Motorized Throttle Quadrant for Flight Simulation

By

Team 45 — Wendi Fu, Ziang Guo, Yuqi Xue

{wendifu2, ziangg2, yuqixue2}@illinois.edu

Project Proposal for ECE 445, Senior Design, Spring 2021

TA: Chaitanya Sindagi

18 Feb 2021

1 Introduction

1.1 Objective

Modern airliners are equipped with multiple flight automation systems to alleviate the workload of pilots during long flights. One of these automations common fitted is the Auto Throttle (A/T) system. A/T is able to fully control the engine thrust demand of the airplane in order to achieve constant airspeed cruise and provide crucial safety functionalities. To improve the situational awareness of the pilots, Boeing, alongside many major airplane manufacturers, fitted their airplanes with motorized throttle quadrants. When A/T commands thrust settings on these airplanes, the thrust levers will move accordingly. Such system allows the pilots to regain control of the throttle at any moment with full awareness of the current thrust settings. This is a crucial control characteristic of these airplanes.

Current main-stream consumer-level flight simulation peripherals use spring tension to simulate the weight felt by the pilot when operating the control column. However, no effort is made to simulate the synchronous movement of the throttle quadrant with A/T commands. Home-based simulation pilots often have to manually adjust the thrust levers to match the actual thrust setting inside the simulator, hence deteriorates the experience. Being able to recreate the motorized throttle will bring the realism of the flight simulation experience to a whole new level.

With this project we aim to design and create a motorized throttle quadrant that can interface with main-stream simulation software (Microsoft FSX & Lockheed Martin Prepar3Dv4) via universal protocol (e.g., USB) and synchronize thrust levers' positions with A/T commands. Figure 1 is an illustration of such process in action which vaguely represents the final product's main functionality.

To better evaluate the outcome of the project, three major requirements must be met:

- The throttle quadrant must be recognized by both the OS (Windows 10) and flight simulation software as a valid input source. The flight simulator must be able to calibrate the throttle quadrant input for accurate, smooth operation of the aircraft.
- If A/T is engaged and commands a thrust change, the throttle quadrant must mimic the movement of the thrust levers shown in the simulator's virtual cockpit smoothly at comparable rates. The response delay must be unnoticeable. The thrust levers must stop accurately and firmly in the commanded

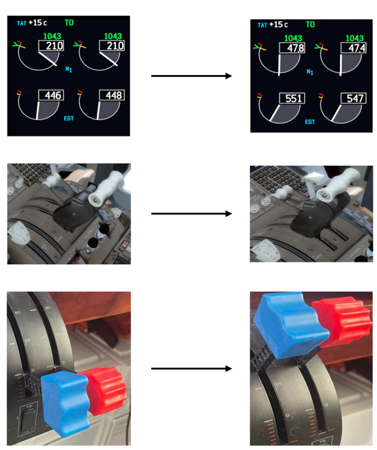

Figure 1: Illustration of device operation.

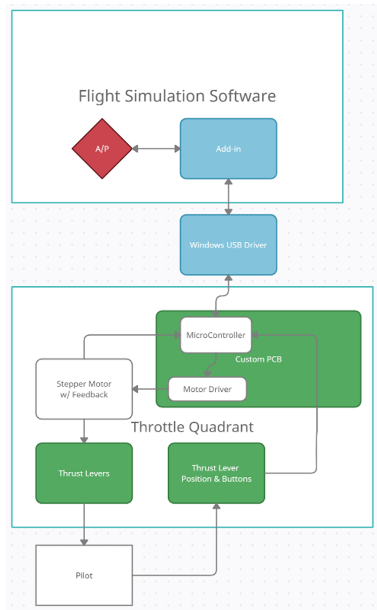

Figure 2: Block diagram of the design.

position.

- If the pilot overrides A/T thrust settings while A/T is engaged, the throttle quadrant must unlock the thrust levers immediately to allow for free inputs. The movement of the thrust levers from then on must be unhindered until A/T is engaged again.

2 Design

2.1 Block Diagram

Figure 2 is an overall view of our design. Our design is divided into two parts that will work in tandem to achieve the requirements listed above. The software part includes an add-in for select flight simulation software and an USB driver. The add-in will gather airplane status information from APIs provided by the flight simulation software. The USB driver will allow bidirectional communication to be established between the PC and the throttle quadrant. The hardware part includes a custom control board carrying a microcontroller and a stepper-motor driver chip. The control board will translate between thrust lever position and throttle input. Stepper motors with feedback enabled will provide accurate movement of the thrust levers and slip detection for override detection.

2.2 Functional Overview

2.2.1 Software

A/P. The A/P system is provided by the flight simulation software with well-documented APIs allowing access to aircraft status data and control inputs. This block will be the software endpoint of our system.

Flight simulator add-in. The add-in block will be written to be compatible with select flight simulation software. It will be responsible for gathering flight simulation data from the A/P system API as well as transmitting pilot input into the flight simulation software for interpretation. It must be able to achieve continuous, real-time bidirectional communication.

Windows USB driver. The USB driver will allow the Windows 10 operating system to recognize our hardware as a valid peripheral connected to the USB bus. All data and signal traffic to and from the software add-in will be handled by the USB driver. Once installed on the host PC, the USB driver should enable PnP for the throttle peripheral.

2.2.2 Hardware

Microcontroller. The microcontroller will be the processing center on the peripheral side. Connected to the host PC via USB, the microcontroller will interface with the USB driver to deliver and collect necessary data. The microcontroller will also be responsible for managing control signals to instruct movements of the motors, respond to button presses, and decoding thrust lever positions. This will likely to be an ATMEGA microcontroller running custom codes.

Motor driver. The motor driver manages power and control signals for the stepper motors. Special stepper motor drivers are necessary due to the special construction of stepper motors and the unique requirements to run them. This will likely to be an aftermarket component.

Stepper motors. Three stepper motors will be used to physically move the levers. Each stepper motor will be responsible for only one lever. The stepper motors will be powered externally by a power adapter and controlled by the stepper motor driver. They will remain unpowered unless an instruction was received to move. The stepper motors will have closed feedback loops for accurate determination of the rotors' positions.

Levers and buttons. The thrusts levers and buttons are the main user interface. They will be fixed on one end with a hard-limited travel range. Rotation of the thrust levers about the fixed end will correspond to adjustments to the thrust setting. The travel range and thrust mapping will be as close to a Boeing 737-800 as possible for realism. Buttons will be attached to the top of both thrust levers and one on the left thrust lever. These button presses will be handled by the microcontroller and be mapped to corresponding functionalities found on the Boeing 737-800.

2.3 Block Requirements

Software subsystem. The software subsystem consists of the simulation software's A/P functionalities, an add-in for data gathering and virtual inputs, and an USB driver for device registration on the USB bus. The software subsystem must be capable of constant polling from both the simulation software and the throttle peripheral to ensure continuous and immediate response once any change is made.

Hardware subsystem. The hardware subsystem consists of a microcontroller, a stepper motor driver unit, three stepper motors, and the user interfaces (levers and buttons). The hardware subsystem should provide the pilot with a firm but effortless operational feel as close to that of real airplanes as possible. The levers must be able to move at a constant rate smoothly whenever A/T commands a thrust adjustment. The levers must stop exactly and firmly at the designated position without slipping or drifting. The motors should never impose opposing torque to pilot inputs and must report any override to the software subsystem.

2.4 Risk Analysis

The stepper motor block is identified as the "X-factor" of our design. It is the only true mechanical design block of the system and is crucial to the fulfillment of all high-level requirements. This block is chosen as the one with the most challenge as all three of our team members do not have abundant experiences with stepper motor applications. We are concerned about being able to correctly start and stop the motors with desired movement characteristics. Since the stepper motors will be a part of the pilot-throttle interface, we must consider extreme cases where the motors encounter resistance imposed by foreign objects jamming the mechanism or the pilot physically preventing the levers from moving. Our design must be able to stop the motor from overheating or injuring the pilot should these cases occur. Another potential challenge is sharpening the motors' response time for accurate motion during extreme aircraft handling situations such as windshears and turbulances.

3 Ethics and Safety

Our throttle quadrant aims to provide an actual and smooth experience to the audience of flight simulations, we believe that our product can bring a more authentic and affordable option to the publicity, which facilitates the wide audience of flight enthusiasts to access actual flight experience, and may serve as training equipment before they step up to planes. Thus our design aligns with the IEEE Code of Ethics Section 7.8.2: 'to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems' [1].

Since our design has moving components and 5V/24V power supplies, both of them could be potential safety threats towards users, also the misuse of our design may cause undesired consequences to the user and the peripheral itself. The moving thrust lever during A/T may crush the user, and the power leakage may cause significant damage as current passes through the user. To optimally be following the IEEE Code of Ethics Section 7.8.9, striving to reduce potential harm to the user and the property, we make efforts to optimize our design [1]. To address the safety concerns about the moving components, we control the torque of the motor by limiting the passing-through current in the microcontroller. Moreover, we use a strengthened frame to limit the position of the lever and prevent the user from reaching dangerous locations. To address the concern regarding the electric shock, the outer shell of our design will be made from insulating materials to isolate the user from the electrical components as well as to prevent external items from entering the throttle quadrant and damage the internal components.

References

[1] IEEE, "IEEE Code of Ethics," 2020, IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 18 Feb 2021].