Experiment A11
Flight Control System
Procedure

Deliverables: checked lab notebook, demonstration of working device to Lab TA

Recommended Reading: Chapters 2, 9, 16, and 18 of the textbook

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab. **Each item on the score sheet must be completed before the end of lab for you to receive credit for it.**

Overview

The amount of lift force generated by an airfoil or wing can be change by adjusting the angle of a flap. (Flaps are formally known as "aerodynamic control surfaces", but most people just call them flaps.)

Shown in Fig. 1, the horizontal tail of an aircraft has a flap known as the "elevator", which is used to control the angle of attack or pitch angle of the aircraft. A special motor known as a servo is used to adjust the flap angle δ_e, which causes the aircraft to pitch up or down to an equilibrium pitch angle θ.

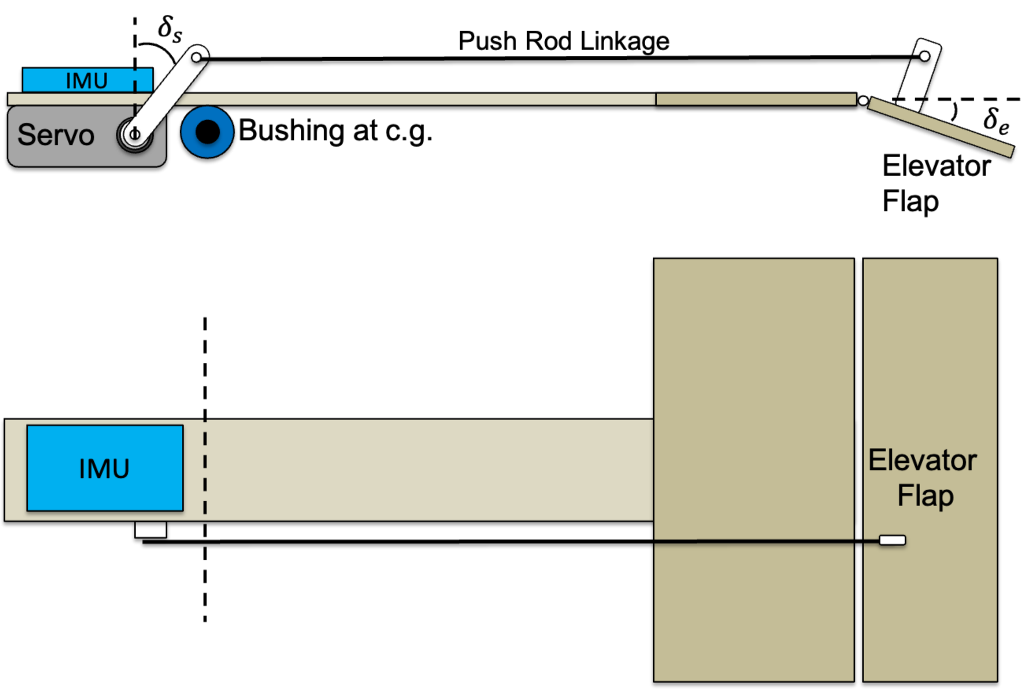

Figure 1 – A servo motor is used to adjust the elevator flap angle. The entire airframe is allowed to rotate about its center of gravity (c.g.) via a set of pillow block bearings. The pitch angle of the airframe is measured using an inertial measurement unit (IMU).

In this lab, you will create a fly-by-wire flight control system, where a pilot can control the pitch angle θ using an analog joystick. An Arduino UNO microcontroller will read the analog signal from the joystick, map it to servo angle, and adjust the servo angle accordingly. The rotation of the servo arm will cause the flap to rotate via a push rod linkage. An inertial measurement unit (IMU) will measure the pitch angle and provide the pilot with real-time flight data.

Part I: Microcontroller and Analog Joystick

You will begin by wiring up the Arduino and analog joystick to a breadboard. Then, you will program the Arduino to read the analog voltage output of the joystick and print it to the serial monitor.

analogRead() – This function reads an analog voltage input from 0 – 5V and maps the voltage to a 10-bit integer value from 0 – 1023.

Procedure

1. Connect the Arduino UNO to the lab computer via the USB cable. You should see a green LED light up on the Arduino.
2. Take the joystick, jumper wires, and small white breadboard out of the plastic bag.
3. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the white breadboard. Use the orange handheld DMM to verify that it is providing +5V of power.
4. The analog joystick uses two spring-loaded potentiometers to measure the angle of the stick about two axes. You will only use one of the two axes of the joystick—the one with wires connected to the potentiometer. Measure the resistance between the various wires coming off the joystick. Based on what you learned about potentiometers in the lecture video, decide if it is working or not.
5. Wire up the joystick. Make the following connections:
    a. Blue wire → +5V on breadboard
    b. Yellow wire → GND on breadboard
    c. Green wire → A0 analog input on the Arduino
6. Download the Joystick Code Template from the A11 webpage. Fill in the '***' with the pin number for the analog joystick.
7. Save the code with an intelligent file name (i.e. "A11_joystick_yourName.ino").
8. In the Arduino IDE software, go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
9. Press the arrow button to compile the program and send it to the Arduino.
10. Go to "Tools" > "Serial Monitor" (or press "Ctrl + Shift + M") to view the output from the "Serial.print()" commands. You should see the joystick output printed as values from 0 - 1023.
11. Move the joystick and observe how the output changes. (Note that only one of the two joystick axes is connected to the Arduino.) Write down the values corresponding to the minimum, maximum, and home joystick positions.

12. **Demonstrate the working system to the TA to receive points on your score sheet.**
13. Leave the joystick and Arduino connected. You will need them for later parts of the lab.

Part II: Servo Motor and Four-bar Linkages

You will now implement and test the servo motor which controls the elevator flap angle via a four-bar mechanism. The servo motor takes a digital pulse signal and maps it to a motor shaft angle. The Arduino servo code library has functions for generating these digital pulses.

Procedure

1. Use the 6mm stainless steel shaft to mount the airframe to the pillow block bearings.
    a. Hold the blue bushing between the two pillow block bearings, and slide the shaft through all three.
    b. Slide the shaft collars on the ends of the shaft, flush with the pillow block bearings, and gently tighten the set screws with a 2.5mm allen wrench.
2. Use male-to-male jumper wires to make the following connections.
    a. Servo Brown → GND on breadboard
    b. Servo Red → +5V on breadboard
    c. Servo Orange → Pin 9 on Arduino
3. Focus your engineering awareness on the four-bar mechanism that couples the servo to the elevator flap. Check that all of the linkages are connected between the servo shaft, servo arm, push rod, and elevator flap, as illustrated in Fig. 1. If not, ask the TA or lab instructor for help.
4. Download the Servo Example code from the A11 webpage. Fill in the '***' with a servo angle value between 45 and 135. Save the code, then upload it to the Arduino. You should see the servo arm rotate.
5. Try different values for the servo angle δ_s and observe how it effects the elevator flap angle δ_e. Note that the two angles will not necessarily be the same, as the 4-bar linkages do not form a parallelogram. Record the following servo angle $\delta_s$values in your lab notebook corresponding to:
    a. Home position, $\delta_e = 0°$ - The elevator flap is parallel with the airframe
    b. Maximum position – The push rod begins to bend or it reaches a geometric limitation of the four-bar linkages. (It is OK for it to be greater than 135 degrees.)
    c. Minimum position – Going the opposite direction, the push rod begins to bend or it reaches another geometric limitation of the four-bar linkages. (It is OK for it to be less than 45 degrees.)
6. **Demonstrate the working system and show your lab notebook to the TA to receive points on your score sheet.**

Part III: Design Challenge 1

Create a system where a pilot can control the angular position of the elevator flap using the joystick. The system should allow for very fine control of the flap angle using the joystick.

- Combine your codes from Parts I and II.
- Use the map() function to map the minimum value of the joystick to the minimum position of servo, and the home value of the joystick to the home position of servo.
- The map() function should take the joystick value as an input and output an angle for the servo.
- The elevator should return to $\delta_e = 0°$ when the joystick is released.
- **Demonstrate the working system to the TA to receive points on your score sheet.**

Part IV: Inertial Measurement Unit (IMU) – MEMS Gyroscope

You will now implement an IMU to measure the pitch angle of the airframe and provide real-time flight data to the pilot.

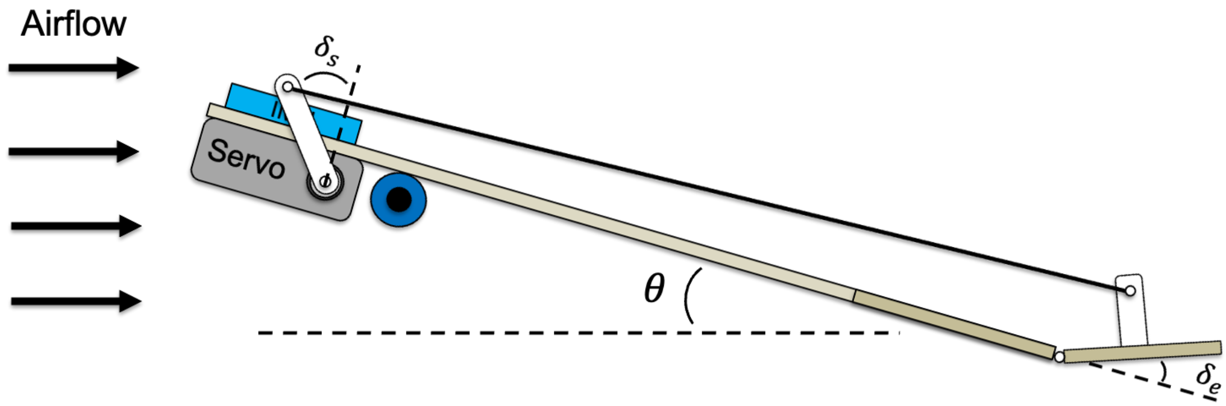

Figure 2 – Adjusting the elevator flap angle $\boldsymbol{\delta_e}$ creates an aerodynamic force that rotates the airframe to a trim pitch angle $\boldsymbol{\theta}$. The pitch angle $\boldsymbol{\theta}$ will be measured by a BNO-055 MEMS IMU gyroscope mounted to the airframe.

Procedure

1. The BNO-055 9-Axis IMU has a digital I2C output, similar to the sensors you used in the digital sensors lab. Make sure the BNO-055 is securelymounted to the small breadboard on the airframe, and use the longer male-to-male jumper wires to make the following connections.
   a. GND on IMU → GND on breadboard
   b. Vin on IMU → +5V on breadboard
   c. SDA on IMU → SDA on Arduino
   d. SCL on IMU → SCL on Arduino

2. Go to the Adafruit webpage linked below, and follow the "Software" instructions for installing the Arduino driver and libraries for the BNO-055.

   https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/arduino-code
3. Download the BNO-055 9-axis IMU code from the A11 website, and save it to your computer.
4. Connect the Arduino to the computer, select the appropriate COM port, and load the IMU code to the Arduino.
5. Open the "Serial Monitor" to see the output displayed as text. Make sure the Baud rate is set to 115200. You should see the three Euler angles—pitch θ, roll ϕ, and yaw ψ—printed in the serial monitor in values of degrees along with their respective angular speeds.
6. Slowly rotate the airframe to its limits. You should see that one of the angles changes significantly more than the others. This angle corresponds to the pitch. Modify the code to print just the pitch angle θ as a single vertical column of values using Serial.println().
7. **Demonstrate the working sensor and code to the TA to receive points on your score sheet.**

Part V: Design Challenge 2

Combine all of the previous parts of the lab to create a fly-by-wire system with real-time flight data displayed in the serial monitor. You must be able to control the pitch using the joystick to within an error of $\pm 1°$.

- Make sure you include all of the libraries at the beginning of the code.
- Use a Baud rate of 115200 in the Serial.begin() function of the setup. Make sure the serial monitor is set to this baud rate.
- The parameters "joystick value, servo angle, pitch angle" should be printed to the serial monitor separated by commas and spaces.
- **Demonstrate the working system to the TA. (The TA will bring a blower fan to your work station when you are ready.) You should be able to use the joystick to set the pitch angle to either +10° or -10° with an allowable error of 1°.**

Clean-up

To receive full credit, you must return the lab bench to its initial state:

- Disconnect the joystick and servo. Remove all jumper wires from the Arduino and breadboard.
- Put the joystick, jumper wires, and white breadboard back into the plastic bag.
- Leave the BNO-055 IMU chip in the small breadboard on the airframe.
- Loosen the set screws on the shaft collars a half turn. Remove the shaft collars and the 6mm shaft. Gently set the airframe off to the side.
- Disconnect the USB cable from the computer.

Data Analysis and Deliverables

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab.

Appendix A

Equipment

- Arduino UNO Microcontroller
- 6ft USB cable
- Bag containing:
    - Small breadboard
    - Jumper wires
    - Joystick (ServoCity SKU: 605612)
- 2060 slotted rail with pillow block bearings
- 6mm shaft with collars
- Airframe with 6mm shaft coupling, servo, and BNO-055 IMU
- XPOWER P-80A 600 CFM Centrifugal Air Mover – Amazon # B076VR5PJC