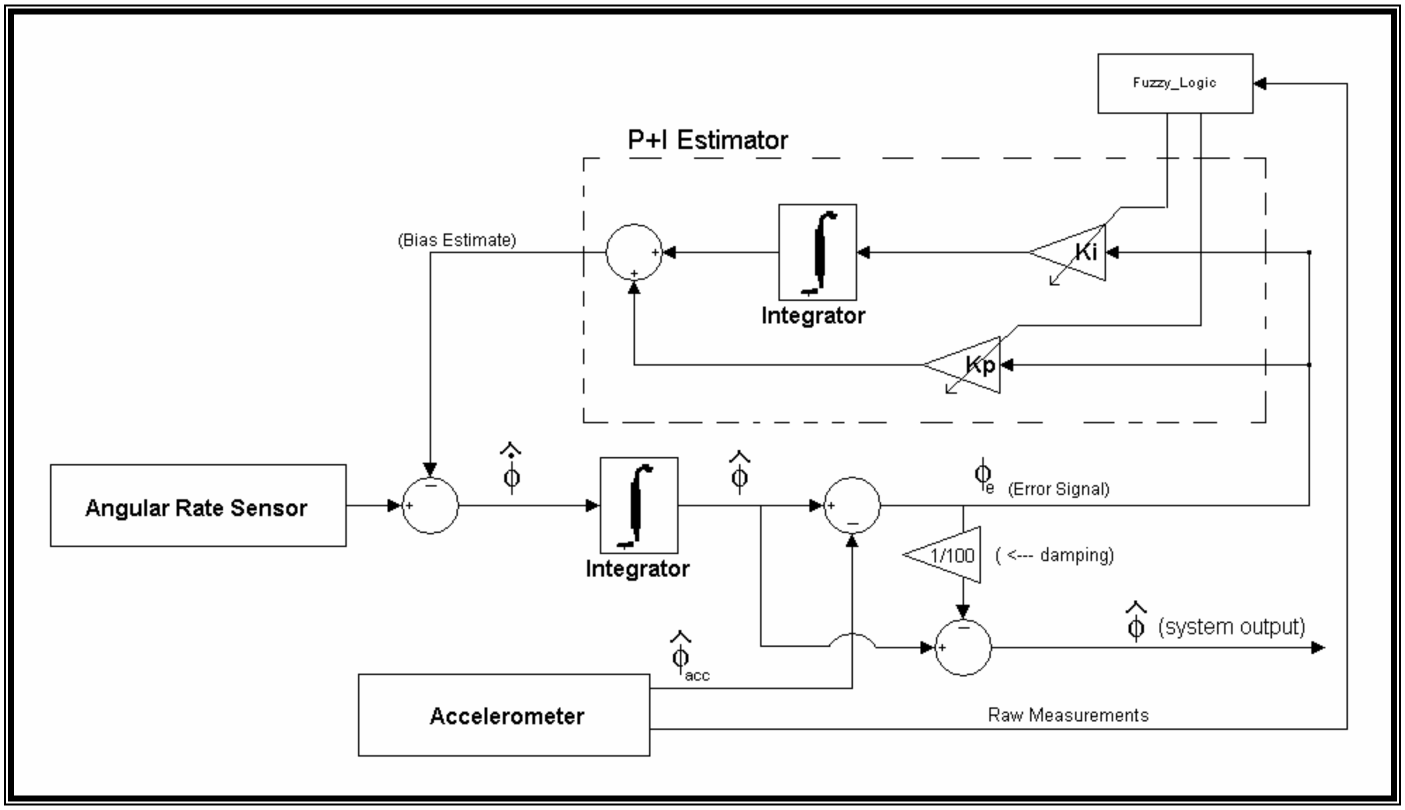


Figure 2, OughtToPilot Fuzzy PI Estimator

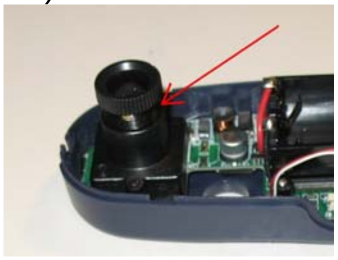
The metric for dynamic motion is the total vector of acceleration in all three axes minus one G. If the system is in low dynamic motion, then the total acceleration vector should equal one G, subtracting one G would then yield a value of zero indicating no dynamic motion. The tuning of the parameters goes something like this – if the dynamic motion is low then the parameters are increased by a factor of 100, if the motion is medium, then the parameters are increased by a factor of 10, otherwise the motion is large and the factor should be 1. This logarithmic scale has shown good results in the author's testing experience.

Also note the damping factor (1/100 indicated in the drawing). This is an ad-hoc adjustment that helps overcome the extreme jitter and high frequency noise experienced in small electric airframes. The value of 1/100 may be a little high; currently in OughtToPilot it varies between 1/100 to 1/400 based upon the dynamic motion metric.

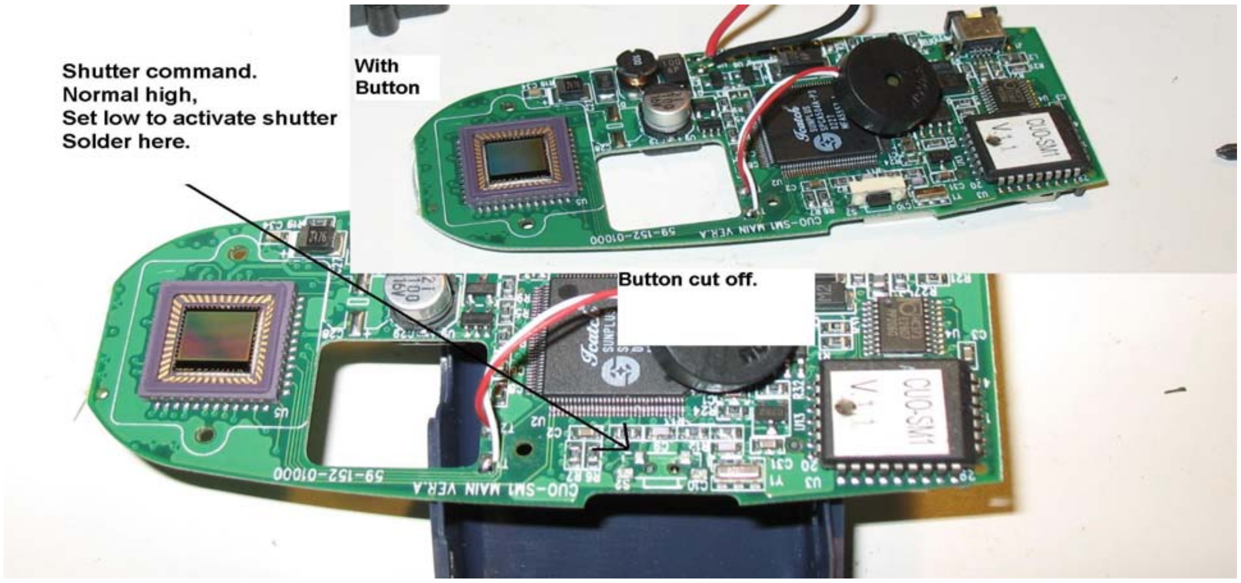
Appendix B: Camera Modification for Use with Propeller Chip

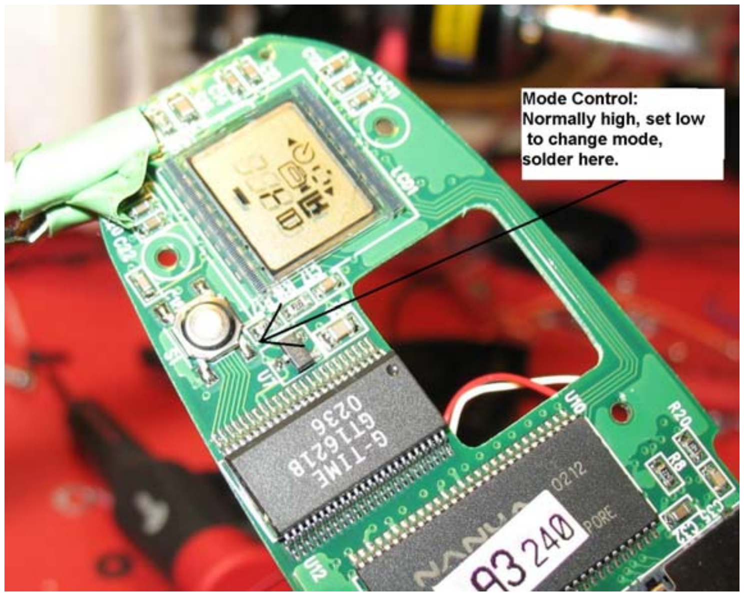
Camera: Aiptek PenCam 1.3M SD, Cost $14.99

1. Set the camera focus to infinity (mountain, not flower). Open the camera by removing the two screws on the side and gently pulling apart.
2. Dab superglue into threads of the lens piece so it is permanently fixed. Remove screws on side of the lens to remove the camera card from the housing (do not touch the surface of the sensor under the lens).



3. Solder a wire to (A) the shutter command solder point and (B) the mode command solder point.
4. Make a connector to: connect the red battery connector wire to 3.3 or 5.0 v, the black battery connector wire to GND, and the two control wires to output pins on the Propeller chip.




Mode Control:
Normally high, set low
to change mode,
solder here.
G-TIME
GT16218
0236