4 Navigation System

4.1 *Overview*

This chapter describes the computer algorithms that were used in the navigation computer developed for the Glidersonde. This chapter also describes the computer interfaces with the Global Positioning System (GPS) receiver, control servos, and the human operator.

4.2 *Concept*

The navigation computer is the brains of the Glindersonde system. The microcontoller, which is a microprocessor with built-in memory and communications interfaces, performs all the computations and serves as the master controller for the vehicle components. In order to accomplish these tasks, several computer interfaces are needed: (1) the GPS interface, (2) the human interface, and (3) the control interfaces. The human operator determines at what altitude the vehicle will terminate its ascent. The GPS interface feeds the navigation computer the data needed to perform the required manipulations. The control interfaces actually maneuver the vehicle, and terminates the ascent and descent phases.

The computer reads input data from the GPS receiver and automatically separates the aircraft from the balloon, computes the course to the recovery area, and navigates the aircraft back to the recovery area. This process is broken down into two phases: Ascent and Descent.

4.3 *Computer and Computer Interface Details*

4.3.1 Navigation Computer

The computer chosen was the Motorola MC68HC11, an offshoot (and upgrade) of the 6808 processor. This is an 8 bit processor with an 8 megahertz clock speed. An MC68HC11E9 EPROM (Electronically Programmable Read Only Memory) was used to store the navigation program. This form of storage is less prone to being over-written as on-board ROM, which is also of limited size on this particular chip, and therefore useless to store the entire program; all the ROM, except for 2 bytes, is taken by storing constants that are used in the program. The total program uses approximately

8 kilobytes from a total of 12 on the chip. This processor was the standard for Electrical Engineering students throughout the country, and is therefore easily obtainable and has a wide support network. This network allowed several canned math routines to be used, speeding the programming effort.

4.3.2 Computational Format

The intricate computations performed by the processor often exceeded the maximum integer size that the 8 bit process can process directly, which is a number larger than 256. A set of floating point math routines was obtained that allowed for larger numbers by converting the ASCII characters to a floating point format and adding an increment of 128 to the exponent. This gave a range of exponents from 0 to 255, which allows for large numbers.

4.3.3 Computer/GPS Interface

The computer communicates with the GPS through the National Marine Electronics Association (NMEA) protocol, using RS-232 communications ports. The GPS sentences were read in at 4800 baud. The time required to read in the input sentence was computed as 0.074 seconds. The input sentence contains the latitude, longitude, and altitude of the receiver.

4.3.4 Computer/Human Interface

The computer interfaces with the human operator using a laptop computer. The flight computer algorithms stays in a loop until the descent is terminated 150 feet above the home location. The operator breaks into the loop by entering a “&” from the laptop terminal. The algorithm detects the character, which is not a character that is part of the data stream, and responds with the menu portion of the program. The menu portion allows the operator to change the home location, change the drop altitude, change the neutral adjustment of the rudder servo, and change the balloon servo holding and release positions.

4.3.5 Servo Interfaces

The servo responsible for altering the direction of the aircraft is the rudder servo. This servo requires a constant holding torque, which was provided by an input signal consisting of pulses. The pulse width varies from one to two milliseconds, and a pulse is required every 20 milliseconds to maintain adequate torque from the servo gears. A 1 millisecond pulse corresponds to the extreme counterclockwise position of the servo gear head, and the 2 millisecond pulse corresponds to the extreme clockwise servo position. The

pulse rate, or frame rate is it is called, is accomplished by using an interrupt service routine that is built into the microprocessor. The interrupt routine was programmed to interrupt normal processor operation every 20 milliseconds, and to toggle one of the outputs to a high signal (5 volts) for a specified duration. The pulse duration is read from a memory location, which changed as the computer navigated the aircraft.

The second servo serves the dual roles of separating the Glidersonde Vehicle from the ascent carrier, and also terminates the descent phase via a parachute release. The two servo pulses are fed in a train, contained in the same interrupt service routine. The second servo pulse width is determined from a memory location, as is the rudder servo pulse width.

4.4 *Navigation Description*

The navigation program was broken down into two main parts: the ascent phase and the descent phase. The flight computer algorithms stays in an infinite loop until the descent is terminated 150 feet above the home location. The ascent phase contains two altitude checks, and the descent portion contains a single altitude check. Otherwise, the two portions are identical.

The computer is constantly attempting to navigate to the launch point (until the 150 feet point is reached during the descent phase). During the ascent portion, the balloon servo is set to a position to apply a holding torque to the release mechanism. From the release point on, the servo is set to release the balloon. The ascent portion is detected by the Ascent Bit. When this bit is set to “00”, the ascent checks are performed. During descent, this bit is cleared (“FF”).

4.4.1 Ascent

The ascent phase is initiated by interfacing with the laptop via the laptop interface. This is the only way to execute the ascent portion of the algorithm. When the menu portion of the program begins executing, the algorithm sets the Ascent Bit. This bit is used to determine which parts of the program to execute in the event of a power failure. When the operator types a “&” to enter the menu portion, the computer takes a new pressure reading for the home location in addition to setting the Ascent bit. This is to obtain updated pressure information.

Upon the next reset of powerup, the algorithm runs the ascent specific portions of the program, until the conditions are detected that are required for the initiation of the descent portion.

The ascent specific checks are the drop altitude check and the premature descent checks. The current altitude was compared with the separation altitude to determine if the separation altitude has been reached. The computer saves the last altitude input and compares the current altitude with the previous altitude to determine if the receiver is moving downward toward the earth, indicating an ascent vehicle failure. Once either an ascent failure condition is determined, or after the current altitude is greater than or equal to release altitude, the processor turns changes the balloon servo setting and clears the Ascent Bit. This action terminates the ascent portion of the algorithm and initiates the descent phase.

4.4.2 Descent Phase

The descent phase operates in a loop until the minimum altitude is attained (150 ft.). The microprocessor computes the required course the aircraft needs to achieve to return to the launch point. The processor also calculates the current heading of the vehicle. Based on these two angles, the processor calculates which way to turn the vehicle. The processor then calculates the servo deflection based on the deviation of the angles, and stores this deflection in the proper memory location for the interrupt service routine.

The initial phase of the descent begins immediately after separation from the ascent carrier. The first GPS sentences read after separation are saved as the initial separation point. This data point is the first of two data points required to determine the vehicle's heading. Although the code is operating in a loop, and this step is therefore not required, this gives an extra one second delay during which time the aircraft is pulling out of the dive. As soon as conditions are correct to terminate the ascent phase, the rudder is set to neutral, the Ascent bit is cleared, and the algorithm then reads in the initial point. This gives two seconds of flights during which the rudder servo is neutralized. This two seconds correspond to the pullup maneuver, during which no controls should be applied; as this might cause the aircraft to enter a spin.

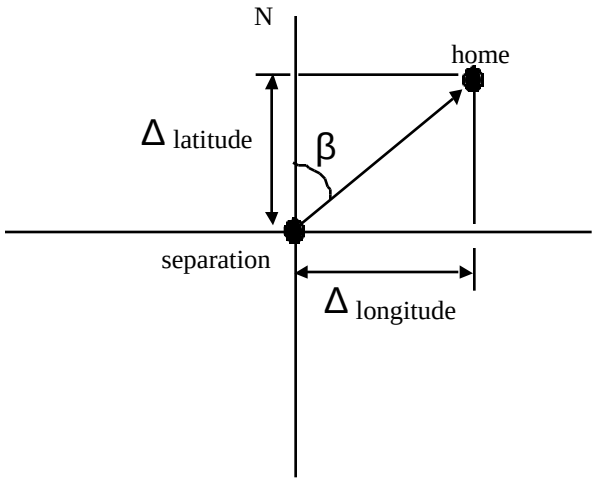

Figure 4-1 Coordinate System Used in the Calculations

Figure 4-1 shows illustrates the coordinate system that was used for computations. In this coordinate system, due North is 0º and positive angles are measured clockwise. The angle β is computed from:

$$\sin(\beta) = \frac{Lat_{separation} - Lat_{home}}{\sqrt{\left(Lat_{separation} - Lat_{home}\right)^2 + \left(Long_{separation} - Long_{home}\right)^2}} \qquad -90^o \le \beta \le 90^o$$

$$\beta = \beta + 90^o \qquad 0 \le \beta \le 180^o$$

The latitude and longitude sent by the GPS receiver and are read as a DDMM.MMMM format for the latitude and DDDMM.MMMM format for the longitude (Degree Degree Degree Minute Minute . Minute Minute Minute Minute). All positions are converted to minutes to ease the calculations.

The longitude in all calculations is corrected by:

$$\Delta\text{longitude} = \cos(\text{latitude}_{launch})\Delta\text{longitude}$$

This correction is necessary because the lines of longitude all meet at the poles of the Earth, so the lines compress toward both poles. The home latitude was chosen to correct all the data because the vehicle was not expected to fly more than approximately 40 minutes (~40 nautical miles) away from the launch point.

For β in quadrants three and four, the angle is normalized by subtracting β from 360º. These quadrants are determined by comparing the longitude at the launch point to the longitude at the separation point.

The numeric value of β is determined by using an iterative technique known as the Cordic routine. The infinite series representing the inverse sine function did not converge well on the correct result unless at least 12 terms in the series were used, and the additional programming effort was not deemed necessary. By simply re-arranging the equation format to avoid this function, math routines already obtained could be used. The sine function is also well behaved in the domain of interest, allowing the Cordic routine to always converge on the correct root. Converge criteria was an error of ±0.001 radians. Convergence was always within 5 iterations or less. With the value of β, the ground track to return to the launch site is known.

During the computation of β, the aircraft has separated from the ascent carrier and is allowed several seconds to stabilize before any active guidance takes place. Once the calculation of β is complete, the course over ground is computed. The algorithm saves the current and previous locations, and these locations are used to determine the vehicle heading. Figure 4.2 shows the geometry.

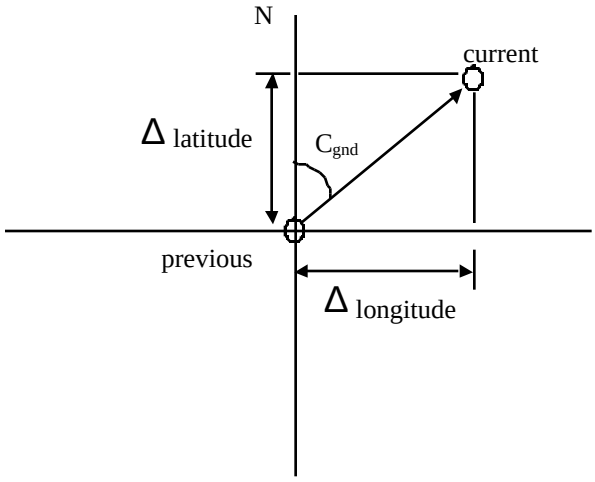

Figure 4-2 Calculation of Vehicle Heading

The course over ground is calculated according to:

$$\sin(C_{gnd}) = \frac{Lat_{current} - Lat_{previous}}{\sqrt{(Lat_{current} - Lat_{previous})^2 + (Long_{current} - Long_{previous})^2}} \qquad -90^o \le C_{gnd} \le 90^o$$

$$C_{gnd} = C_{gnd} + 90^o \qquad 0 \le C_{gnd} \le 180^o$$

With the course over ground and the heading home calculated, the Course Deviation is Calculated according to:

$$CDI = \left| C_{gnd} - \beta \right|$$

where C_{gnd} is the calculated instantaneous vehicle course over ground.

With CDI calculated, the pulse width (the pulse width determines the servo deflection) is computed from,

$$PW = Gain ? CDI$$

The pulse width is compared to a maximum pulse width. The maximum pulse width corresponds to the maximum allowed rudder deflection, so the aircraft will not turn too fast and enter a spin. The combination of gain setting and maximum deflection defines a "proportional cone". Inside the cone, the rudder deflection will be proportional to the course deviation; outside the cone, the deflection will be the maximum allowed.

After the proper pulse width is computed, it is either added or subtracted from the neutral pulse width value to obtain the required rudder deflection. This pulse width is then stored a memory location that is used by the interrupt service routine. When the next interrupt service routine is executed (20 ms max), the new pulse width will drive the servo.

4.5 *Logic Flowchart Description*

The first actions upon a reset or power up are the neutralization of rudder and the transfer of the critical flight variables. The home location that is stored in EEPROM is converted from ASCII characters to floating point numbers and then converted to proper units for calculations. These converted numbers are then stored in the proper RAM memory locations. The home altitude and drop altitude are also converted and stored in RAM. During this initial phase, the balloon servo is defaulted to the holding position. The algorithm then checks the Ascent Bit to set the balloon servo position.

After the servo check, the computer takes the first line of GPS data. The GPS sentence is checked to see if the data is valid. This is done through a transmitted bit in the data stream. If the data is not valid, the data is ignored. After the first line of data is read in, it is stored and another data point is taken. With the two points of data, the heading back to the launch point and the vehicle heading are computed. In ascent mode, the computer performs the drop altitude and premature drop checks. In descent mode, the 150 foot altitude check is performed.

After the altitude checks are complete, the course deviation is computed. The pulse width is then computed and stored in the proper memory location. The algorithm then copies the current position into the previous position and reads in the next GPS sentence. This process repeats until it is terminated by a menu command.

4.6 *Flight Simulation Software*

To assist in developing the flight navigation system and to understand the interactions of the aircraft and the control system, a Glidersonde flight simulation program was developed specifically for this project. The program is a windows application that communicates with the flight computer via the serial line that is normally interfaced to the GPS. The program simulates the GPS function and the directional flight characteristics of the glider. A screen capture of the program in a simulated flight is shown in Figure 4-4.

The simulation program allows the user to start a flight at any location in the world. The latitude, longitude, and initial heading can be entered and the aircraft speed and turn radius are user determined. In the open loop mode, the simulation program just sends one or two GPS sentences (GPRMC and/or GPGGA) every second. The program calculates a new position, based on the current position, current heading and speed, and current turn radius every second to send to the flight computer. In the closed loop mode, the simulation program looks for a response from the flight computer on the receive serial line. The response can be changed but now consists of two integers, one giving the aircraft altitude and the other giving the rudder position (actually the pulse width being sent to the rudder servo). The flight simulator then responds to the rudder position by implementing a turn proportional to the pulse width deviation from the neutral rudder pulse width.

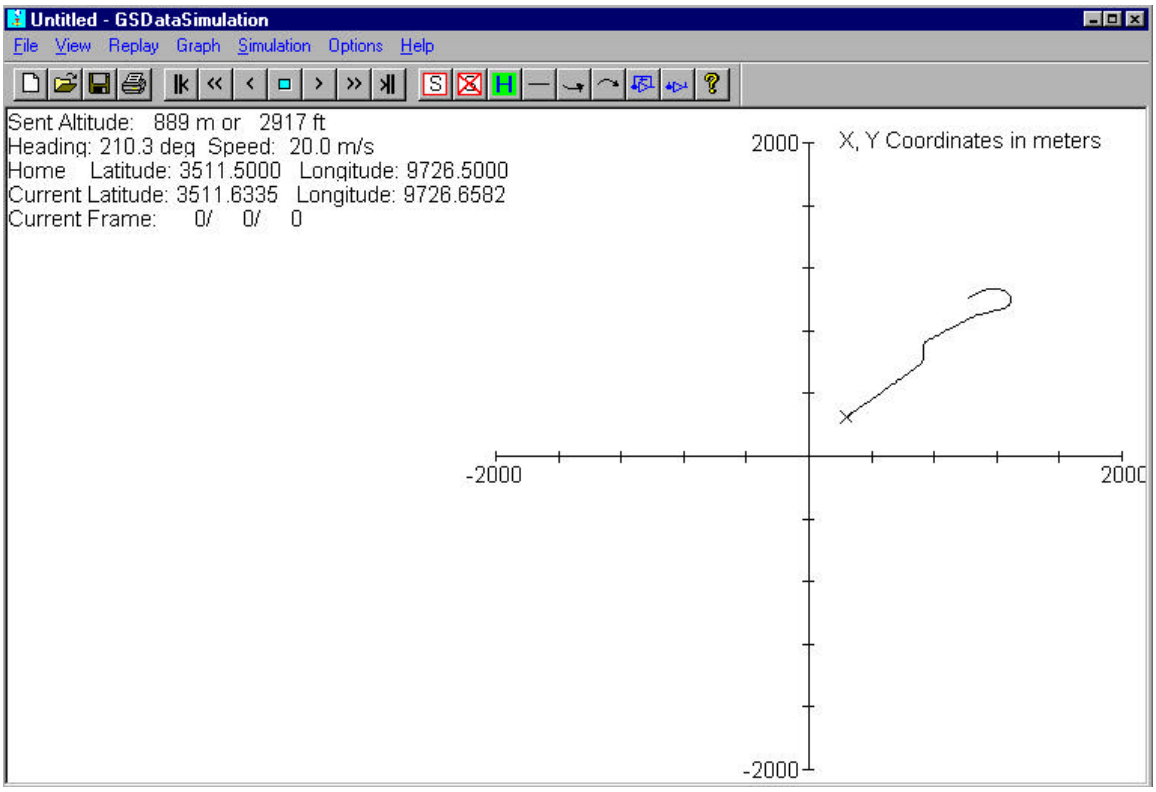

Figure 4-3 Simulation Program for Glidersonde Flights

Features of the program are:

1. Allows setting of the home location (the origin) anywhere in the world.
2. Home location can be sent to flight computer at any time.
3. Aircraft flight parameters speed, turn radius, initial flight position and heading, rudder zero and rudder max value can be set or changed in flight.
4. User can select GPS sentence to send
5. Particular flight paths can be forced: left or right turns, straight-line path, figure 8, etc..
6. Feedback from the flight computer can be enabled or disabled to study navigation algorithm.
7. GPS sentence GPGGA sends simulated altitude with user determined rate of change.
8. Flight computer feedback displayed on screen and used to determine turn radius in the closed loop mode.

The simulation program has proven very useful for debugging the flight navigation program without endangering an aircraft.