Single Axis Stability Autonomous Glider Control

Brian Study
Embry-Riddle Aeronautical University

Follow this and additional works at: https://commons.erau.edu/student-works

Part of the Aeronautical Vehicles Commons, Aviation and Space Education Commons, and the Navigation, Guidance, Control and Dynamics Commons

Scholarly Commons Citation

This Article is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Student Works by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.
Single Axis Stability Autonomous Glider Control

Brian J. Study
Embry-Riddle Aeronautical University, Prescott, AZ, 86301

To investigate and increase knowledge on autonomous control systems, an autonomous glider was fitted with a control system with the purpose of creating a craft that would be capable of maintaining a wings level condition despite perturbations to the trim condition. The glider measures bank angle and roll rate information from an accelerometer and gyro, before it relays the data to an equipped microcontroller. Programmed on the microcontroller is a control law to take the input from the sensors and issue a command to a single servo that controls both ailerons, allowing for the vehicle to autonomously correct its bank angle. The glider has proven to recover up to 15° in a relatively straight path. Although at higher initial bank angles the system is able to correct, the path is less straight as the small angle approximations and other assumptions become less accurate, leading to a greater amount of lateral drift. Corrections can be made to implement a fix by increasing the gain of the system, but in doing so presents structural issues in the current model.

Nomenclature
AR = Aspect Ratio
Φ = Bank Angle
Lipo = Lithium Polymer
KΦ = Gain Bank Angle
KP = Gain Roll Rate
Ksys = Gain System
P = Roll Rate
λ = Taper Ratio
τ = Aileron Chord %

Introduction
As the push towards autonomous aircraft grows in industry, the need for autonomous and self-guiding systems also increases. Effective control systems increase the autonomy of an aircraft, reducing the need for a competent pilot to obtain a controllable system. The control theory introduced in [1] applies for small perturbations, making small angle and other approximations. The further from the small angles the craft reaches, the less accurate the predictions become. The focus of this study is on Φ (bank angle), as the control in a single axis can be obtained, it is a simple manner to gain control of all other axis. Accelerometers and Gyros can be used in combination to obtain P (roll rate) and Φ, which can then be utilized as an input to the system to give a command for aileron deflection. These systems are small enough to be carried by a small balsa glider, which was built and tested to maintain the wings level even with the presence of perturbations.

1. Aerospace Engineering Student (Junior), College of Engineering, studyb@my.erau.edu
**I Equipment**

The accelerometer chosen was an MPU-6050 Accelerometer and Gyroscope integrated circuit. The accelerometer is capable of reading both acceleration and orientation across all three axes, making it capable of fully tracking motion in all directions. The accelerometer has multiple sensitivity settings, but the chosen settings were 2gs for the accelerometer and 250 degrees/s for the gyro [2]. Using this system, it is possible to obtain both Φ and P for the system, which can be used by the microcontroller to make decisions. In most situations, the optimal location for the gyro-accelerometer would be at the center of gravity, however as the craft only controls one axis the only placement that matters is the correct Y-axis position, so that the aircraft rotates around the gyro-accelerometer (X-axis).

The chosen microcontroller is an Arduino Nano. It contains a variety of pins capable of taking in signals, and outputting signals, and can be powered by anywhere from 6 to 14 volts depending upon the power draw and the power port being used [3]. The Arduino, servo, accelerometer system can be seen in Fig. 1.

As the heaviest piece of equipment was the battery, efforts were made to find a lightweight battery capable of producing the voltage in the required range of the Arduino. However, many of the batteries considered were fairly large and never intended for lightweight glider applications. To get the proper voltage, two 3.7 volt single cell Lithium-Polymer batteries were chosen to power the system. They can be connected in series, to provide 7.4 volts, and can interface with other on hand equipment for charging and other maintenance.

The overall cost of the system and glider is detailed below in Table 1:

<table>
<thead>
<tr>
<th>Table 1: System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
</tr>
<tr>
<td>Arduino Nano</td>
</tr>
<tr>
<td>MPU-6050</td>
</tr>
<tr>
<td>Build Materials</td>
</tr>
<tr>
<td>Servo</td>
</tr>
<tr>
<td>1 Cell Lipo Battery (2)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
</tr>
</tbody>
</table>

Prices listed above are approximated, as often when larger quantities are purchased, the price per unit decreases. The entire glider can be built for $31, and can be programmed using the Arduino software, available online for free.

**II Procedure**

A simple and efficient way to calculate the designs of a balsa glider is to use an excel spreadsheet, and apply the applicable equations to the design specifications. The primary method of calculation for the effects of the lifting surfaces was the Polhamus formula [1], which converts the two-dimensional lift characteristics into 3-D. As the Polhamus Formula is only valid for $3 \leq AR \leq 8$ and $0.4 \leq \lambda \leq 1$, the tail surfaces and wings need to fall within that range.

Due to available materials and other constraints, the maximum possible chord allowable was 4 inches, to allow the wing to be constructed from a single piece of balsa wood to avoid complications.

Control of the craft will be completed through ailerons on the wings, which can be accounted for as a combination of $\tau$ (control surface effectiveness) and dimensional coefficients as seen below:

$$L_{\delta A} = \frac{\bar{q}SCL_{\delta A}}{I_{XX}}$$  \hspace{1cm} (eq. 1) [1]

$$Y_{\delta A} = \frac{\bar{q}SCY_{\delta A}}{m}$$  \hspace{1cm} (eq. 2) [1]
\[ \tau = \text{chosen by a convenient geometry of the aircraft wing (0.25), and the rest of the aileron was sized according to historical data seen in [4]. Originally the ailerons were chosen to be at the wingtips, but due to mechanical limitations discussed later they were moved inboard on the wing and expanded in size.}

Once the geometry of the glider and the ailerons is created, the control law can then be calculated. This is accomplished by creating a gain matrix, defined from the dynamic equations of motion of an aircraft system. The dynamic equations define transfer functions, which defines the poles and zeroes of a system. The poles and zeroes of the system are then utilized to create a gain matrix \( K \), which when combined with the output from command signals defines the control law. By changing the locations of the poles and zeroes, the response of the system can be changed, allowing an aircraft to respond in a more desired manner. The original poles of the system for the given geometry were determined and displayed below in Table 2:

<table>
<thead>
<tr>
<th>Longitudinal</th>
<th>Lateral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Period</td>
<td>-11.4 ± 12.5i</td>
</tr>
<tr>
<td>Phugoid</td>
<td>-0.145 ±1.85i</td>
</tr>
<tr>
<td></td>
<td>Spiral</td>
</tr>
</tbody>
</table>

Any pole in the left half plane indicates instability. As the above table indicates, all the modes were inherently stable except the spiral mode, which was highly unstable. Thus, to create a stable glider, the spiral mode pole needed to be moved.

The number of poles and zeroes that can be moved to change the response are limited by the number of control systems, and the relevance of each control system to the desired parameters. As the study glider only possesses a single axis of control, the only meaningful gains that the control law matrix can effect are \( K_{\Phi} \) and \( K_{\Phi} \). Additionally, only a single pole, belonging to the lateral directional set can be moved.

In this case, the spiral pole will be changed to modify the response of the vehicle. By iterating the system for a variety of pole locations, a damped system can be created that, given the proper commands, can control the \( \Phi \) of the craft and allow the craft to recover to a wings level position.

The response of the dynamic system was simulated with a base script in MATLAB provided for the Aircraft Stability and Control class and modified for the purposes of this directed study, and for the aileron roll system, a simulated control input was sent to the controller, which allowed for the visualization of the vehicle response. The spiral mode pole was moved until the response seen in the simulation damped out and reached a reasonably damped system with relatively little overshoot. The location of the new spiral mode pole was -50, which lead to the creation of the gain matrix \( K_{\Phi} \) and \( K_{\Phi} \) to equal 0.1590 and 7.5100 respectively. The following image seen below in Fig. 2 details the response of the glider, and the simulation results utilized to shape the control response.

![Fig. 2: Glider Response to Control Input](image-url)
initial \( \Phi \) perturbations. The simulation with no controller can be seen on the left, with the simulation with the controller can be seen on the right. For simulation purposes to test the case with the most potential to experience lateral drift, the glider launch speed was 10 ft/s above the trim condition, launched with no angle of attack, but a varied \( \Phi \) depending on the launch.

As expected, the further from the wings level condition, the worse the perturbation becomes and the harder it is for the system to properly compensate, as it gets farther away from the small angle approximations and other assumptions. It is however observed that the glider shows significant improvement in even the most extreme cases. And according to the simulation, the aircraft can recover and still land within a ten feet margin while thrown at a \( \Phi \) of 15\(^\circ\), while the uncompensated glider falls outside of these bounds.

By integrating the input from the accelerometer, it is possible to obtain the \( \Phi \) of the system, and thus formulate a desired command to the ailerons, allowing the system to function autonomously and react to the changing conditions along the entirety of the glide path. The Arduino was programmed to take the data, run some conversions to convert raw data into usable information, and then directed into a command for servo direction. The following equation from the code indicates the aileron deflection commanded by the system:

\[
\delta_{\text{Aileron}} = K_{\text{Sys}} \ast (K_{\Phi} \ast \Phi + K_P \ast P) \quad \text{(Eq. 4)}
\]

\( K_{\text{sys}} \) is a gain applied to the entire system to tune the magnitude of the response. It was initially introduced to reduce the servo deflection, to prevent the servo from overextending and breaking structural components of the control system. The initial \( K_{\text{sys}} \) was chosen to be 0.15, with the intention of changing the value if mechanical issues presented a difficulty, or if the response was undesired and required a new gain and an updated simulation. The construction of the model created geometry in the control system such that angular displacement of the servo equated to an equal angular displacement for each aileron.

### III Results and Discussion

As the design was being built, it was noticed that the mechanical constraints on the ailerons and other mechanisms would not allow for the ailerons to be placed on the wingtips because of complicated geometry that required materials beyond the limits of available parts. Due to this mechanical limitation, the ailerons were moved further inward and expanded in length due to the loss of effectiveness created by moving the control surfaces inward. The modification still resulted in a net loss of control authority, but it was deemed to be an acceptable amount. Due to time constraints, the current model was adapted to the design change, resulting in the design seen in Fig. 6.

The aircraft was observed to pitch up and stall, due to the center of gravity location, which was observed to be further aft than expected. This was due to an overestimation of the weight of the electronic system, which was quickly remedied with ballast. Despite the pitching moment leading to stall, the ailerons still performed as expected before and after the center of gravity correction. When given a hand launch of approximately level launch, the aircraft damped out nearly instantly and glided the entire flight path with wings level.

When launched with a \( \Phi \) of approximately 15 degrees, the aircraft first flew a few feet with the given \( \Phi \) before very quickly deflecting the ailerons and correcting the angle, bringing the wings level.
The craft reached a stable flight within two seconds of flight time, without oscillations or a large amount of overshoot. When launched at more extreme banks angles (+30°), the craft was observed to correct to wings level very quickly in the first few seconds, and then overshoot slightly, to approximately 15°. At this point, the glider impacted the ground before it was ascertained that the oscillation damps out. However, this angle is likely beyond the small perturbation and other assumptions. The more extreme bank angles are farther outside the small perturbation assumption, and as such the results at these angles can be expected to have greater error than at smaller Φ.

Additionally, Ksys was investigated as a possible cause to the inability to compensate at higher Φ perturbations. Ksys was increased to 0.20, and a new simulation was run, seen in Fig. 7, which determined that the system was more controllable than the old system, but grew worse at Ksys = 0.25.

![Fig. 7: Increased Ksys Simulation Φ Initial 45°](image)

Although Ksys of 0.2 shows greater results, the current model will be unable to structurally handle the larger gain, as the larger deflection angles may cause structural damage to the glider.

The test flights compare well to the simulation results and are observed to follow many of the same trends, with the exception of the range. The simulation creates a good approximation of the glider’s behavior, even though the response is not as controlled as desired, it can still be predicted.

The decreased range and flight time is due in part to being launched at or below the trim speed. This is due to the launcher not possessing the ability and skill to launch the glider at an acceptable pitch angle with the speed required to reach the longer-range estimates of the glider. The glider does obtain enough time in the air to prove that it functions as designed.

The methodology for converting the Φ correction system to other control surfaces such as elevator and rudder is a simple matter that can be easily adapted to other control systems. The mechanical modifications would involve the introduction of additional servos and corresponding control surfaces, with the relocation of the gyro-accelerometer to the center of gravity, so that accuracy can be obtained in all axis. The same method used for the Φ angle to obtain gains can be used, but requires modifying additional poles to change the response of the controller.

IV Conclusion

After an aileron redesign for structural reasons, the glider is capable of correcting for bank angles within the limitations of small angle assumptions. Outside the bounds of these assumptions, the effectiveness is greatly reduced and it becomes much harder to control, but the aircraft behavior can still be accurately predicted by simulations. This leads to the idea that a gain increase would grant greater control at extreme Φ, but the physical system would be unable to handle the additional servo deflection.

V Acknowledgements

An overwhelming thank you to Dr. Kenneth Bordignon for agreeing to be my mentor and assisting with the design and providing the knowledge and tools necessary to make this directed study possible. Also, another thank you to the Embry-Riddle Aeronautical University Honors Program for allowing me to use this as an Honors directed study.

VI References


<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Units</th>
<th>Notes</th>
<th>Errors</th>
<th>Name</th>
<th>Value</th>
<th>Units</th>
<th>Notes</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glider ID</td>
<td>1001</td>
<td></td>
<td></td>
<td></td>
<td>Glider ID</td>
<td>1001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>15.0</td>
<td>m</td>
<td></td>
<td></td>
<td>Length</td>
<td>15.0</td>
<td>m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wingspan</td>
<td>12.0</td>
<td>m</td>
<td></td>
<td></td>
<td>Wingspan</td>
<td>12.0</td>
<td>m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
<td>Wing Area</td>
<td>12.0</td>
<td>m²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td>Aspect Ratio</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix: Arduino Code

// Control Law Arduino Code for Stability Axis Control Glider
#include <Wire.h>
#include <Servo.h> // Servo Library
Servo servolo; // Servo
const int MPU_addr = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

// Variable Setup
float COMMAND = 90;
float COMMANDb;

// Gain Variables
float Kphi = 7.5100; // Gain for Kphi
float Kp = 0.1590; // Gain for Kp
float GainMod = 0.15; //

// Mechanical Variable Values
float Def = 0;
float DefFactor = 1; // Experimentally Determined Gain on Deflection of Aileron
float Conv;

// Gyro and accel Variables
float GyXd;
float AcXd;
float vleoc;
float V0 = 0;
int Flag = 0;
float Gravity = 32.2; // Feet/Second^2

// Time Variables
unsigned long Time; // Time for accel calculation
unsigned long To = 0.0; // Initial Time
unsigned long TCur; // Current Time
int PosYd = 0;

void setup(){
    Wire.begin();
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x6B);
    Wire.write(0);
    Wire.endTransmission(true);
    Serial.begin(9600);
    // servo setup
    servolo.attach(9);
}

void loop(){
    // Accelerometer Functionality
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B);
    Wire.write(0);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true);
    AcX = Wire.read()<<8|Wire.read();
    AcY = Wire.read()<<8|Wire.read();
    AcZ = Wire.read()<<8|Wire.read();
    GyX = Wire.read()<<8|Wire.read();
    GyY = Wire.read()<<8|Wire.read();
    GyZ = Wire.read()<<8|Wire.read();

    // Data Conversions
    GyXd = GyY/131.0*3.14/180; // Converts Raw data to acceleration in ft/s^2

    // Time Calculation For Acceleration
    TCur = millis();
    Time = (TCur-To);
    To = millis();

    // Velocity calculation (RollRate)
    vleoc = V0 + AcXd*Time/1000;
    V0 = vleoc;

    // Control Law Command Sequence
    Def = GainMod*(Kphi*GyXd + Kp*AcXd); // Deflection Command
    COMMANDb = Def*180/3.14*DefFactor; //kök COMMAND = 90-COMMANDb. // Data Sent to Servo

    // Servo Operation
    servolo.write(COMMAND); // Servo Command Line
delay(5); // Allows servo time to move before receiving new command
}