The Workings of Autopilot

Tomer Polak

Abstract:

Autopilot is a tool that almost every flight uses. Virtually every modern aircraft has some level of automation and autopilot. Much research has been done on different varieties of autopilot and different effects it has on flights. Studies have shown that autopilot greatly increases the efficiency and confidence of the pilots in the plane and pilots feel safer knowing they have a button that they can press which will allow the plane to almost fly itself. However, most pilots don't know what happens when they press that magical "AP" button, and this paper aims to explain how autopilot and aircraft automation work.

1. Introduction:

Every day, about 130,000 planes are in the sky using Autopilot[1]. An autopilot is a tool that almost every pilot uses to create a safer and more efficient flight. Since its inception, this tool has been meticulously developed and tested to create more advanced systems. From the early 1900s to today, autopilots have continuously developed. The Wright Flyer first flew in 1903[2], and by 1912, the Sperry Corporation developed an autopilot that connected a gyroscopic heading indicator and an attitude indicator to a hydraulically operated rudder and elevators. This autopilot had no roll authority and could keep the plane on its current trajectory; it utilized wing dihedral, the angle between the left and right wing, to maintain level flight[3].This Autopilot system was further developed and used in 1930 to keep Army Air Corps aircraft on their heading and at their altitudes for three hours[4]. In 1947, different autopilot systems emerged with more advanced features, such as the ability to use radio navigation, which allows aircraft to operate in IMC conditions and at night[5].

Furthermore, automation has progressed significantly since the mid-1900s. Fly-by-wire is one of the most outstanding achievements in this space. Fly-by-wire eliminates mechanical and hydraulic links directly from the pilot to the control surfaces; the control inputs are fed into a computer. The computer then processes the input and decides which flight controls to move to get the desired effect. The computer also has tremendous safety benefits as it ensures the pilot does not overstress the aircraft; it can veto any input to take the plane out of its performance envelope[6]. At first, flyby-wire was used with a mechanical backup in aircraft like the Concorde, but in 1973[7], the F16 became the first aircraft to use flyby-wire for all its flight controls with no mechanical backup. The F16's fly-by-wire system is also why the aircraft is so agile. The F16 is aerodynamically unstable, meaning the plane would lose control if the pilot and computer made no control inputs. The fly-by-wire system constantly makes small inputs to the control surfaces to keep the aircraft flying[8]. This technology allows for unique airframe designs that serve different purposes and further the aerospace industry.

Modern autopilots have evolved into awe-inspiring tools, now a standard feature in nearly every aircraft, from large airliners to small Cessnas. These autopilots can assume full control of the plane from 100 ft off the ground after takeoff up to a quarter mile from landing, demonstrating their versatility and precision. Larger jets can execute category 3 approaches, a level of automation that allows the plane to land itself and stop on the runway, underscoring the advanced capabilities of modern autopilots[9]. Even smaller general aviation aircraft are equipped with autopilots, such as the GFC 700 by Garmin[10], which features electronic stability protection. This system ensures that when the aircraft is outside of its performance envelope, the autopilot servos automatically kick in and deflect the flight controls to restore the plane to a stable flight[11]. In military aviation, this technology plays a crucial role in preventing aircraft from impacting the surface if the pilot has lost consciousness due to excessive G forces. All of these technologies have been shown to decrease pilot workload and make flying safer[12].

Moreover, a new system is being implemented and developed for select general aviation aircraft. This system is a button that passengers can press, prompting the computer to land the plane. This safety feature is revolutionary because it makes single-plot operations significantly safer. If a pilot becomes incapacitated, then any passenger can simply press the emergency Autoland button, and the computer will pick the closest suitable airport and control everything, including air traffic control communications, until the aircraft is at a complete stop on the runway with its engines turned off[13].

Lastly, autopilots are constantly improving. Companies like X-Wing are experimenting with fully autonomous planes that can perform gate-to-gate operations without pilot input[14]. Students at MIT created an algorithm that automatically guides an aircraft through canyons and tight spaces, like the scene from Top Gun[15]. Eventually, autopilots will be so advanced that commercial and cargo flights in massive airliners require only one pilot. New features such as emergency takeover could immediately take control of the aircraft in an emergency, eliminating the startle factor. The startle factor is the about 5-second amount of time pilots take to diagnose and figure out how to deal with the situation. A computer could instantly take control and find the safest solution, saving countless lives each year.

2. Methodology:

a. Open-loop System

Before designing the autopilot, the dynamics of the system need to be set. An example of why this is important comes from the Boeing 737 Max incidents. The engineers took the older version 737 and added larger engines. This changed the center of gravity of the aircraft. To counter this change Boeing implemented the Maneuvering Characteristics Augmentation

System (MCAS). This system was meant to pitch the aircraft down to prevent an excessive angle of attack. However, this system was flawed. Had the engineers taken the time to redesign the 737 to fit larger engines and not just add larger engines to an old version and try to remedy the problem with software, the 737max story would be different.

b. Spring Mass Damper Closed loop system

The spring mass damper is a good place to start because of its relatively simple nature. To start the system needs to be defined in terms of equations and variables. First, the three main variables for the system. M is the Mass, K is the spring coefficient which parameterizes the amount of force exerted by the spring due to a change in position, and C is the damping Coefficient which parameterizes force of the damper. Mass is relatively simple, the spring resists changes in position, and the damper resists changes in velocity.

To write the system Newton's second law (F=ma) is used, in which is force equals mass times acceleration. F and a can be vectors because both force and acceleration have direction and magnitude. Then start to substitute more in-depth variables into the equation. x is position, and $\dot{x}$ is velocity. Velocity is described by taking a change in position and dividing it by a change in time. Instantaneous velocity can be computed by using a very small change in time. This is important because if the distance and time are large values then the velocity is just the average and does not account for any variation that occurred. Additionally, angular velocity can be calculated using gyros.

$$\frac{\Delta x}{\Delta t} = \dot{x}$$

Another important concept is acceleration, which is a change in velocity divided by a change in time. This variable is expressed with $\ddot{x}$ and the units meters per second per second.

$$\frac{\Delta \dot{x}}{\Delta t} = \ddot{x}$$

So then because the spring resists changes in position we can take the spring variable K and make it Kx. F=Kx

Then with the damping coefficient C we can make it $C\dot{x}$ because it resists changes in velocity. F= $C\dot{x}$

Both the spring and damper exert forces, so we can substitute newton's second law to F-F=Ma so $Kx+C\dot{x}=M\ddot{x}$

This equation is simply Newton's second law in terms of our system.

The goal of this simulation is to take a change in force, then convert that change into acceleration using the equation. To do this we need to create a state vector, which is a vector of position and velocity.

$$\vec{x} = \frac{x}{\dot{x}}$$

This vector functions in the state space which is a graph that has position on the x-axis and velocity on the y-axis. The vector meets the state trajectory which is a curve that goes through the state space. The place where the state vector interests with the space trajectory is the answer for one set of position and velocity.

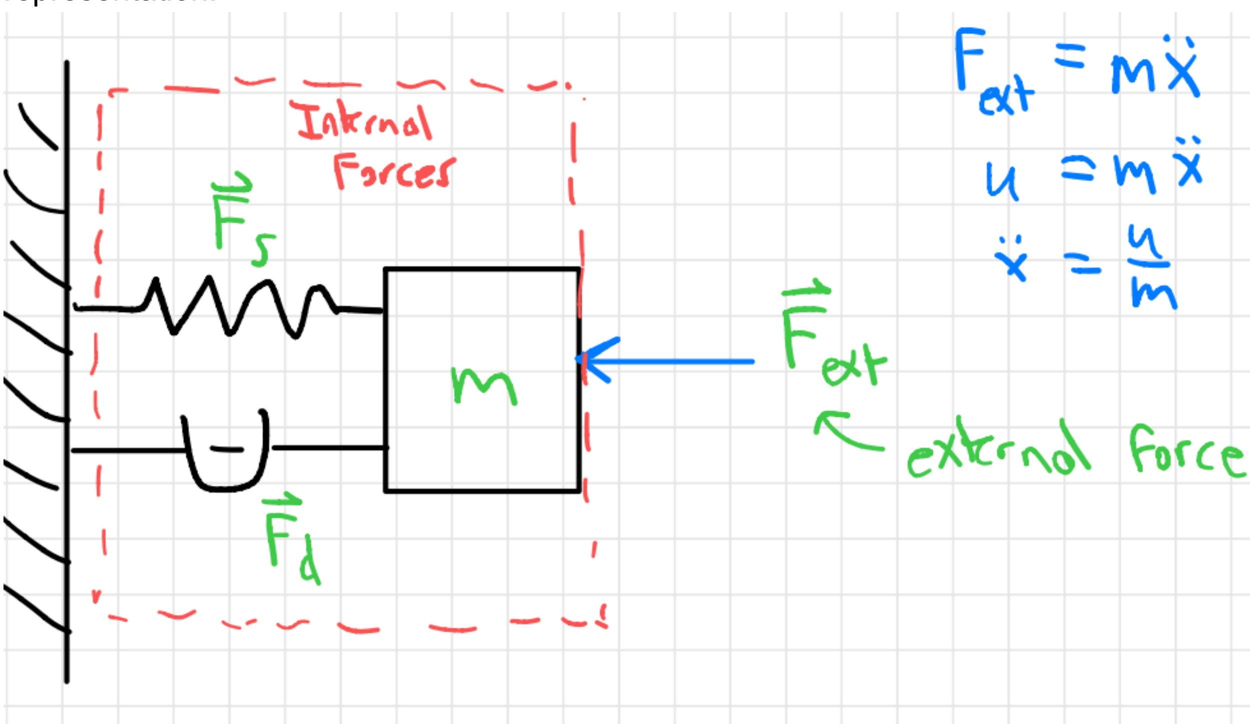
To place our system in the state space we need to write our equation in state space form, which is ẋ= Ax. This form is a compact way to represent a dynamic system. A is a matrix in which the parameters of the system are placed, and the position and velocity variables are the same. To do this take the state space equation : $\overrightarrow{Kx} + \overrightarrow{C\dot{x}} = M\vec{\ddot{x}}$.
Then define a few more equations:

$\vec{\dot{x}} = \frac{\dot{x}}{\ddot{x}}$ This is the velocity of the state vector

$\vec{x} = \frac{x}{\dot{x}}$ This is the rate of change of the state vector

$\ddot{x} = \frac{k}{m}x + \frac{c}{m}\dot{x}$ and this is the state space equation divided by mass.

These equations are then placed in a matrix using element wise multiplication to define the system: $\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -k/m & -c/m\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix}$. This equation is more compact for the computer simulation, and is how the system's dynamics are understood by the computer. Here is a visual representation:



Finally, the closed loop control can be incorporated. The most effective way of doing this is with proportional integral derivative control (PID). This method of closed loop the control is simple and effective. Additionally, in most cases a simpler PD controller can stabilize the system. The PID controller can be simply added to the existing system. The current system is expressed as:

ẋ= Ax+Bu.

B is a simple matrix of $\begin{bmatrix} 0 \\ 1/m \end{bmatrix}$. B is the control input matrix, and it affects way force impacts the system. U is a force exerted by a human on the system. The objective of the closed loop system is to have force U generated by the system automatically by a computer, and then have the computer activate a motor which then exerts force U on the system. To have this happen U needs to be a function of system state and this is where PD control comes in. U is made a function of system sates by having the sum of force the Proportional and derivative force made equal to it.

$u = k_P x + k_D \dot{x}$.

Now U needs to be restated in matrix form, which is done using element wise multiplication:

$$u = \begin{bmatrix} k_P & k_D \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = k_P x + k_D \dot{x}$$

Now from here U is incorporated into the existing system:

$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \begin{bmatrix} k_P & k_D \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$.

From here $\begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ is factored out to get:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \left(\begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \begin{bmatrix} k_P & k_D \end{bmatrix} \right) \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

Now the three matrices are combined using element wise multiplication:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} + \frac{k_P}{m} & -\frac{c}{m} + \frac{k_D}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

This matrix can be abbreviated as A_{cl} because the PD controller embedded in the matrix creates a closed loop control system. This whole system can be simply rewritten as:

$$\dot{\vec{x}} = A_{cl} \vec{x}$$

This equation will be the backbone of the automation with the f16 simulation as the parameters of the system will be changed to match the dynamics of an F16 rather than a spring mass damper The variables k_P and k_D can be changed to create the desired outcome for the system.

c. Euler's Method of numerical integration

With the equation in state space form the next step is to have the simulation calculate position and velocity data at discrete time steps. When using Euler's method of integration, the computer only knows what is happening at specific predetermined time steps. At time T=K which is the beginning of the simulation. $\dot{X}_k = \frac{x_k}{t_k}$. This means that the velocity at K is equal to the change in position at K divided by the change in time at K which is simply the equation for velocity. Euler's method is $\Delta x_k = \dot{X}_k t_k$. Now with this equation is substituted into the state space equation: $X_{k+1} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x_k t_k$. This equation is the backbone of the simulation and changing the matrix A can make it applicable to any system. In this case it contains the parameters of a spring mass damper, but later in the paper it will contain F16 flight dynamics.

d. Designing a Stable Controller

In designing a controller, it is critical to ensure that the controller can keep the system stable. To do this Eigen values need to be considered. In the linear case of $\dot{\vec{x}} = A_{cl}\vec{x}$ the state space is defined by its eigen vectors $\vec{v}_i$, and each vector has its eigen value λ_i. In the f16 there are 13 variables, which means 13 vectors. Each vector has an associated eigen value. In a stable system all system vectors approach the origin of the state space. To achieve this all eigen vales must be negative.

3. F16 simulation

This F16 simulation is from the *AIRCRAFT CONTROL AND SIMULATION* Third Edition by BRIAN L. STEVENS, FRANK L. LEWIS, AND ERIC N. JOHNSON[16]. This simulation is complete F16 which is an incredibly complex and involved system with 4 input variables and 13 state variables. Because of this complexion, building a controller that functions at any system state is virtually impossible. This in in part due to the inherent behavior of the system which is nonlinear and the solution to this is linearizing the system around a specific trim condition. This means that out of the nonlinear equation for the complex system, a linear equation is made around a specific trim condition which allows for more careful analysis of the system and its variables. This is like solving for x in algebra. In some equation types such as a line, solving for x is simple, but in larger more involved equations, solving for x could be more difficult. Mathematically this looks like:

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) \quad \Rightarrow \quad \dot{\vec{x}} = A\vec{x} + B\vec{u}$$

In the linearized F16 model, $\vec{x}$ is a 13x1 vector, A is a 13x13 matrix, B is a 13x4 matrix, and $\vec{u}$ is a 4x1vector. Defining the vectors and their variables:

$\vec{x}$ = [v_t, α, β, $\varnothing$, θ, Ψ, P, Q, R, N, E, D, P]

v_t : relative forward velocity (ft/sec)
α: angle of attack (rad)
β: sideslip angle (rad)
$\varnothing$: roll angle (rad)
θ: pull up angle (rad)
Ψ: turn angle (rad)
P: roll rate (rad/sec)

Q: Pitch rate (rad/sec)
R: yaw rate (rad/sec)
N: north speed (ft/sec)
E: east speed (ft/sec)
D: vertical speed (ft/sec)

Additionally input vector $\vec{u}$ = $[T, \delta_a, \delta_e, \delta_r]$ where

T: throttle (0-1)
δ_a: aileron (deg.)
δ_e: elevator (deg.)
δ_r: rudder (deg.)

a. F16 control

With the F16 simulation explained, controlling the simulation is next. There are many different options when it comes to choosing controller, and what they aim to do. One option is choosing a controller that optimizes something such as fuel consumption, time, or some other variable. Another option is an adaptive controller which is used when the underlying system is changing. This type if controller requires gain scheduling which is when the control gains change depending on the condition. A relevant example is changing the gains in a fly by wire system to account for icing on the aircraft's frame. The last option is robust control, which is the safest because constant gains are chosen to ensure that the objectives are met consistently. However, this variation of control is difficult to design due to model uncertainty. The real world is almost always different from the models used to design controller, and the real world is constantly changing.

In the F16 case there are a lot of options, but the main two are a full state feedback controller or a controller that does one variable such as pitch, and leaves the rest to the human pilot inside. Both cases involve the concept of a feedback loop. This is where a sensor such as a pitot tube or angle sensors, feed information into the A_{cl} model and the model then changes one of the input variables such as throttle or ailerons.

Choosing how to model A_{cl} and the control gains is another variable to consider. Questions such as is the full nonlinear system necessary ($\dot{\vec{x}} = f(\vec{x}, \vec{u})$, or can a linear $\dot{\vec{x}} = A\vec{x} + B\vec{u}$ work? If the full nonlinear system is necessary, is it possible to get away with controlling only one variable such as angle of attack and leaving the rest to the human pilot. Using the full nonlinear system and designing a full state feedback controller, where every state is measured and fed to every actuator would result in a closed loop system which looks like : $\dot{\vec{x}} = A_{cl}\vec{x} = (A - BK)\vec{x}$, where K is a 13x4 gain matrix which correlates every system state to every actuator. Yet, this model for controlling the aircraft is rarely practical due to a variety of reasons. First, it is not always possible to measure every system state, and secondly, in most cases the objective of the controller would not require control of every system state, so there is no reason to overcomplicate the control system.

Therefore, in this F16 simulation, the focus will be on controlling only one or two system states. A very common form of autopilot is Pitch-Axis Stability Augmentation (SAS). This form of control uses angle of attack and pitch rate data from sensors to regulate the pitch angle. The input for this system is elevator deflection (δ_e) in degrees, and the two outputs are angle of attack (α) in radians, and pitch rate (Q) in radians per second. For this model the linear model of $\dot{\vec{x}} = A\vec{x} + B\vec{u}$, and the output vector of $y = C\vec{x}$ is used. Where $\vec{x}$=[v_t, α, θ, q] , $\vec{u}$=[δ_e], A is a 4x4 matrix, B is a 4x1 matrix, and C is a 2x4 matrix. With this model, the gains for angle of attack, and pitch rate can be selected to achieve the desired close looped performance. The final equation for the control law is $\delta_e = K_\alpha \alpha + K_q Q$.

4. Conclusion:

In conclusion, autopilot systems have revolutionized modern aviation, transforming how pilots navigate and manage flights. From their early development in the 20th century to today's advanced, integrated systems, autopilots have consistently enhanced safety, efficiency, and pilot confidence. However, while these systems reduce the workload and stress for pilots, understanding the intricate workings behind the "magic button" remains essential for ensuring their effective use. As technology continues to advance, the aviation industry must balance innovation with pilot training to foster a deeper understanding of automation and maintain the high standards of safety that define modern air travel.

Sources

[1]Flightradar24, “Live Flight Tracker - Real-Time Flight Tracker Map | Flightradar24,” Flightradar24, 2020. https://www.flightradar24.com/data/statistics

[2]National Air and Space Museum, “The Wright Brothers,” airandspace.si.edu. https://airandspace.si.edu/explore/stories/wright-brothers

[3]“Sperry Instrumentation: Shifting to Autopilot,” Lockheed Martin, 2018. https://www.lockheedmartin.com/en-us/news/features/history/sperry.html

[4]W. Scheck, “Lawrence Sperry: Genius on autopilot,” Army Times, Nov. 28, 2017. https://www.armytimes.com/veterans/2017/11/21/lawrence-sperry-genius-on-autopilot/ (accessed May 31, 2024).

[5]M. C. Elish and T. Hwang, “Praise the Machine! Punish the Human! The Contradictory History of Accountability in Automated Aviation,” SSRN Electronic Journal, 2015, doi: https://doi.org/10.2139/ssrn.2720477.

[6]. Nicolin and A. Nicolin, "The fly-by-wire system," *INCAS Bulletin*, vol. 11, pp. 217-222, 2019, doi: 10.13111/2066-8201.2019.11.4.19.

[7]“Concorde Fly by Wire,” heritage-concorde, 2016. https://www.heritageconcorde.com/fly-by-wire

[8]C. Anderson, “DEVELOPMENT OF AN ACTIVE FLY-BY-WIRE FLIGHT CONTROL SYSTEM.” Available: https://ntrs.nasa.gov/api/citations/19760024050/downloads/19760024050.pdf

[9]R. Tan and Z. Yao, “Research on Monte Carlo simulation of CAT III auto-landing system,” Journal of physics, vol. 2551, no. 1, pp. 012003–012003, Jul. 2023, doi: https://doi.org/10.1088/1742-6596/2551/1/012003.

[10]“Effect of the autopilot on pilot performance during its approaches in general aviation technically advanced aircraft - ProQuest,” www.proquest.com. https://www.proquest.com/openview/345c8bd88a73e2ba4b1d6afc2bece7d6/1?pq-origsite=gscholar&cbl=18750 (accessed May 31, 2024).

[11]“Understanding Garmin Electronic Stability and Protection (ESP),” Garmin Blog, Jan. 25, 2018. https://www.garmin.com/en-US/blog/aviation/understanding-garmin-electronic-stability-protection-esp/

[12]R. Hoh, J. Smith, and D. Hinton, “NASA Contractor i . The Effects of Display and Autopilot Functions on Pilot Workload for I I I Single U Pilot Instrument Flight n+lo,.,-,+;r\,., n-uit: t.31 AI 1~) W~L I ULIVIIS I 1.” Available: https://ntrs.nasa.gov/api/citations/19870012488/downloads/19870012488.pdf

[13]“Emergency Autoland | EASA Community,” www.easa.europa.eu. https://www.easa.europa.eu/community/topics/emergency-autoland-0

[14]“NASA, Xwing to develop autonomous flight standards,” www.aopa.org, Oct. 13, 2022. https://www.aopa.org/news-and-media/all-news/2022/october/13/nasa-xwing-develop-autonomy (accessed May 31, 2024).

[15]“A step toward safe and reliable autopilots for flying,” MIT News | Massachusetts Institute of Technology, Jun. 12, 2023. https://news.mit.edu/2023/safe-and-reliable-autopilots-flying-0612

[16] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation : dynamics, controls design, and autonomous systems*. Hoboken, N.J.: John Wiley & Sons, 2016.