SUST Journal of Engineering and Computer Sciences (JECS), Vol. 16, No. 3, 2015

Implementation of Direction Control Algorithms for Fixed Wings UAV

Mohamed K. Abdalla[1], Aisha H. Abdullah[2], Tahani A. Attia[3] and Sharief F. Babikir[3]
[1]Faculty of Engineering, Neelain University, Sudan
[2]Faculty of Engineering, International Islamic University Malaysia, Malaysia,
[3]Electronics and Electrical Engineering, University of Khartoum, Sudan
banash07@gmail.com,

Received: 23/05/2014
Accepted: 02/07/2014

ABSTRACT- The demand for unmanned aircraft vehicles, UAV, has grown quite significantly in recent years. It is therefore important to implement efficient and accurate computer controlled navigation algorithms on such vehicles to guarantee the successful and reliable delivery of the vehicle to its final destination. This paper describes algorithms that use the Global Positioning system data to navigate from the current position to the destination point. The algorithm continuously adjusts the direction of flight thus ensuring the shortest path to the target. The algorithm has been successfully tested within Khartoum Province.

Keywords: *Fixed wing, longitude, latitude, UAV.*

المستخلص **– هنالك إزدياد مستمر فى الحوجة إلى المركبات الطائرة التى تعمل بدون طيار. عليه إزدادت أهمية خوارزميات التحكم الالى للملاحة التى تضمن الوصول الآمن للمركبة إلى نقطة الهدف النهائية. توصف هذه الورقة خوارزمية تستعمل نظام التحديد العالمى للقيام بالملاحة الى النقطة النهائية. و تعمل الخوارزمية على التعديل المستمر لاتجاه المركبة بهدف الوصول للنقطة النهائية. و تم إختبار النظام بنجاح داخل ولاية الخرطوم.**

I. INTRODUCTION

The main purpose of navigation is to be able to get from point A to point B as easy as possible. The word navigation (Latin navis, “boat”; agire, “guide”) traditionally meant the art or science of conducting ships and other watercraft from one place to another. Today, navigators guide craft on land and in the air as well as on the water (and underwater in submarines)[1].This paper presents an algorithm that uses the received GPS data to control the navigation of an UAV all the way from a given starting point to the destination point. The algorithm tries to avoid excessive steering of the vehicle. The algorithm will be implemented in two microcontrollers.

Position and direction: finding direction is a very old problem which solved by the early scientists which defines the directions as North, South, East and West. This can be easily implemented at days, because the east and west can be determined by the sun movement direction and then north and south can be determined. It was discovered that all magnetic materials have two polar, northern polar and southern polar. The real compass consists of a freely moving magnetic blank and a scaled circle with 360 full scale degrees.

Position with GPS: The GPS is made up of three parts: satellites orbiting the Earth; control and monitoring stations on Earth; and the GPS receivers owned by users. GPS satellites broadcast signals from space that are picked up and identified by GPS receivers. Each GPS receiver then provides three-dimensional location (latitude, longitude, and altitude) plus the time, [1].

True Course: the true course is the current heading of the aircraft (in degrees) and is measured clockwise from the North direction; it used because it saves an enormous amount of computation. The target heading is not given to us and must be calculated. The correct way of calculating the heading to a waypoint is to use the bearing formula, [2]:

$$c = \cos^{-1}\left[\frac{\sin(LAT2) - \cos(LAT1)\cos(d)}{\cos(LAT1)\sin(d)}\right] \quad (1)$$

where d is the great circle distance. Unfortunately, trig functions (i.e. sines, cosines, tangents and their inverses) tend to eat up a lot of microcontroller memory. To minimize the use of these functions, we are going to use the Pythagorean Theorem to calculate our waypoint heading. This is an approximation because it ignores the curvature of the earth [2].

Space Segment: The space segment (SS) is composed of the orbiting GPS satellites. The GPS design originally called for 24 Satellites, eight each in three circular orbital planes [3], but this was modified to six planes with four satellites each [4]. The orbital planes are centred on the Earth, not rotating with respect to the distant stars. The six planes have approximately 55° inclination (tilt relative to Earth's equator) and are separated by 60° right ascension of the ascending node (angle along the equator from a reference point to the orbit's intersection) [5].The orbits are arranged so that at least six satellites are always within line of sight from almost everywhere on Earth's surface [6]. The result of this objective is that the four satellites are not evenly spaced (90 degrees) apart within each orbit. In general terms, the angular difference between satellites in each orbit is 30, 105, 120, and 105 degrees apart which, of course, sum to 360 degrees [7].

Orbiting at an altitude of approximately 20,200 kilometers and orbital radius of approximately 26,600 [8], each satellite makes two complete orbits each sidereal day, repeating the same ground track each day [9]. Control segment: The control segment is composed of a master control station (MCS), an alternate master control station, four dedicated ground antennas and six dedicated monitor stations [10]. It monitors and controls the global GPS stations, and it uses automated computer systems to retrieve and analyze data from the receivers at those stations. GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly stable clock (often a crystal oscillator). They may also include a display for providing location and speed information to the user. A receiver is often described by its number of channels: this signifies how many satellites it can monitor simultaneously. Originally limited to four or five, this has progressively increased over the years so that; receivers typically have between 12 and 20 channels [10].

User Segment: GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly stable clock (often a crystal oscillator). They may also include a display for providing location and speed information to the user. A receiver is often described by its number of channels: this signifies how many satellites it can monitor simultaneously. Originally limited to four or five, this has progressively increased over the years so that; receivers typically have between 12 and 20 channels [10].

This paper presents an algorithm that uses the received GPS data to control the navigation of an UAV all the way from a given starting point to the destination point. The algorithm tries to avoid excessive steering of the vehicle. The algorithm will be implemented in two microcontrollers.

II. MODEL AND ALGORITHMS

Two microcontrollers were used to control the aircraft. One of them (positioning microcontroller) reads in the latitude, longitude and true course data from the GPS receiver and then calculates the current aircraft heading and the heading to the next target. Once the direction and amount of turn is known, this microcontroller communicates this information to the other microcontroller (controlling microcontroller) which is used to deflect the rudder’s servo accordingly.

The Positing microcontroller

Location Extraction: The National Marine Electronics Association (NMEA) has developed a standard for all GPS devices. The $GPRMC data string which is the Recommended Minimum Specific GPS/TRANSIT Data was used in this project. The $GPRMC data string contains 12 pieces of information separated by commas within the string:

$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70

The latitude is the third word, the longitude is the fourth word, and the true course is the eighth word. The following flow chart shows how to extract the latitude, longitude and the true course [1].

The true course is the current heading of the aircraft (in degrees) and is measured clockwise from the North direction. If the plane is heading east, for example, then the true course would be equal to 90 degrees. This single value saves an enormous amount of computation. Otherwise, a vector from the previous latitude/longitude point to the current latitude/longitude point would have to be drawn and then the angle it forms with the North direction would have to be calculated [2].

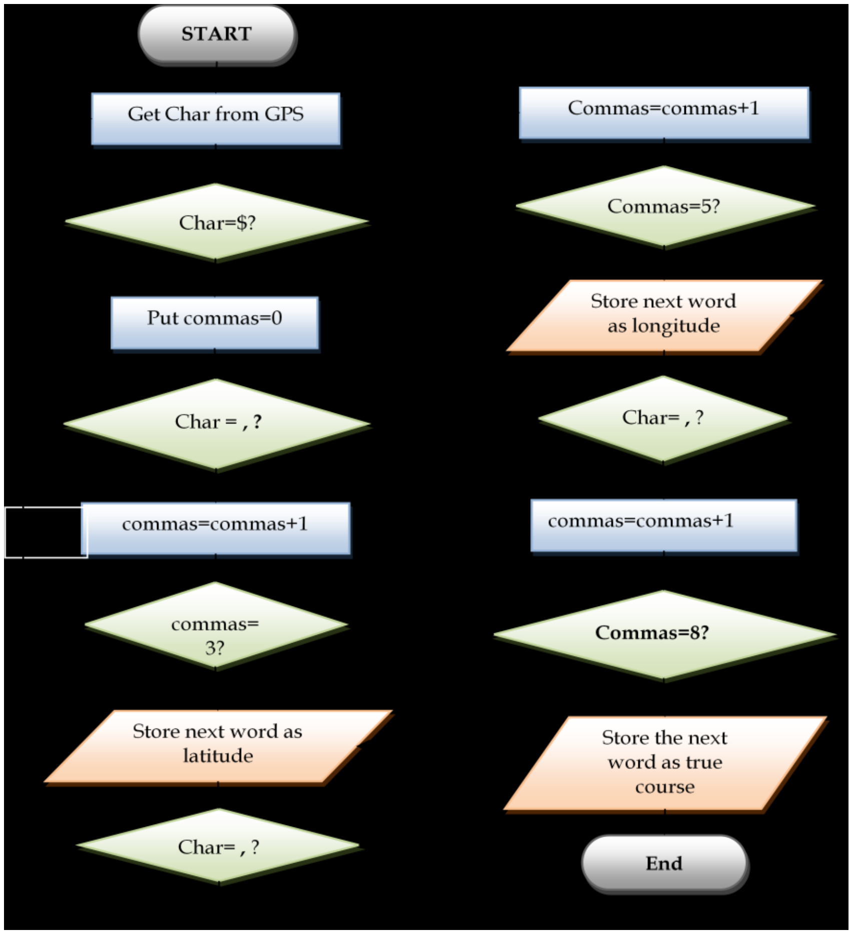

Figure 1: Extraction of data items from string received from the GPS module

Convert extracted data to degrees

The received latitude, longitude or true course is a decimal number e.g: 1537.3637 which mean 15 degrees and 37.3637 minutes. This must be converted to degrees as follow:

Since 1 degree = 60 minutes. So the latitude, longitude and true course can be converted to degrees using the formula:

$$Latitude = \frac{Latitude}{100} + \left[(latitude\%100) + \left(\frac{After\,decimal}{100000.0} \right) \right] / 60 \qquad (2)$$

Then it must be converted to radians by:

$$Latitude = Laitude \times \pi / 180 \qquad (3)$$

The same formula applied to the longitude and the true course.

Define reaching area: since the GPS receiver has some error (about 3 meters), so an area around the target must be defined, when reaching that area the target is considered to be reached.

If this area has radius R [2]:

$$1^{o} = 111Km$$

$$X^{o} = R$$

$$X = \left(\frac{R}{111Km} \right) \times \pi / 180 \qquad (4)$$

The algorithm to define that area using flow chart:

For All Targets

Repeat

diff Lat.= target Lat − current Lat

diff Long.= target Long. − current Long.

Until ((| diff Lat |< X) and (| diff Lat |< X))

Algorithm 2: Algorithm to minimize the jitter in navigation control

If the heading is to be determined using the inverse tan function with the N-S axis as the y-axis and the E-W axis as the x-axis, it would yield an angle about the East axis. However, we need the target heading about the North axis since the aircraft heading is given with respect to this axis. Rotating this coordinate system by 90 degrees clockwise will yield solution to this problem. The new coordinate system is shown in Figure 3.

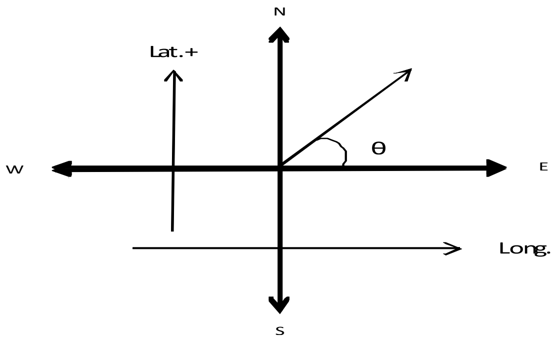

Figure 2: Coordinates of the earth

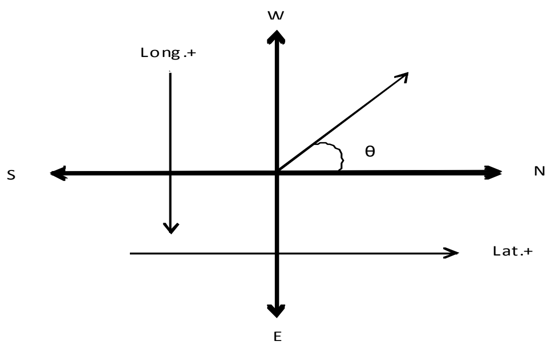

Figure 3: Coordinates after rotating by 90◦

Now, taking the inverse tan in this coordinate frame will yield the angle about the North axis. One last problem remains, however, as the angle yielded will take the counter-clockwise direction as positive. The algorithm in flow chart no.2 was used to change the positive direction to clockwise in order to coincide with the true heading value [2]. The coordinate should like Figure 4.

ComputeTarget Heading = atan2(diff Long−diff Lat.)

Restrict Target heading to be within the range -180..180

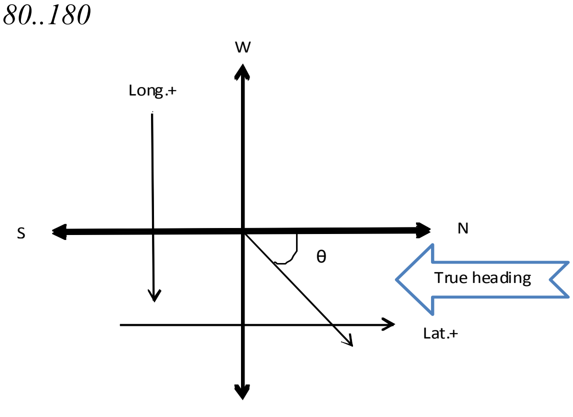

Figure 4: Coordinate of the earth after applying the algorithm

Now that the current heading of the airplane and the heading to the next waypoint are known, we can discern which way and how much the aircraft should turn in order to hit the next waypoint. For example, Figure 5 shows one scenario given the waypoint heading angle (measured from the North direction) and the true course angle (which is the current heading of the aircraft, also measured from the North direction). Intuitively, it is obvious the aircraft should turn right here to head towards the next waypoint [2].

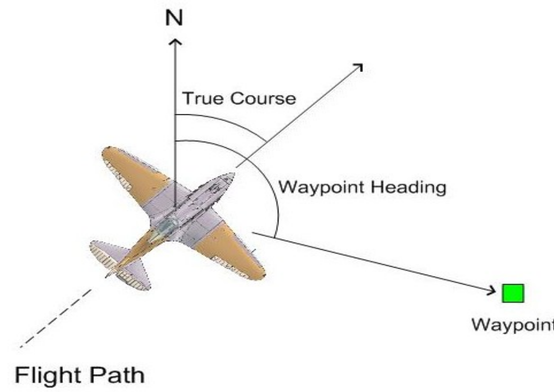

Figure 5: True course and waypoint heading (scenario1)

To code this algorithm we can say that if the aircraft heading is less than the waypoint heading, turn right. Otherwise turn left. However, this can be very inefficient in certain instances. For example, if the waypoint is just to the left of the North direction and the plane is heading a little right of North, the plane should turn slightly left to hit the waypoint. However, with the above algorithm the plane will get to the waypoint, but not before flying in a complete circle (well almost). This scenario is demonstrated in the Figure 6.

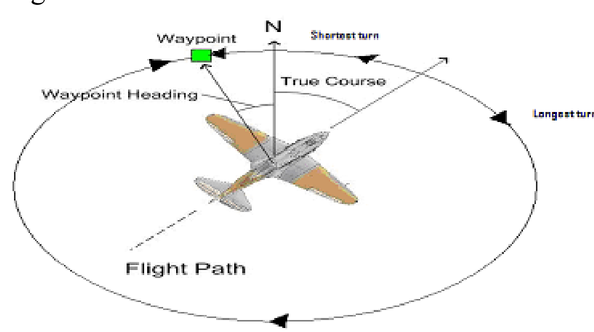

Figure (6): True course and waypoint heading (scenario 2).

i.e.: when Target Heading – True course >= 180 the vehicle will take the longest turn. The algorithm used in this project is based on proportional control of the error between the aircraft heading and the waypoint heading. So to force the plane to take the shortest turn to the target, a condition can be set:

If True course less than Target heading → turn right.

If True course greater than Target heading→ turn left

For the algorithm to take the shortest turn can be expressed in flow chart below [2].

Error in angle = Target Heading – True course

Update Error angle so that it is in range -180-180 degrees

The algorithm above was used to keep the error between 180 and -180 (to force the aircraft to take the shortest turn possible to the target. That is, it will not make a 300 degree right turn when it can make a 60 left turn to head toward the next target.

Duration of the duty cycle:

The whole turn of the servo is one byte (256)
256/2 = 127.5 (which is the neutral of the servo)

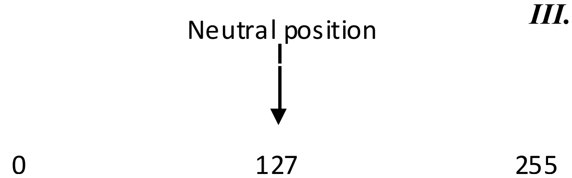

Figure 7: Neutral and full deflection in each direction values of the servo

Note that 0 is full deflection in one direction, 255 is full deflection in another direction, and the neutral position is 127. Therefore the duty cycle can be calculated using:

$$Duty\,cycle = rud_{pwm} = 1`27 - Error \tag{5}$$

Limits of Deflection:

Finally, the code of this microcontroller ends by limiting the deflection in each direction, because there are now ailerons in the plane used in this project. Deflecting the rudder fully in one direction will cause the aircraft to roll in addition to yawing. If it's deflected long enough, the plane will begin a downward spiral and the stabilization will be lost. So the plane needs a small amount of deflection in each direction [2].

This can be done by:

$$if\,(rud_{pwm} < 107) \longrightarrow$$

$$rud_{pwm} = 122 \;//\; turn\,right\,(neutral - 20)$$

$$elseif\,(rud_{pwm} < 147) \longrightarrow$$

$$rud_{pwm} = 162 \;//\; turn\,left\;(neutral + 20)$$

The Master Microcontroller:

The timer: the value of the timer of the PIC microcontroller can be set using the formula:

T=256-(period/(4*prescaler/(OSC_Freq))

The period of the timer, T, was set to (the interrupt occurs every) = 20ms as required by the servo motor

T = 256-(20ms/(4*8/(10MHz)) = 61

The timer computed above is used to generate and control the PWM signal:

Detection of Manual Override:

Since there may be SW interrupt triggers during HW interrupt, bogus pulse length value must be prevented by pressing the manual bottom for more than 1.5ms.

To convert 1.5ms to pulse length the following formula can be used:

$$Pulse\,width = 1.5ms \times \frac{clock}{4} = 3750 \tag{6}$$

III. *RESULTS AND DISCUSSIONS*

The algorithms described above were implemented on two microcontrollers that could communicate with each other over a serial interface. The hardware implementation is not described in this paper. The GPA module was used to feed the data to one microcontroller that decodes the message and extracts the data needed for computation purposes.

The tests performed with this research included the definition of a final destination target and following the directions provided by the algorithms. The tests were predominantly performed on the road where strict steering could not be followed. However, the instructions were always as expected to the last destination. It is concluded that the algorithms performed as expected and the steering commands could be relied on to navigate to the destination point. Several scenarios were tested and the performance of the navigation system was good.

REFERENCES

[1] Akash Deep Sharma, (2008) “Global Positioning System”, M.D. Publications Pvt. Ltd.

[2] URL:http://dasl.mem.drexel.edu , accessed Sept. 2014

[3] Daly, P., (1991), "Navstar GPS and GLONASS: global satellite navigation systems", Acta Astronautica, 25(7), Pages 399-406.
[4] Dana, Peter H. (1996). "GPS Orbital Planes"
[5] GPS Overview from the NAVSTAR Joint Program Office. Retrieved December 15, 2006.
[6] "USCG Navcen: GPS Frequently Asked Questions". Retrieved January 31, 2007.
[7] Thomassen, Keith. "How GPS Works". avionicswest.com. Retrieved 22 April 2014.
[8] Samama, Nel (2008). Global Positioning: Technologies and Performance. John Wiley & Sons. p. 65. ISBN 0-470-24190-X., Extract of page 65
[9] Agnew, D.C. and Larson, K.M. (2007). "Finding the repeat times of the GPS constellation". GPS Solutions (Springer) 11 (1): 71–76. doi:10.1007/s10291-006-0038-4
[10] http://en.wikipedia.org/wiki/Global_Positioning_System#cite_note-61, Retrieved July 31, 2014.
[11] http://airandspace.si.edu/exhibitions/gps/ Retrieved July 31, 2014.