Classical Control Based Autopilot Design Using PC/104

Mohammed A. Elsadig,
*Alneelain University, Dr. Mohammed A. Hussien, Alneelain University.*

**Abstract**—Many recent papers have been written in unmanned aerial systems (UAS) autopilot designing for fixed wing aircraft, in addition to vertical takeoff and landing aerial vehicles such as helicopter and quad-copters. The process of designing such a system needs great facilities, great cost, and has a significant risks of crashes. This paper introduces PID-based (proportional, integral, and derivative) autopilot design using Matlab/Simulink modeling to reduce design cost and time; in addition to minimizing risks. The PC/104 board was used as the hardware for the system. This paper assured the validity, simplicity, and time efficient of PID-based design, and the significant reduction of cost and effort in autopilot design using the Software in Loop (SILS) and Hardware in Loop simulations (HILS).

**Indexed terms**—PC/104, HILS, SILS, PID controller, fixed wings aircraft.

**I. INTRODUCTION**

Unmanned aircraft or (UAS) is that it is an aircraft with its aircrew removed and replaced by a computer system and a radio-link [1]. After many “UAV” concepts were implemented in the ancient history the first successful implementation of such a system in the modern history was done in the First World War at 1917 of “aerial torpedo” of US Navy [2]. Others consider that it is the British army who invented the first UAS at 1914 by “aerial Target” [1].

Civilian applications of UAS include but not excluded to: surveillance, agricultural imaging, mapping, border patrol, search and rescue Pollution monitoring, traffic monitoring, and many others [2].

While military applications include: Relaying Radio Signals, Surveillance of enemy activity, Target detection, and many others [1].

For newcomers to the field of unmanned aircrafts it may be confusing with the terms: UAV and UAS; while sometimes they are used to give the same meaning [3], other times they used to give different meanings [1]. In this research UAS is used to indicate the combination of Aircraft which is called UAV, the Avionics, the telecommunication system, and the monitoring station in ground. In fact there are many names for unmanned aircraft systems such as: not crewed aerial vehicle, unmanned autonomous vehicle, unmanned airborne vehicle and others [3]; but in this research just UAV and UAS will be used to clear any confusion.

UAS mainly consists of the following systems:

1- UAV; which includes airframe, engine, and propeller.

2- Avionics system; which includes Autopilot including all necessary sensors, telecommunication system, batteries, and servomotors.

3- GCS; which includes PC, GUI package for telemetry visualizing, telecommunication system, RC remote control device, and it may exceed those systems to include video and image processing and visualizing unit, payload control unit.
UAS can be categorized in many bases; for example UAS can be classified based on airframe configuration as:

1- Rotary aircrafts: then this again can be categorized as: helicopter, quad-copter, and octa-copter based on the numbers of fans, this type also can be named VTOL or Vertical takeoff and landing vehicles.

2- Fixed wing aircrafts: then this again can be categorized as: conventional and unconventional airframes, this type can be named HTOL or Horizontal takeoff and landing vehicles.

There is a third airframe configuration which is the hybrid configuration in which both the rotor and wings are used, but it is rarely used.

Autopilot system is the electronic device which is programmed to control and guide the aircraft instead of the human; and it usually consists of the following:

1- Processing unit: this can be a microcontroller or microprocessor SOC.

2- Flash memory: to save the control code.

3- RAM memory and registers which are used during execution of the program.

4- ADC either embedded into the microcontroller or used as a standalone IC with the main processor to deal with the analog sensors.

5- Gyroscope sensor to measure attitude of the aircraft.

6- GPS to give position, heading and other data to the processor.

7- Accelerometer to measure airspeed.

8- Altimeter sensor to measure altitude.

Autopilot is expected to read the sensors data such as GPS and Gyroscope in real-time in order to get the attitude of the vehicle; then it is supposed to send correction commands to drive servo motors which control the UAV airframe motion [3].

Autopilot design process plays an essential role in Unmanned Aerial vehicle industry, this process normally takes a significant amount of time and cost, the autopilot design process involves algorithm design. The second process is the validation of this algorithm in term of control; this validation process normally is done using fly tests especially if the control algorithm was PID controller based which is a trial and error process. The previous approach of validation obviously would take long time in validation of logic and implementation, in addition to risks of crashes which if happened would cost much.

This paper explains a PID-based autopilot design process for fixed wings aircraft using Matlab and Simulink libraries such as Aerosim Blockset in order to validate the system function and stability by software and hardware simulation processes which will reduce the time, effort, and cost significantly.
II. METHODOLOGY

The process of designing the system was done through multiple steps; firstly the model of the airframe was made into a Simulink file; this is called open loop design this step is out of the research scope, so ready fixed wing airframe coefficients were used to develop the open loop simulation. Secondly, the open loop should be closed using PID controller from the Simulink library and the Matlab automatic tuner tool was used to get the PID coefficients. Thirdly the closed loop logic was translated into a C code in order to do Program in loop test. Fourthly the C code was interfaced with the aircraft model in the Simulink file to control the airframe; this is called Software in Loop. Fifthly the PC104 board was programmed with the C code and interfaced with the open loop Simulink file to control airframe through PC104; this is called Hardware in Loop test. Finally, PC104 was interfaced with Attitude and heading reference system (AHRS), and GPS sensors, to read the required data for flight, in addition to that PC104, was programmed to generate PWM signals according to the control algorithm commands in order to control the servo motors which control the aircraft motion.

![Fig. 2. Open loop demonstration [4]](image)

A. Modeling the airframe in Matlab/Simulink

The Simulink library contains Aerosim Blockset library which contains a complete aerodynamic model, this model needs the aerodynamic coefficients of the specific airframe which should be entered to through a (.mat) file. Airframe modeling software packages used to model the airframe and get the desired coefficients and enter it to the (.mat) file. The open loop model contains actuators as inputs and sensors data as output, test should be done using a simple joystick which controls the actuators directly; open loop demonstration is shown in the figure below. FlightGear simulator was used to show the airframe motion in a 3D graphics after it was interfaced with the Simulink model through UDP socket.

B. Design the Closed Loop model in Simulink

PID controller was applied to control the model the next figure shows how it was designed, sensors was put as the feedback to the controller and joystick command as the command input. The Error was get after subtracting command from the feedback, a PID controller was applied to the resulted error then the result is used to control the actuator.

The design of the control algorithm was done in three steps:

1. Stability level: which contained designing roll damping function to control airframe in the roll axis, pitch damping function to control airframe in Pitch axis, and yaw damping to control airframe in...
Yaw axis, in addition to that Throttle function was designed to control airspeed. The next figure shows how these damping functions were designed.

![PID-based closed loop model](image)

**Fig. 3. PID-based closed loop model [4]**

2- **Control level**: contained designing Altitude control function uses pitch damper and throttle functions. Heading control function which uses roll damper and yaw damper functions.

3- **Navigation level**: all previously mentioned functions were used to control the flight of the airframe in a predefined route. Two functions for calculating heading and distance were designed to give command to both altitude control and heading functions. This level was done in SILS instead of closed loop model due to the simplicity of the developing in the SILS.

**C. Software in Loop Simulation**

The PID control algorithm which was designed in the Simulink was replaced by a C code. CodeBlocks software was used to write the code and interface it with the Simulink model through virtual serial connection; this test is named Software in Loop simulation or (SILS). The next figure shows how to do SIL test.

**D. Hardware in Loop Simulation**

The PC104 board was programmed using the previous C code with the aid of both DOS operating system and Turbo C as a cross compiler. PC104 board was programmed to interface with the model in Simulink through physical serial Ports of the Computer. The next figure shows how HILS block diagram. The selection of the PC/104 board was due to its high capabilities in term of clocking rate which reaches from 300MHz up to 800MHz in recent versions, another advantages of PC/104 are number of communication serial ports, GPIO and PWM signals for actuator commanding, USB port for data transferring, VGA port for screen interfacing, SPI port for mouse and keyboard.

![HILS block diagram](image)

**Fig. 4. HILS block diagram**
PC/104 board components is shown in fig. 5, the components are not limited to serial communication ports, GPOIs, VGA, LAN, and others.

![PC/104 board components](image)

**Fig. 5. PC/104 board [5]**

### III. RESULTS AND DISCUSSION

1. **SILS test results**
   After interfacing the Codeblocks C code with the aircraft model in Matlab\Simulink a satisfied stability and control levels were reached with a bearable response time the stability

![SILS roll stability](image)

**Fig. 6. SILS roll stability**

graphs for lateral and longitudinal axis are shown in the figures figure 6 and figure 7.

The sudden drop of roll angle in the middle of time is due to the change of pitch angle at the same time. A 10 degrees was applied to both roll and pitch control functions.

![SILS pitch stability](image)

**Fig. 7. SILS pitch stability**
For airspeed function an airspeed of 120km/h was applied to the control loop; result can be seen in figure 8. A zero degree command was applied to the heading control function, result can be seen in figure 9. An altitude of 1400 meters was applied to the altitude holding function, result is shown in figure 10.

![Fig. 8. SILS airspeed control](image1)

![Fig. 9. SILS heading control](image2)

![Fig. 10. SILS altitude control](image3)

2. **HILS test results**

After building the logic of the C Code into the PC104 and interfacing it with the Matlab\Simulink model physically same stability level was reached with a similar response time with a little modification in the PID controller coefficients; the stability graphs for lateral and longitudinal axis are shown in the figures below.
Fig. 11. HILS roll stability

Fig. 12. HILS pitch stability

Fig. 13. HILS airspeed control

Fig. 14. HILS heading control

Fig. 15. HILS altitude control
The stability functions of roll and pitch were tested by applying zero degree for each of them and the results are shown figure 11, and figure 12.

The airspeed control function was tested by applying 140 km/h to the control loop, the result is shown in figure 13.

Heading control function was tested by applying 100 degrees to the control loop, the result is shown in figure 14.

Altitude control function was tested by applying 1400 meters height to the control loop.

Navigation mode is the mode in which the system ability to follow a pre-specified route was tested successfully: four waypoints were given to the system and the system leaded the aircraft to track the route successfully and came back to the home waypoint. The four waypoints of the route is shown in numbers in each of the figure 17 and figure 18.

The system both in SILS, and HILS followed the route correctly and pass all the waypoints and came back to the starting waypoint. The reaching distance to the waypoint was selected to be 200 meters to smoothing the movement of the system along the route, the followed route both in SILS and HILS is shown in figure 17, and figure 18 respectively.

There is shifting of the system from the straight line of the route that can be due to the imperfect PID tuning of the control loops or due to the simplicity of the navigation algorithm.

IV. CONCLUSION

An accurate, efficient, low cost, real time response autopilot system was designed based on PID controller using the PC104 board with the aid of MatlabSimulink modeling, and a low cost, accurate, real time lab test methodology was verified using the previously mentioned methodology of HILS.

The accuracy of the system can be increased if additional PID tuning was done, and if more data was taking into consideration in the algorithm.
ACKNOWLEDGMENT

Thanks to Alneelain University, especially postgraduate studies faculty for their support, thanks to our colleagues for their significant help to make this research valid.

REFERENCES

1. Reg Austin, Unmanned Aircraft Systems, Wiley press, 2011.
2. Kimon P. Valavanis, Advances in unmanned Aerial Vehicle, Springer press, 2007.
3. Rafael Yanushevesky, Guidance of unmanned aerial vehicles, CRC press, 2011.
4. Unmanned dynamics, Aerosim aeronautical simulation blockset v1.2 user guide.
5. VDX-5364 user manual, revision 1.1A, February 2009.