

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

Kevin Shen, Olympia High School, Olympia, WA

Abstract

The most theoretically efficient airplane design is the flying wing due to its simplicity, resulting in low weight and low drag compared to more conventional configurations. However, a perfect flying wing is very unstable in pitch and yaw. The conventional solution is to sweep the wings back, creating a V-shape. This comes with a sacrifice of either maneuverability or efficiency, depending on the sweep angle and center of gravity of the flying wing. A wing with a high angle of sweep, resulting in a forward center of gravity will yield a highly maneuverable and stable, though inefficient aircraft, and vice versa. During different phases of flight, such as the difference between maneuvers and cruise, different amounts of sweep would be optimal. A design which could vary its sweep, thus aspect ratio, center of gravity, and trim, in flight, would be ideal. This project aimed to verify this concept by comparing the variable sweep flying wing with a sweep-optimized but fixed sweep flying wing. This was done by designing and flying a model of both the variable sweep design and fixed sweep design. A self designed flight computer was used to take signals from a remote control and determine how to actuate the various mechanisms on the flying wings. To determine maneuverability, a simple loop was timed to determine pitch rate, pitch being the axis which is affected by sweep. To determine efficiency, the declining battery voltage was used to measure power consumption in flight. Though the variable sweep design didn't outperform the fixed sweep design due to excess weight, this project was able to confirm the concept of varying wing sweep to increase efficiency was true.

Acronyms

VSWF- Variable Sweep Flying Wing
FSFW- Fixed Sweep Flying Wing
UAV- Unmanned Aerial Vehicle
CG- Center of Gravity
CP- Center of Pressure

AOA- Angle of Attack

AC- Aerodynamic Center

MAC- Mean Aerodynamic Chord

T/W- Thrust to Weight ratio

W/S- Weight to Surface (wing surface) ratio

Introduction

This research project looks into the possibility of the variable-sweep design to satisfy the sought-after characteristics of maneuverability and stability of a high sweep angle when needed while increasing efficiency and range. This can be utilized with missions like search and rescue where large areas of land or water need to be searched, requiring a high-efficiency aircraft, but also requiring accurate, stable, and maneuverable flight characteristics while circling or delivering a package to a recipient.

Like any engineering problem, there is no one solution that fits all. In most cases, conventional designs or even fixed sweep flying wing configurations would be all that is needed. If designed well, those can meet the maneuverability and efficiency requirements most of the time. However, when certain practical limitations don't allow for improvements in other areas, such as a wingspan constraint or a cost constraint limiting quality of materials, a variable sweep flying wing would be an attractive option.

The goal of this research project was to test the viability of a variable sweep flying wing in which the end result would be a proof of concept flying model that would be compared to an already fully optimized fixed wing flying wing. This would mean meeting a variety of smaller goals, such as:

1. Sizing, optimization, and construction of a fixed flying wing as a base design.
2. Design and construction of a variable sweep flying wing.
3. Developing software to control and manage a variable sweep flying wing by

interfacing pilot and automated commands to actuators.

Experimental Setup and Reasonings

Because the goal of this research project was to test the viability of a variable sweep flying wing, there must be some sort of comparison, and thus it was decided to design two identical flying wings, and modify one to be variable sweep capable. Since the variable sweep configuration should demonstrate that varying sweep to adjust effective aspect ratio, CG, CP, and trim can increase a flying wing's efficiency, the fixed sweep flying wing should already be fully optimized in terms of sweep, so no additional efficiency can be gained from simply placing the wings in a more efficient sweep angle, *with a set stability margin*. The hypothesis for this research project is that by sweeping the wings forward in flight, the stability margin and maneuverability can be sacrificed for additional efficiency through low sweep, high effective aspect ratio and less trim, and be regained when swept back. The experiment would involve designing, building, flying and testing both the Fixed Sweep Flying Wing (FSFW) and the Variable Sweep Flying Wing (VSFW) to compare their efficiencies.

However, to compare their efficiencies it would need to be shown that both aircraft could maneuver within a reasonable amount, because an aircraft that is able to be more efficient won't even be considered if it can't maneuver comparably with the alternative. A swept forward wing with its elevon control surfaces so close to the CG would result in a large decrease in pitch control, and because there isn't the yaw stability normally keeping the aircraft in line with oncoming airflow, turning and recovering from sideslips would be very difficult. Onboard sensors were considered for figuring out how fast the aircraft could pitch, but this was not used as it could be unreliable and there would be no way to tell if there were offsets or calibration issues. Therefore, a camera was used to time a simple loop, defined by a 360 degree

pitch up maneuver, which the pitch rate could be derived from. (Roll was not measured for two reasons; it wouldn't be affected by a different wing sweep angle, and most maneuvering on small flying wings is done by using roll to point the aircraft in the correct direction before pitch changes the aircraft's heading).

The efficiency was simply calculated by timing total flight time and dividing it by the difference in battery voltage. Lithium polymer batteries range from 3.3 to 4.2 volts per cell, which is dependent on the energy remaining in the battery. The more trials the better, but the important factor would be flight time.

Aircraft Sizing

Designing any sort of UAV requires initial sizing, from which the rest of the design can be based on. For the purpose of this research project, initial sizing would be adequate since the only purpose was to get a flying aircraft. Sizing always requires at least one known constraint, and in this case the constraints were the class of UAV and available power system. The class of UAV was important, since having something too large would be harder to make, test, and experiment on, as well as being exponentially more expensive. Being too small would change the Reynolds numbers enough where aerodynamic qualities would be too different from the intended real world use as a small scale UAV. The rough size determined the electronics and power system to be based around. Using a relatively standard quadcopter motor configuration, using information on the product packaging it was concluded to produce approximately 0.375 kilograms of thrust with the used configuration, as shown below:

- EMAX 1806 2280kv brushless motor: A quadcopter motor which runs at 2280 rpm for each volt applied.
- 6030 propeller: a 6 inch propeller which most optimally rotates 3 times per 6 inches of forward travel.

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

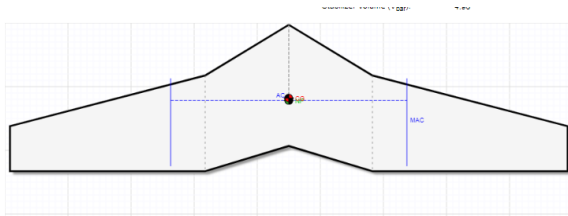
- EMAX 12A ESC: An electronic speed controller rated up to 12 amps.
- 650 milliamp hour (mAh), 2 cell, 7.4 volt lithium polymer battery.

The rest of the sizing process was based on historically working values for RC aircraft of this scale, and served not much more purpose than getting a flying aircraft. A thrust to weight ratio of about 1.6 was used, a weight to wing area ratio of 1.9. The calculations are below:

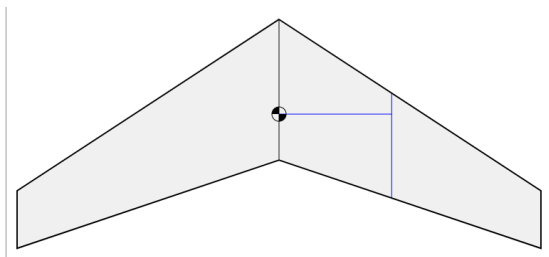
- Thrust of 0.375kg
- Desired T/W ≈ 1.5
- $0.375/\text{weight} \approx 1.5$; weight = 0.25kg
- Desired W/S ≈ 1.9
- $0.25\text{kg}/\text{wing area} \approx 1.9$
- Wing area $\approx 0.13\text{m}^2$

Sweep, Aircraft Dimensions, CG, and CP Optimizations

VSW CG and CP placement in forward sweep



FSFW CG placement, VSW CG placement in rearward sweep



As previously mentioned, to have a fair comparison the FSFW model needs to have optimal sweep angle. This was done by creating a mathematical 2D model of the flying wing in MATLAB, a programming software designed for calculations. Since the wing area was set, the only

other aspects that would shape the wing, (beside sweep) were the taper ratio and aspect ratio, which were chosen based on a maximum wingspan of 1 meter. It was chosen quite loosely since it wouldn't affect the comparison as the values would be applied to both aircraft. An iterative program structure tested every sweep angle from 0 to 45 degrees of centerline sweep, and resulted in an optimal sweep angle of 27.7 degrees, which resulted in an approximately 32 degree leading edge sweep. The code with further explanation is in the Appendix.

The CG was set to be $\approx 15\text{-}20\%$ of the MAC, as per standard CG distances for experimental flying wings which are designed to be relatively stable. The MATLAB program needed to calculate the CG and CP to determine certain aspects of drag, the CG location of the FSFW and VSW in the swept back position was set. From this, deciding how far the VSW should sweep forward was simply figuring out at what sweep angle would move the CP on top of the CG, which turned out to be a sweep forward of 19 degrees. The idea was to place the CG on top of the CP and reduce trim, reducing the effective weight on the aircraft, decreasing effective wing loading, decreasing induced drag, and increasing efficiency, at the cost of maneuverability and stability.

Preliminary Design and Prototype Versions

Once the basic dimensions of the flying wings were established, most of the work was done with Computer Assisted Design, or CAD. There were many partially completed and tested versions made in CAD, but only three versions of the FSFW and VSW were flown and tested.

Originally in the conceptual phase of the project, a single joint in the center of the wing to vary the sweep was considered, as it was the simplest and would retain the wing's straight shape. However, problems like motor mounting and placement of sensors lead to the conclusion that it would be best to have some sort of central

“fuselage” section that would stay unmoving. This resulted in a design which had two joints in the flying wing: one on each side, 150mm from the centerline. 150mm was chosen since it was close to the diameter of the propeller, and having some space to each side of the propeller for the wing to actuate was important.

The physical design of the joint was another important part of the preliminary design phase, and deciding the type and structure was quite important. It needed to be very simple and lightweight, yet still be strong enough to bear many times the weight of the aircraft in high G maneuvers. Simplicity was key, so the first design employed a compliant joint, which uses the elastic deformation of material to actuate. While this was simple and lightweight, it posed a number of problems, such as bending slightly in unintended axes, and having the servos constantly fight the springing force of the joint. During initial tests of the VSFW with this joint, the downwards force on the tips of the wing from the trim of the elevons rotated the outer portion of the wing positively about the pitch axis, creating a downwards force canceling out the trim force resulting in the inability to climb, resulting in an eventual crash. The event prompted rigid two part joint designs in the later and final versions.

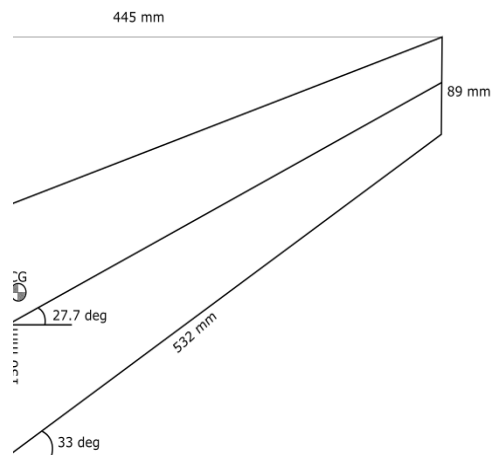
The shape of the airfoil was experimented with during the prototype versions. Originally it was planned to use a true airfoil shape, and I based the first versions around the Clark-Y airfoil because of its high lift to drag ratio and general forgiveness, especially during stall. However, it was difficult to maintain a steady shape over the wing as it pivoted, creating drag-inducing gaps where the two parts of the wing met, and because it was difficult to get a joint that was wide enough to prevent another wing-twist-caused crash. Thus, a stepped and even flat airfoil was designed, which proved to easily take a wide and sturdy joint. However, it was worse at creating lift than originally predicted, and simply was not flyable, resulting in a return to a true airfoil.

The flight computer also needed to be redesigned multiple times. The first iteration used an ATMEGA328 processor, the same on Arduino Nanos. It was very similar to an Arduino Nano, but was smaller and had fewer components. This necessitated the use of external components to flash and read output from the flight computer, which was tedious and very unreliable. It also performed quite poorly as far as processing time, resulting in jerky, delayed, and inaccurate movement of the motor and servos. The next designs all used the Teensy 3.2 microcontroller, which had 4.5 times the clock speed. This solved most issues.

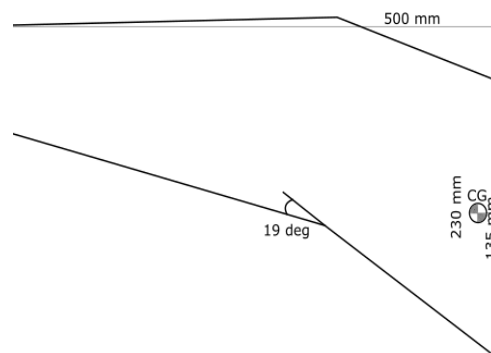
Final Design (Aero)

A Clark-Y airfoil with no reflex was used for the root airfoil, and was linearly interpolated to a flat tip airfoil geometrically twisted -5 degrees. This gave a flyable flight characteristic on the first try, after trim was applied.

Base design



Variable Sweep in forwards sweep



Final Design (Structure)

The final UAV was constructed from foam board, which is a 4mm thick paper-foam-paper sandwich, usually used as presentation boards. It was chosen due to the high strength to weight ratio for the cost, as well as ease of manufacture and unique build techniques. This comprised much of the structure, and with different build techniques along with tape and hobby hot glue could form spars, airfoils, control surfaces, hinges, and more. For parts that required more strength and accuracy, such as the flight controller mounts, wing sweep joints, and control horns, Fused Deposition Modeling (FDM) style 3D printing with Polylactic Acid (PLA) material was used. The FSFW takeoff weight was 218 grams, and the VSFW takeoff weight was 272 grams.

Final Design (Electronics)

The electronics are listed below:

- Custom flight computer designed on EasyEDA, custom manufactured by JLCPCB. Given the small space required on the UAV, traditional Printed Circuit Boards (PCBs) like the Arduino Uno are too large and heavy.
- PJRC Teensy 3.2 Microcontroller. It runs at 72 MHz.
- MPU6050 6 Axis Gyroscope and accelerometer. It measures rotational rate and acceleration in all 3 axes.
- Spektrum 4ch Sport Receiver. This takes the 2.5 GHz signal from the transmitter (controller) and feeds out Pulse Width Modulated (PWM) signals to the Teensy to process.
- Micro Quadcopter Motor. The motor of a micro quadcopter was calibrated to measure airspeed.

These electronics work together to take in pilot inputs, pilot flight mode commands, sensor inputs, and calculate how to actuate the motor, and

four servos- two for the elevons, two to change wing sweep.

The microcontroller was programmed, compiled, and flashed through the Arduino Internal Development Environment (IDE), using the C++ language.

The program first reads pilot input, and determines what flight mode to be in. There originally were three: manual flight, where the wings would be swept back and full control given to the pilot, cruising flight, in which the wings would be swept forward and a PID control law applied to the throttle to maintain airspeed (a sort of 'cruise control' for the UAV), and a short autonomous loop maneuver. Unfortunately, during testing, due to the inconsistencies with the Teensy controlling the throttle, and its slow reaction time, bits of turbulence could slow and stall parts of the aircraft.

A similar code structure was used with the FSFW, with the only difference being there was no code to vary sweep or adjust trim.

Experimental Execution

UAVs can be dangerous, so measures were taken to make sure nobody was injured. Both aircraft are under 300 grams, have a less than 1meter wingspan, and are categorized as "Small UAVs." The propeller is only 6 inches in diameter, and is mounted in the rear of the aircraft, but a throttle cutoff for emergencies was put on the transmitter, and in the event that signal was lost the throttle would instantly drop to zero. All tests and flights were done at a local park or flying field, with no people around.

Because it would be best to keep as many aspects of the tests the same between the FSFW and VSFW, it was decided to use the same electronics and simply transfer them over once the test was completed.

The FSFW was flown three times, to collect both efficiency and maneuverability data.

Maneuverability data was collected by executing the loop a total of 12 times, four for each

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

flight. Each loop was recorded with a phone camera at 30 frames per second, which was used to determine loop time and thus pitch rate.

Each flight lasted 2 minutes, and the start voltage and end voltage were written down, but because they were all done on a single charge it can be simplified by being thought of as a single 6 minute flight, with a start and end voltage.

Once the tests were completed, the electronics were transferred over to the VSFW, where many technical issues were found, delaying the data collecting flight. Firstly, weight distribution and CG issues resulted in a very nose heavy aircraft. It caused multiple crashes on takeoff, requiring multiple panels of the wing to be replaced or built over, adding extra weight. The plastic joint of the wing suffered the greatest damage, particularly from cartwheeling on the ground, requiring a complete replacement of that section of the wing, once again necessitating additional support, raising the weight to 272 grams, 22 grams above the intended weight of 0.25kg.

Only one flight was done with the VSFW, lasting 2 minutes and 40 seconds, ending in a human caused crash. Only two loops were executed. I, as the pilot, was not experienced with flying such an aircraft and during a turn forgot to set the sweep rearwards, and ended up in a sideslip into a nearby tree. Once recovered it was found that the outer portion of the wing had been torn off, flight computer damaged, and the motor completely dead, a complete rebuild of the wing and electronics would have been necessary to fly again. Because the end date of the project was just days away with unflyable weather predicted, unfortunately, the tests had to be ended short.

Data and Results

Raw Efficiency Data

UAV	Voltage Start and End	Average Voltage Difference/ Minutes	Total Data Collecting Flight Time
FSFW	8.39-7.65	0.12v /Min	6 Min
VSFW	8.33-7.9	0.14v /Min	2 Min, 40 Sec

Raw Maneuverability Data

FSFW	Total Loop Time	Deg/Sec
1	4	90
2	4.5	80
3	3	120
4	3.5	102.8571429
5	3.5	102.8571429
6	4	90
7	3.5	102.8571429
8	3	120
9	3	120
10	4.5	80
11	3.5	102.8571429
12	4	90
	Avg. Deg/Sec:	100.1190476

VSFW	Total Loop Time	Deg/Sec
1	4	90
2	5	72
	Avg. Deg/Sec:	81

Since the sample size was not very large due to crashes that broke critical components such as motors and flight computers it was difficult to make any significant interpretations for the flight data. Sources of error were also numerous, which

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

included me being a human pilot who doesn't necessarily fly consistently, weather, temperature and thus air pressure, servo trim, and general defects as a result of hard landings, crashes, repairs and possibly more.

More testing was definitely required to more confidently say whether both aircraft have similar maneuverability, but for the time being, it was safe to say that the variable sweep flying wing had similar maneuverability in comparison to the FSFW. Looking at the averages it seemed that the variable sweep flying wing was slightly less maneuverable, but given the clearly apparent variability in the data, it's difficult to say. For now, it can be concluded that both have the capability reasonable to change direction in flight (something the VSFW UAV can only do with a rearward sweep, showing that it's also possible to change maneuverability in flight).

As far as efficiency, there seemed to be no advantage of the VSFW over the FSFW. Although the variable sweep design overall had a slightly lower efficiency in comparison to the fixed sweep design, it is important to note that the changing wing sweep definitely still had a positive effect on efficiency. The variable sweep design had a 25 percent high takeoff weight but is only 17 percent less efficient. Though this wasn't a perfectly linear relationship between takeoff weight and efficiency, based on the data lowering the sweep angle for higher efficiency has been interpreted as true. With better design the VSFW theoretically can outperform a FSFW, but more work will need to be done.

Conclusions

Because of the low sample size, and the project still being in a prototype phase, the conclusion is that while the variable sweep flying wing concept would definitely be viable with more work, as changing the wing sweep does increase efficiency, more research and testing needs to be done to provide evidence that it can outperform a fully optimized FSFW. The things that were taken

into consideration while reaching this conclusion were not only the raw flight data but also a few qualitative observations from when I was building and flying.

Another reason why I believe this concept has more capability than what was shown directly in the data was due to qualitative observation during the project. Because the same airframe was used throughout the year to experiment with different mechanisms, resulting in it being patched up from many crashes, there were many defects resulting in additional weight and drag. I planned to rebuild both aircraft to get as accurate a comparison as possible, but in the end time constraints ended up halting the plan.

Furthermore, throughout this project I've found that there are plenty of other, likely better ways to increase efficiency and maneuverability. One example would be to employ thrust vector control. Rather than changing the aerodynamic qualities to exchange maneuverability with efficiency, changing the thrust angle could be a way to maintain the aerodynamic shape of the wing while using vectored thrust to rapidly force the aircraft into a new orientation.

The next step to exploring this proof of concept is to continue testing with a better and more accurately built UAV. Something made with foam board which is roughly assembled with hot glue works well as a first proof of concept but there are so many places in the building process that can be slightly misaligned which can affect efficiency. Each hard landing on wet ground can just slightly warp the foam, not to mention damages and repairs, which add to the weight and increase drag, something that likely negatively affected the variable-sweep design as it was experimented on the most.

Acknowledgments

This project was aided by Ethan Uehara and Daniel Moore, at Design Build Fly club at the University of Washington, for answering questions when I became stuck. I was also aided by Professor

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

Prazenica at the Embry Riddle Aeronautical University, for introducing control theory topics.

Sources Used

V., Author Rob. "History of the Flying Wing (from the Early 1900's & Beyond)." Aviation History - Century of Flight, 15 Dec. 2020,

<https://www.century-of-flight.net/history-of-flying-wings/>.

Hepperle, Martin. "Basic Design of Flying Wing Models." Basic Design of Flying Wing Models, 2018, <https://mh-aerotools.de/airfoils/flywing1.htm>.

Simpson, Bruce. "The Aerodynamics of Flying Wings (Part 1)." YouTube, YouTube, 19 May 2017, <https://www.youtube.com/watch?v=gkb11eKXM14>.

Sheperd, Samm. "Engineered Mini Flying Wing." YouTube, YouTube, 28 Feb. 2016, <https://www.youtube.com/watch?v=aSD69jdi2CE>.

Dunbar, Brian. "Past Projects - Ad-1 Oblique Wing." NASA, NASA, 10 May 2017, <https://www.nasa.gov/centers/dryden/history/pastprojects/AD1/index.html>.

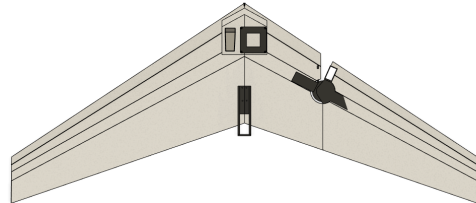
Smith, H.C. : The Illustrated Guide to Aerodynamics. 2nd edition. United States: McGraw-Hill, Inc, 1992. Print.

"Video Portal Main Page." Video Portal Main Page - MATLAB & Simulink, www.mathworks.com/videos.html.

Raymer, Daniel. Aircraft Design: A Conceptual Approach. Virginia: American Institute of Aeronautics and Astronautics, Inc., 2019. Print.

Appendix

CAD Model: Left half is the FSFW, right half is the VSFW



Completed VSFW



Completed FSFW



The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

VSWF executing a loop



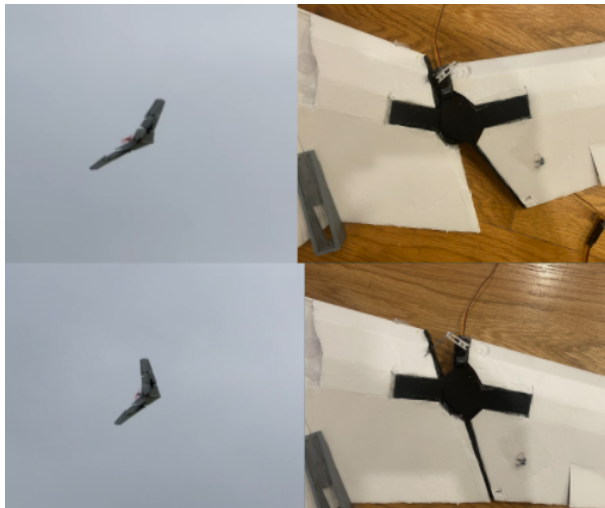
Demonstration of forward and rearward sweep.

Top Left: Forward sweep in flight

Top Right: Forward sweep during testing

Bottom Left: Rearward sweep in flight

Bottom Right: rearward sweep during testing



```
for i=1:d
PrevCdip=Cdip; %Variable to compare iterations
sdeg = sdegV(1,i); %sweep for this combination

HorizSp=cosd(sdeg)*sp; %Aerodynamic span
Ar=(HorizSp^2)/A; %Aspect Ratio

TaperRatio=tch/rch; %taper ratio
MAClength= (rch+tch)/2; %length of the Mean Aerodynamic
Chord (MAC)
MACdist=(sp/6)*((1+2*TaperRatio)/(1+TaperRatio));
%Centerline to MAC
AC=MAClength/4; %MAC distance (from leading edge)

%Calculate the force needed to trim the airplane, and add
it to the weight
ACDistFromTP=tand(sdeg)*MACdist+AC;
CGDistFromTP=tand(sdeg)*MACdist+(StabSetAsPercOfMAC)*MAClen
gth;
CGtAC=abs(ACDistFromTP-CGDistFromTP);
ActWT=(tand(sdeg)*(HorizSp-MACdist)-AC)+tch;
TrimForce=((wa*9.81*CGtAC)/ActWT)-AirfoilTorque;
we=wa+TrimForce;

e=1.78*(1-0.045*Ar.^0.68)-0.64; %Oswald Span Efficiency

%Calculate Wing Drag (take in account parasite drag of wing,
induced drag,
%fuselage drag (determined through CFD estimates), and trim
drag
v=5; %Velocity at 10m/s cruising speed
p=1.225; %Air pressure at sea level
q=(1/2)*p*v^2; %dynamic pressure
L=we*9.81; %Calculate lift force
Cl=L/(q*A); %Coefficient of lift
Cdi=(Cl^2)/(pi*Ar*e); %Induced drag coefficient

Cdp=0.023*q*A; %Parasite drag coefficient

Cdip=(Cdi+Cdp); %Wing coefficient of drag

%iteration comparison
if Cdip < PrevCdip
    bestCdp=Cdip;
    bestCdi=Cdi;
    bestDrg=Cdip;
    Sweep=sdeg;
    Span=HorizSp;
else
end
end
%request Sweep variable for answer
```

Matlab Program:

```
%modifiable variables
A=0.129; %Wing area
rch = 0.188; %root chord
tch = 0.07; %tip chord
StabSetAsPercOfMAC=0.15; %Percent of MAC for CG location
wa=0.25; %flying weight
AirfoilTorque=0; %positive is pitch up, zero for a
symmetrical airfoil
sp=0.7; %Structural span.

%iteration code
a=0; %sweep iteration start
b=0.1; %sweep iteration interval
c=40; %sweep iteration end
sdegV=[a:b:c]; %sweep iteration vector
d=(c-a)/0.1; %number of iterations

Cdip=100000; %starting drag value (arbitrary number)
```

Flight Computer Program

```
//Libraries
#include <Arduino.h>
#include <TinyMPU6050.h>
#include <Wire.h>
#include <Servo.h>

//setup connection to MPU6050
MPU6050 mpu (Wire);

//servos
Servo LeftElevon;
Servo RightElevon;
Servo ESC;
Servo LeftWing;
Servo RightWing;
```

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

```
//trim
int leftTrim = 0;
int rightTrim = 0;

int ThrottlePgain = 22.5; //throttle proportional gain
int ThrottleIgain = 1; //throttle integral gain

int Extend; //extend or retract the wing

int vel; //velocity variable
int velSetpoint = 4; //speed setpoint
int velError; //velsetpoint - vel

int throttle; //rx input left stick y axis
int rollIn; //rx input right stick x axis
int pitchIn; //rx input right stick y axis
int mode; //rx input for switch

int PitchGyro; //pitch angle measured soley based off
gyroscope
int Pitch; //pitch angle measured
int Roll; //roll angle measured

int RollRate; //rate of roll
int PitchRate; //rate of pitch

int pitchIntegral; //integral of pitch error
int rollIntegral; //inegral of roll error
int velIntegral; //integral of velocity error

int reOut;
int leOut;
int escOut;

void setup() {

    Serial.begin(115200); //serial output for debugging

    mpu.Initialize();//inititalize the mpu6050

    //assign the outputs to pins
    ESC.attach(23, 1000, 2000); //servo 1
    LeftElevon.attach(22, 1000, 2000); //servo 2
    RightElevon.attach(21, 1000, 2000); //servo 3
    LeftWing.attach(1, 1000, 2000); //servo 4
    RightWing.attach(2, 1000, 2000); //servo 5

    pinMode(7, OUTPUT); //LED pin
    pinMode(A2, INPUT); //airspeed

    //for the rx input
    pinMode(10, INPUT);
    pinMode(3, INPUT);
    pinMode(4, INPUT);
    pinMode(5, INPUT);

    //arm the ESC
    ESC.write(40);
    delay(1000);
    ESC.write(0);

    //startup LED two, one second blinks
    digitalWrite(7, HIGH);
    delay(1000);
    digitalWrite(7, LOW);
    delay(1000);
    digitalWrite(7, HIGH);
    delay(1000);
    digitalWrite(7, LOW);

}

void loop() {

    RxDetect(); //detect pilot input

    //fully pilot controlled flight
    if (mode == 1) {
        Sensor_Inputs(); //detect environment
        escOut = throttle; //ESC output is manual
        ServoOutput(); //output to actuators.
```

```
        Extend = 0;
    }

    //efficiency testing
    else if (mode == 2) {
        Sensor_Inputs();
        ThrustPID(); //PID controller for thrust
        ServoOutput();
        Extend = 0;
    }

    //do a loop
    else if (mode == 3) {

        //Get the aircraft into level flight
        while ((Roll < -20 or Roll > 20 or Pitch < -20 or Pitch
> 20 or RollRate > 60 or RollRate < -60 or PitchRate > 60
or PitchRate < -60) and mode == 3) {
            Sensor_Inputs();
            RxDetect();
            ThrustPID();
            ServoOutput();
        }

        //initate the loop, with full extension of the elevons
        and full throttle. Wait until the first threshold at aprox
        135 degrees to continue on.
        while (mode == 3) {
            Extend = 1;
            RxDetect();
            rollIn = 90; //stay level
            pitchIn = 45; //pitch up
            throttle = 180; //full thrust
            ServoOutput();
        }
    }

    //Detect signal from sensors
    void Sensor_Inputs() {
        //speed
        int constant = 1;
        int RawMotor = analogRead(A1);
        vel = sqrt(RawMotor / 3);
        //attitude
        mpu.Execute();
        Pitch = (int(mpu.GetAngY()));
        Roll = (int(mpu.GetAngX()));

        PitchRate = (int(mpu.GetGyroY()));
        RollRate = (int(mpu.GetGyroX()));
    }

    //control law for the throttle, which uses a PI controller
    void ThrustPID() {

        velError = velSetpoint - vel; //find the error

        //integral cutoff
        if (velIntegral < 0) {
            velIntegral = velIntegral + 10;
        }

        else if (velIntegral > 180) {
            velIntegral = velIntegral - 10;
        }

        //integral
        else {
            velIntegral = velIntegral + velError;
        }
        //Set throttle to the controller's command
        escOut = (velError * ThrottlePgain) + (velIntegral *
ThrottleIgain);
        escOut = (constrain(escOut, 0, 180));
    }

    //Detect pilot commands
    void RxDetect() {
        //pilot command
```

The Variable Sweep Flying Wing: Investigating a Novel Concept for Small-Scale UAVs

```
throttle = pulseIn(10, HIGH, 25000); //channel 1
throttle = map(throttle, 1100, 1900, 0, 180);
rollIn = pulseIn(4, HIGH, 25000);
rollIn = map(rollIn, 1100, 1900, 45, 135);
pitchIn = pulseIn(5, HIGH, 25000);
pitchIn = map(pitchIn, 1100, 1900, 45, 135);
mode = pulseIn(3, HIGH, 25000); //channel 4

if (mode < 1300) {
    mode = 1;
}
else if (mode < 1700) {
    mode = 2;
}
else {
    mode = 3;
}
}

//output signals to servos and ESC
void ServoOutput() {

    leOut; //left elevon output
    reOut; //right elevon output

    //mixing this one is one axis maxes the servos out
    leOut = (rollIn - 45) + (pitchIn - 45);
    reOut = (rollIn - 45) + (135 - pitchIn);

    leOut = constrain(leOut, 45, 135);
    reOut = constrain(reOut, 45, 135);

    leOut = leOut + leftTrim;
    reOut = reOut + rightTrim;

    leOut = constrain(leOut, 0, 180);
    reOut = constrain(reOut, 0, 180);

    if (Extend == 0) {
        LeftWing.write(90);
        RightWing.write(90);
    }

    else if (Extend == 1) {
        LeftWing.write(90);
        RightWing.write(90);
    }

    LeftElevon.write(leOut);
    RightElevon.write(reOut);
    ESC.write(escOut);
}
```