การออกแบบและจำลองอากาศยานไร้คนขับด้วย MATLAB-Simulink

จาตุรนต์ จันทะมาตร[1] จิรศักดิ์ คำโสภา[2] ศิวะโรจน์ บำเพ็ญทาน[2] และ ณัฐพล ใจสำรวม[1, 2*]

วันที่รับ 22 มีนาคม 2568 วันที่แก้ไข 6 พฤษภาคม 2568 วันตอบรับ 21 พฤษภาคม 2568

บทคัดย่อ

อากาศยานไร้คนขับมีบทบาทสำคัญในหลากหลายด้าน เช่น การทำแผนที่ทางอากาศ การติดตามทางการเกษตร การลาดตระเวน และการขนส่ง อย่างไรก็ตาม การพัฒนาและทดสอบระบบควบคุมการบินของอากาศยานไร้คนขับยังคงเป็นความท้าทาย เนื่องจากความซับซ้อนของแบบจำลองพลศาสตร์และความเสี่ยงจากการทดสอบในสภาพแวดล้อมจริง วิธีการแบบดั้งเดิมมักต้องอาศัยการทดสอบภาคสนามที่ใช้ทรัพยากรมากและมีความเสี่ยงสูง งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมควบคุมการบินของอากาศยานแบบควอดโรเตอร์โดยใช้แนวทางการออกแบบตามแบบจำลอง (Model-Based Design) ใน MATLAB-Simulink ร่วมกับเฟิร์มแวร์ PX4 ซึ่งเป็นระบบควบคุมการบินแบบเปิดเผยซอร์สโค้ด เพื่อเพิ่มความแม่นยำและลดความเสี่ยงก่อนใช้งานจริง ได้มีการนำเทคนิค Hardware-in-the-Loop (HITL) มาใช้ในการจำลองการทำงานของระบบในสภาพแวดล้อมเสมือนจริง วิธีการที่นำเสนอในงานนี้ช่วยให้สามารถเชื่อมต่อระหว่างการจำลองกับการใช้งานจริงได้อย่างราบรื่น รองรับฟังก์ชันการนำทางแบบกำหนดจุด (Waypoint Navigation) การปรับแต่งพารามิเตอร์แบบเรียลไทม์ และการวางแผนภารกิจผ่าน QGroundControl ซึ่งช่วยยกระดับความปลอดภัยและประสิทธิภาพของระบบ และสามารถเป็นรากฐานสำหรับการวิจัยต่อยอดในระบบอากาศยานไร้คนขับแบบอัตโนมัติในอนาคต

คำสำคัญ: MATLAB-Simulink PX4, โปรแกรมควบคุมตำแหน่งการบิน, การจำลองการบิน, การออกแบบควบคุมการบิน, การจำลองระบบนำทางอัตโนมัติ, การจำลองฮาร์ดแวร์ในห่วงโซ่ (HITL)

[1] สาขาวิชาเทคโนโลยีเพื่อการพัฒนายั่งยืน, คณะวิทยาศาสตร์และเทคโนโลยี, มหาวิทยาลัยธรรมศาสตร์

[2] หน่วยวิจัยด้านการประยุกต์ใช้เทคโนโลยีและระบบอัตโนมัติในอุตสาหกรรมเพื่อความยั่งยืนแห่งมหาวิทยาลัยธรรมศาสตร์

[*] ผู้แต่ง, อีเมล: nattaponj89@gmail.com, nattaponj@tu.ac.th

The UAV Design and Simulation using MATLAB-Simulink

Chaturon Chantamat[1] Jirasak Kamsopa[2] Siwaroj Bumpenthan[2] and Nattapon Jaisumroum[1, 2*]

Received 22 March 2025, Revised 6 May 2025, Accepted 21 May 2025

Abstract

Unmanned Aerial Vehicles (UAVs) have played a vital role in a wide range of applications such as aerial mapping, agricultural monitoring, surveillance, and logistics. However, the development and testing of UAV flight control systems remain challenging due to the complexity of dynamic modeling and the risks associated with real-world testing. Traditional methods often require extensive field trials, which can be costly, time-consuming, and potentially hazardous. This study aims to develop a reliable and efficient flight control system for quadrotor UAVs by utilizing a Model-Based Design (MBD) approach in MATLAB-Simulink, integrated with the PX4 open-source autopilot firmware. To ensure accuracy and reduce deployment risks, the Hardware-in-the-Loop (HITL) simulation technique was employed, allowing for the real-time validation of control algorithms in a simulated yet realistic environment. The proposed method facilitates a seamless transition from simulation to hardware implementation, supporting features such as waypoint navigation, real-time parameter tuning, and mission planning via QGroundControl. This integrated framework enhances both the safety and effectiveness of UAV development and serves as a foundation for further research in autonomous aerial systems.

Keywords: MATLAB-Simulink PX4, QGroundControl, UAV flight control, UAV flight simulations, Model-based design, Autonomous navigation, Hardware-in-the-loop (HITL)

[1] Department of Sustainable Development Technology, Faculty of Science and Technology, Thammasat University.

[2] Thammasat University Research Unit in Application of Technology and Automation Systems in Industrial for Sustainability

[*] Corresponding author, E-mail: nattaponj89@gmail.com, nattaponj@tu.ac.th

1. บทนำ

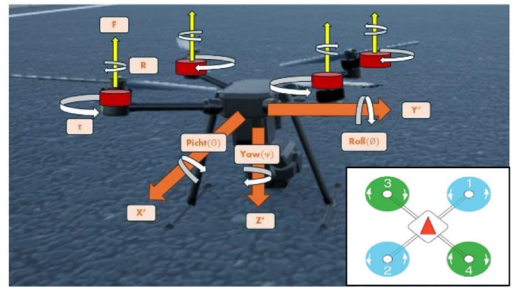
อากาศยานไร้คนขับ (Unmanned Aerial Vehicles: UAVs) และระบบอัตโนมัติก้าวหน้าอย่างรวดเร็ว ได้กลายเป็นหนึ่งในนวัตกรรมที่มีบทบาทสำคัญทั้งในภาคอุตสาหกรรม วิทยาศาสตร์ และงานด้านวิศวกรรมระบบควบคุม โดยเฉพาะอย่างยิ่งในการใช้งานภายในอาคาร ซึ่งจำเป็นต้องอาศัยระบบนำทางที่แม่นยำและความสามารถในการปรับตัวต่อสิ่งแวดล้อมแบบไร้สัญญาณ GPS งานวิจัยของ Bagher Oskooei Fereshte เรื่อง *"Programming of an Indoor Autonomous Drone and Metrological Characterization"* มุ่งเน้นการพัฒนาและประเมินผลระบบควบคุมของโดรนอัตโนมัติภายในอาคาร โดยใช้แนวทางการเขียนโปรแกรมร่วมกับการวิเคราะห์เชิงมาตรวิทยา (Metrological Characterization) เพื่อวัดประสิทธิภาพและความแม่นยำของระบบ

วิทยานิพนธ์ดังกล่าวได้แสดงให้เห็นถึงศักยภาพของโดรนในบริบทของการบินในพื้นที่จำกัด โดยใช้ข้อมูลจากเซนเซอร์และการคำนวณแบบเรียลไทม์ในการรักษาท่าทาง (Attitude) และตำแหน่งของโดรนได้อย่างเสถียร จุดเด่นอยู่ที่การบูรณาการระหว่างด้านวิศวกรรมซอฟต์แวร์และฮาร์ดแวร์ควบคุม ซึ่งช่วยเปิดทางให้กับการนำไปใช้งานในด้านต่าง ๆ เช่น การตรวจสอบภายในโรงงาน การเก็บข้อมูลในพื้นที่เสี่ยง หรือการสนับสนุนระบบอัตโนมัติในอุตสาหกรรม 4.0 [1]

ปัญหาหลักที่พบในการพัฒนาระบบควบคุม UAV คือ ความยุ่งยากในการสร้างแบบจำลองพลศาสตร์ที่ใกล้เคียงกับความเป็นจริง การทดสอบภาคสนามมีค่าใช้จ่ายสูงและมีความเสี่ยงต่ออุปกรณ์ รวมถึงกระบวนการพัฒนาที่ขาดความยืดหยุ่นในการปรับแต่งและตรวจสอบอัลกอริทึมควบคุม

เพื่อลดปัญหาดังกล่าว งานวิจัยนี้จึงมุ่งเน้นการพัฒนาและทดสอบระบบควบคุมการบินของ UAV โดยใช้แนวทางการออกแบบตามแบบจำลอง (Model-Based Design) บนแพลตฟอร์ม MATLAB-Simulink ซึ่งทำงานร่วมกับเฟิร์มแวร์ PX4 และระบบ Hardware-in-the-Loop (HITL) ที่ช่วยให้สามารถจำลองการบินในสภาพแวดล้อมเสมือนจริงได้อย่างแม่นยำ โดยการพัฒนาอัลกอริทึมควบคุมที่สามารถนำไปใช้ได้จริง มีเสถียรภาพ รองรับการนำทางแบบอัตโนมัติ และสามารถปรับแต่งได้อย่างยืดหยุ่นผ่าน QGroundControl ซึ่งจะช่วยลดต้นทุนการทดสอบ เพิ่มประสิทธิภาพในการพัฒนา และปูทางสู่ระบบ UAV ที่มีความอัตโนมัติสูงขึ้น [2]

สมการคณิตศาสตร์พลวัตของอากาศยานไร้คนขับจะเป็นสมการ 6 องศาอิสระ (6DoF) เป็นหนึ่งในแนวคิดสำคัญที่เกี่ยวข้องกับการเคลื่อนที่ของ UAV สามารถเคลื่อนที่ได้อย่างอิสระในอากาศ และสามารถปรับทิศทางได้อย่างแม่นยำ ช่วยเพิ่มประสิทธิภาพในการบินและการควบคุมให้เหมาะสมกับภารกิจต่าง ๆ ตามรูปที่ 1 แสดงการเคลื่อนที่และมุมของควอดโรเตอร์ [3]



รูปที่ 1 6 Degrees of Freedom

การเปลี่ยนพิกัดระหว่าง Earth Frame (inertial frame) และ Body Frame (body-fixed frame) เป็นสิ่งสำคัญในการควบคุม UAV โดยทั่วไปจะถูกควบคุมใน Body Frame แต่ต้องเปลี่ยนพิกัดไปยัง Earth Frame เพื่อทำงานในสภาพแวดล้อมจริง ดังสมการ (1) - (4)

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R(\psi)R(\theta)R(\phi) * \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (1)$$

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3)$$

$$R(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 1 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (4)$$

1.1 เมทริกซ์การหมุน (Rotation Matrices)

เมทริกซ์การหมุนใช้สำหรับการแปลงค่าระหว่างพิกัด Body Frame และ Earth Frame สมการ (5) แสดงเมทริกซ์ความเฉื่อยของควอดโรเตอร์ ซึ่งจะใช้สำหรับการออกแบบตัวควบคุมทัศนคติ เมทริกซ์ความเฉื่อยใช้เพื่อกำหนดแรงบิดที่จำเป็นสำหรับความเร่งเชิงมุมที่ต้องการเกี่ยวกับแกนหมุน

$$J = \begin{bmatrix} Jx & 0 & 0 \\ 0 & Jy & 0 \\ 0 & 0 & Jz \end{bmatrix} \quad (5)$$

โดยใช้กฎของนิวตัน-ออยเลอร์ สามารถกำหนดสูตรแรงและแรงบิดได้ใน สมการที่ (6) และ (7)

ในสมการเหล่านี้ F แทนแรง τ แทนแรงบิด และ m แทนมวล u, v และ w คือความเร็ว ในขณะที่ p, q และ r แทนอัตราเชิงมุม

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \dot{p}J_x \\ \dot{q}J_y \\ \dot{r}J_z \end{bmatrix} + \begin{bmatrix} qr(J_z - J_y) \\ pr(J_x - J_z) \\ pq(J_y - J_x) \end{bmatrix} \quad (7)$$

1.2 เมทริกซ์แปลงอัตราการเปลี่ยนแปลงของมุม (Transformation Matrix)

เพื่อเปลี่ยนค่าการหมุนของ UAV ให้อยู่ในพิกัด Earth Frame ต้องใช้ Transformation Matrix สมการ (8) และ (9) จะแสดงการแปลงเมทริกซ์

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8)$$

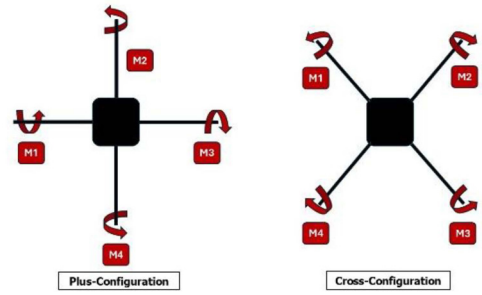
$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\phi) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & \sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (9)$$

1.3 รูปแบบการจัดเรียงของควอดโรเตอร์ (Types of Configuration)

ใน UAV ประเภทควอดโรเตอร์ มีการจัดเรียงใบพัดอยู่ 2 รูปแบบหลัก ได้แก่

1.3.1 Plus-Configuration ใบพัดถูกจัดเรียงเป็นรูปเครื่องหมายบวก (+) มอเตอร์สองตัวทำงานตรงข้ามกันเพื่อควบคุมการหมุนรอบแกน Roll อีกสองตัวใช้สำหรับควบคุมการหมุนรอบแกน Pitch ทุกมอเตอร์ทำงานร่วมกันเพื่อควบคุมการเคลื่อนที่ Altitude และ Yaw

1.3.2 Cross-Configuration ใบพัดถูกจัดเรียงเป็นรูปตัว X ทุกมอเตอร์ต้องทำงานพร้อมกันเพื่อควบคุมทั้ง Roll และ Pitch มีความมั่นคงมากกว่า + Configuration ทำให้เหมาะกับ UAV ที่ใช้กล้องถ่ายภาพเหมาะกับการใช้งานที่ต้องการการทรงตัวสูงดังรูปที่ 2 [4] - [5]



รูปที่ 2 รูปแบบของควอดโรเตอร์

1.4 สมการพลศาสตร์ของควอดโรเตอร์

1.4.1 พลศาสตร์การเคลื่อนที่เชิงเส้น (Translational Dynamics)

พลศาสตร์การเคลื่อนที่เชิงเส้น เป็นพื้นฐานสำคัญในการออกแบบและควบคุมโดรน (UAV) เนื่องจากเกี่ยวข้องกับการเคลื่อนที่ของโดรนในแนวเส้นตรง (แกน x, y, z) และการตอบสนองต่อแรงต่าง ๆ ที่กระทำต่อโดรน เช่น แรงขับจากใบพัด แรงต้านอากาศ และแรงโน้มถ่วง ดังสมการ (10)

$$\begin{aligned} \ddot{x} &= -\frac{1}{m}[(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \\ &\quad (T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{x} \mid \dot{x} \mid] \\ \ddot{y} &= -\frac{1}{m}[(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \\ &\quad (T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{y} \mid \dot{y} \mid] \\ \ddot{z} &= -\frac{1}{m}[(\cos\phi\cos\theta)(T_1 + T_2 + T_3 + T_4) + K_1 \cdot \dot{z} \mid \dot{z} \mid] \end{aligned} \quad (10)$$

1.4.2 พลศาสตร์การหมุน (Rotational Dynamics)

พลศาสตร์การหมุนมีความสำคัญอย่างมากในการออกแบบและควบคุมโดรน เนื่องจากการควบคุมการหมุนรอบแกน Roll, Pitch, และ Yaw ซึ่งจำเป็นสำหรับการเคลื่อนที่และรักษาสมดุลของโดรน ดังสมการ (11)

$$\begin{cases} \dot{p} = \frac{1}{J_x}\left[\frac{\sqrt{2}}{2}l(T_2 + T_3 - T_1 - T_4) - qr(J_z - J_y) - K_2.p|p|\right] \\ \dot{q} = \frac{1}{J_y}\left[\frac{\sqrt{2}}{2}l(T_1 + T_3 - T_2 - T_4) - pr(J_x - J_z) - K_2.q|q|\right] \\ \dot{r} = \frac{1}{J_z}\left[(Q_1 + Q_2 - Q_3 - Q_4) - pq(J_y - J_x) - K_2.r|r|\right] \end{cases} \quad (11)$$

การแปลงความเร็วเชิงมุมไปเป็นมุมออยเลอร์ สมการเหล่านี้ใช้เพื่อแปลงความเร็วเชิงมุมของโดรนไปเป็นค่ามุมออยเลอร์ ดังสมการ (12)

$$\begin{cases} \dot{\phi} = p + q \cdot \tan\theta\sin\phi + r \cdot \tan\theta\cos\phi \\ \dot{\theta} = q \cdot \cos\phi - r \cdot \sin\phi \\ \dot{\psi} = q\frac{\sin\phi}{\cos\phi} + r\frac{\cos\phi}{\cos\theta} \end{cases} \quad (12)$$

แรงต้านจะทำงานตรงกันข้ามกับทิศทางของความเร็วของวัตถุ ซึ่งทำให้มีเครื่องหมายลบอยู่ในสมการ (13) เช่นเดียวกับกรณีของการเคลื่อนที่เชิงเส้น แรงต้านในการหมุนจะต้านทานการหมุนของวัตถุ ซึ่งทำให้มีเครื่องหมายลบในสมการ (14)

$$F_{drag} = -k_1 \cdot v \cdot |v| \quad (13)$$

$$M_{drag} = -k_2 \cdot \omega \cdot |\omega| \quad (14)$$

1.4.3 ความสัมพันธ์ของแรงขับและแรงบิดของใบพัด

$$\begin{aligned} T_{1-4} &= T_{max} \cdot v_{1-4} \\ Q_{1-4} &= Q_{max} \cdot v_{1-4} \end{aligned} \quad (15)$$

โดย $\boldsymbol{v_{1-4}}$ เป็นสัญญาณอินพุตของมอเตอร์

$$t_T = \frac{T_1 + T_2 + T_3 + T_4}{4T_{max}} \quad (16)$$

$$\begin{cases} t_R = \frac{T_2 + T_3 - T_1 - T_4}{4T_{max}} \\ t_P = \frac{T_1 + T_3 - T_2 - T_4}{4T_{max}} \\ t_Y = \frac{Q_1 + Q_2 - Q_3 - Q_4}{4Q_{max}} \end{cases} \quad (17)$$

โดยใช้แรงขับ T_{1-4} และแรงบิดต้าน Q_{1-4} เราสามารถกำหนดสัญญาณควบคุมมาตรฐาน t_T, t_P, t_R และ t_Y ที่ส่งผลต่อแรงขับ การเคลื่อนที่แบบพิทช์ การเคลื่อนที่แบบม้วน และการเคลื่อนที่แบบหันเหของ UAV ตามลำดับ ดังสมการ (15), (16) และ (17)

1.4.4 การจำลองการเคลื่อนที่ของวัตถุแข็งที่มี 6 องศาอิสระ (6 Degrees of Freedom Quaternion) โดยใช้ควอเทอร์เนียนแทนการหมุน

กรรมวิธีนี้ให้สามารถคำนวณการเปลี่ยนแปลงทิศทางของวัตถุได้อย่างราบรื่นและไม่มีปัญหา Gimbal Lock ในการจำลองระบบประกอบไปด้วย 1. อินพุต (Inputs) $F_{xyz}(N)$ เป็นแรงที่กระทำกับวัตถุ และ $M_{xyz}(N \cdot m)$ เป็นโมเมนต์ (แรงบิด) ที่กระทำกับวัตถุในพิกัดตัววัตถุ 2. เอาต์พุต (Outputs) มุมออยเลอร์ Euler Angles $\boldsymbol{\phi, \theta, \psi}$ (rad) (Roll, Pitch, Yaw) ซึ่งสามารถคำนวณจากควอเทอร์เนียน, Direction Cosine Matrix (DCM) ซึ่งเป็นเมทริกซ์ที่แปลงจากพิกัดตัววัตถุ (Body Frame) ไปยังพิกัดเฉื่อย (Inertial Frame) สามารถสร้างได้จากควอเทอร์เนียน, อัตราการหมุนของวัตถุในพิกัดตัววัตถุ $\boldsymbol{\omega_b}$ (rad/s), อัตราการเปลี่ยนแปลงของความเร็วเชิงมุม (Angular Acceleration) $d\omega_b/dt$ (rad/s²) และความเร็วของวัตถุในพิกัดตัววัตถุ $\boldsymbol{v_b}$ (m/s) ควอเทอร์เนียนเป็นวิธีการแทนการหมุนในสามมิติที่ช่วยแก้ปัญหาซิงกูลาริตี้ (Singularity) ของมุมออยเลอร์ (Euler Angles) และสามารถใช้คำนวณการหมุนของวัตถุได้เร็วและมีเสถียรภาพสูงกว่าการใช้เมทริกซ์หมุน [6]

2. การประยุกต์ใช้ MATLAB-Simulink

2.1 การศึกษาลักษณะทางกายภาพของควอดคอปเตอร์

2.1.1 ระบบพิกัด (Coordinate System) และแกนการหมุน (Rotation Axes) ซึ่งเป็นพื้นฐานสำคัญสำหรับการควบคุมและจำลองการบินของโดรน แกน X แทนแกนตามยาวของโดรน เกี่ยวข้องกับการโยนโดรนไปข้างหน้าหรือข้างหลัง แกน Y แทนแกนตามขวางของโดรน เกี่ยวข้องกับการกลิ้งโดรนไปทางซ้ายหรือขวา แกน Z แทนแกนแนวตั้งของโดรน โดยปฏิบัติตามกฎมือขวา การเคลื่อนที่ตามแกน Z เกี่ยวข้องกับการขึ้นหรือลง รวมถึงการหันโดรนเพื่อเปลี่ยนทิศทาง โดยทั่วไปมี 2 ระบบพิกัดหลักที่ใช้ในการวิเคราะห์การเคลื่อนที่ของควอดคอปเตอร์ คือ 1) ระบบพิกัดโลก (Inertial Frame หรือ Earth Frame) เป็นระบบพิกัดอ้างอิงที่ไม่เปลี่ยนแปลง การใช้พิกัดนี้ช่วยให้เราทราบว่าโดรนอยู่ที่ตำแหน่งไหนในอวกาศ และ 2) ระบบพิกัดของตัวควอดคอปเตอร์ (Body Frame หรือ Vehicle-Frame) เป็นระบบพิกัดที่ติดอยู่กับ

ตัวโดรนและหมุนไปตามการเคลื่อนที่ของมัน พิกัดนี้ใช้ในการคำนวณแรงขับของมอเตอร์และโมเมนต์การหมุนของโดรน ซึ่งหมุนรอบ 3 แกนหลัก ซึ่งเรียกว่าการเคลื่อนที่แบบ 6DoF (Degrees of Freedom, องศาอิสระ 6 มิติ)

2.1.2 มวลและความเฉื่อย เป็นปัจจัยสำคัญในระบบควบคุมการบินของโดรน มวลส่งผลต่อแรงขับ พลังงาน และความเร่งของโดรน ในขณะที่ความเฉื่อยส่งผลต่อการตอบสนองต่อแรงบิดและการควบคุมการหมุนรอบแกน มวลจะส่งผลโดยตรงต่อการเคลื่อนที่เชิงเส้น (Translational Motion) และการตอบสนองต่อแรงต่าง ๆ ที่กระทำ เช่น แรงขับจากใบพัด แรงต้านอากาศ และแรงโน้มถ่วง เพื่อที่จะต้องสร้างแรงขับจากใบพัดให้มากพอเพื่อเอาชนะแรงโน้มถ่วง $F_{thrust} \geq \mathrm{mg}$

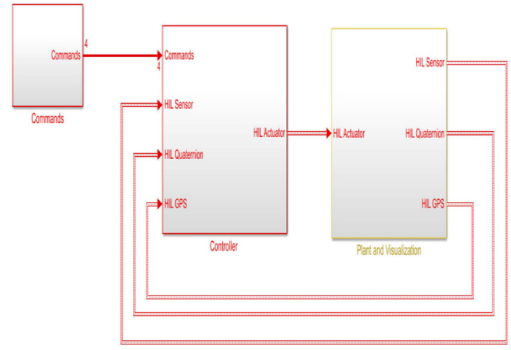
ในระบบควบคุมการบินความเฉื่อย (Inertia) เกี่ยวข้องกับการเคลื่อนที่เชิงมุม (Rotational Motion) ของโดรน ซึ่งส่งผลต่อการควบคุมการหมุนรอบแกน Roll $\emptyset$, Pitch θ และ Yaw φ โมเมนต์ความเฉื่อย (Moment of Inertia), I คือ ความต้านทานของวัตถุต่อการเปลี่ยนแปลงการหมุน $\tau = \mathrm{I}\alpha$ หากโมเมนต์ความเฉื่อยสูง โดรนจะตอบสนองต่อแรงบิดช้าและต้องการแรงบิดมากขึ้นเพื่อเปลี่ยนการหมุน

2.2 การตรวจสอบและปรับแต่งตัวควบคุมการบิน

แสดงการจำลองและทดสอบระบบควบคุมโดยใช้ Simulink กับ PX4 Autopilot โดยใช้ PX4 Host Target เพื่อจำลองพฤติกรรมของยานพาหนะใน PX4 Simulator ทำให้สามารถทดสอบระบบควบคุมได้โดยไม่ต้องใช้ฮาร์ดแวร์จริง

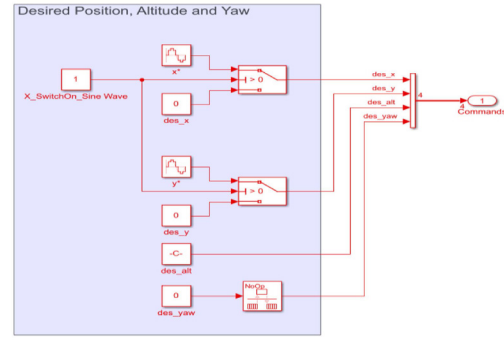
2.2.1 หลักการออกแบบของระบบ HITL (Hardware-in-the-loop) เป็นวิธีการทดสอบที่ช่วยให้สามารถใช้งาน Software เพื่อควบคุม Hardware (PX4 Autopilot) จริงได้ ขณะที่โมเดลฟิสิกส์ของยานพาหนะจะถูกจำลองใน MATLAB-Simulink โมเดลควรจำลองพฤติกรรมของยานพาหนะให้ใกล้เคียงกับความเป็นจริงมากที่สุด เช่น อากาศพลศาสตร์ มอเตอร์ และการควบคุมการเคลื่อนที่ และระบบต้องรองรับการเปลี่ยนพารามิเตอร์ เช่น PID Controller โมเดลสภาพแวดล้อม และตัวแปรระบบควบคุม เป็นต้น [7] - [8]

องค์ประกอบของการใช้งานเพื่อตรวจสอบและปรับแต่งตัวควบคุมการบิน ประกอบด้วย 1) Commands เป็นส่วนที่กำหนดค่าเริ่มต้น 2) Controller เป็นส่วนที่กำหนดค่าการควบคุมควอดโรเตอร์ และ 3) Plant and Visualization เป็นส่วนที่เสมือนและจำลองการบินจริง การตั้งค่า set point จะถูกกำหนดค่าใน commands block ดังรูปที่ 3



รูปที่ 3 Commands, Controller และ Plant and Visualization

X_SwitchOn_Sine Wave เป็นตัวเปิดสวิตช์การทำงานของการสร้างสัญญาณตำแหน่ง x* และ y* ส่งไปยัง comparator ใช้เปรียบเทียบว่า input ที่มาจาก x* และ y* นั้นมากกว่า 0 หรือไม่ ถ้ามากกว่า 0 จะเลือกใช้ x* หรือ y* เป็นตำแหน่งที่ต้องการ ถ้าไม่มากกว่า 0 จะใช้ค่าคงที่จาก des_x และ des_y แทน ซึ่งในที่นี้คือค่า 0 ดังรูปที่ 4

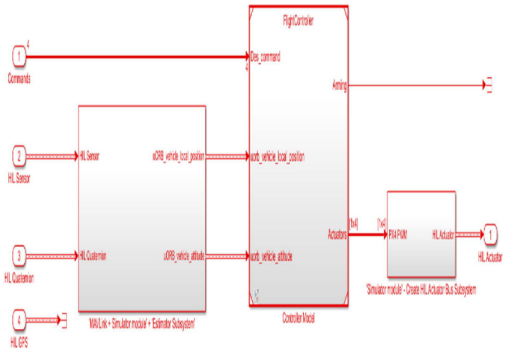


รูปที่ 4 Commands Block

Controller block ประกอบไปด้วย 1) Commands (port 1) เป็นอินพุตขนาดเวกเตอร์ 4 ตัว (เช่น des_x, des_y, des_alt, des_yaw จากภาพก่อนหน้า) เข้าสู่พอร์ต Des_command ในบล็อก FlightController

ส่วนประมวลผลข้อมูลจากเซนเซอร์ ข้อมูลนำเข้าจากเซนเซอร์, 2) HIL Sensor (port 2) ข้อมูลจากเซนเซอร์ที่ใช้ใน HIL เช่น accelerometer และ gyroscope เป็นต้น, 3) HIL- Quaternion (port 3) ค่าควอเทอร์เนียนแทนค่า (orientation) ของอากาศยาน และ 4) HIL GPS (port 4) ข้อมูลตำแหน่งจาก GPS (latitude, longitude, altitude)

'MAVLink + Simulator module' + 'Estimator Subsystem' ใช้สำหรับการแปลงข้อมูลเซนเซอร์ให้กลายเป็นข้อมูลภายในของระบบในส่วนของ Actuator (HIL Actuator Output) พอร์ต PX4 PWM แปลงค่า [1x4] จาก FlightController ให้อยู่ในรูปที่เหมาะสมกับ HIL actuator แล้วส่งผลลัพธ์ออกทาง HIL Actuator เพื่อควบคุมระบบภายนอก เช่น การหมุนของมอเตอร์ในโดรน ดังรูปที่ 5

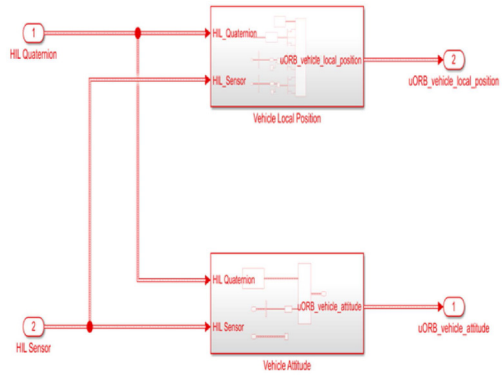


รูปที่ 5 Controller Block

ภายในบล็อก MAVLink + Simulator module + Estimator subsystem block เป็นค่าจำลองเซนเซอร์ที่วัดจากหน่วยจำลองโดรน โดยในบล็อกจะประกอบด้วย

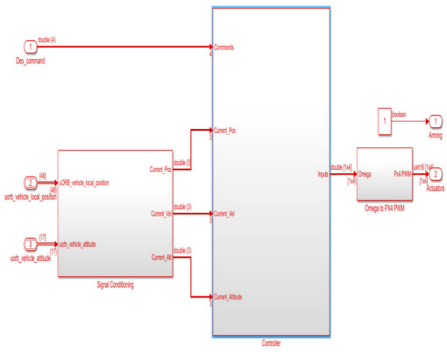
1) Vehicle Local Position block ส่วนนี้มีหน้าที่ประมวลผลข้อมูลจากเซนเซอร์และข้อมูลควอเทอร์เนียน เพื่อประมาณค่าตำแหน่งของยานพาหนะในระบบพิกัดให้ค่าจำลองเซนเซอร์ที่นำมาใช้ ได้แก่ x, y, z, vx, vy, vz, ax, ay และ az

2) Vehicle Attitude Block ส่วนนี้มีหน้าที่ประมวลผลข้อมูลจากเซนเซอร์และข้อมูลควอเทอร์เนียน เพื่อประมาณค่าความเอียงของยานพาหนะ ซึ่งมักจะแสดงในรูปของ Quaternion หรือ Euler Angles ตามรูปที่ 6



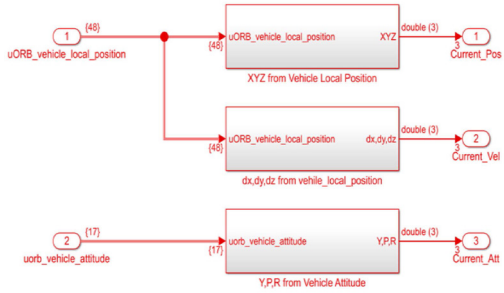
รูปที่ 6 MAVLink + Simulator module + Estimator subsystem

Controller Model ประกอบไปด้วย 1) Desired Command (Des_command) อินพุต ค่าความเร็วที่ต้องการในแนวแกน X, Y, Z และ Yaw Rate, 2) uORB _vehicle_local_position เป็นข้อมูลที่ได้จากเซนเซอร์เกี่ยวกับตำแหน่งของยานพาหนะในพิกัดท้องถิ่น ข้อมูลนี้อาจประกอบด้วยพิกัด North, East, Down (NED) และอาจมีข้อมูลความเร็วและอื่น ๆ ที่เกี่ยวข้อง 3) uORB_vehicle_attitude เป็นข้อมูลมุมของยานพาหนะ (ท่าทางของยาน) มีการแยกค่า Yaw, Pitch, Roll ออกมาเพื่อใช้เป็น Current_Att (มุมปัจจุบันของยาน) เป็นข้อมูลเกี่ยวกับมุมของยานพาหนะ (ท่าทางของยาน) มีการแยกค่า Yaw, Pitch, Roll ออกมาเพื่อใช้เป็น Current_Att (มุมปัจจุบันของยาน) 4) Signal Conditioning ทำหน้าที่ปรับสัญญาณให้เหมาะสมสำหรับการนำไปใช้ มีการแยกข้อมูลออกเป็น 3 ส่วน Current_Pos เป็นข้อมูลตำแหน่งปัจจุบันของยานพาหนะ, Current_Vel เป็นข้อมูลความเร็วปัจจุบันของยานพาหนะ และ Current_Attitude เป็นข้อมูลความเอียงปัจจุบันของยานพาหนะ และ 5) Controller ทำหน้าที่เปรียบเทียบสถานะปัจจุบันของยานพาหนะกับคำสั่งที่ต้องการ จากนั้นจะคำนวณเอาต์พุตที่เหมาะสมเพื่อสั่งการไปยัง Actuators เพื่อให้ยานพาหนะเคลื่อนที่ไปตามเป้าหมายดังรูปที่ 7



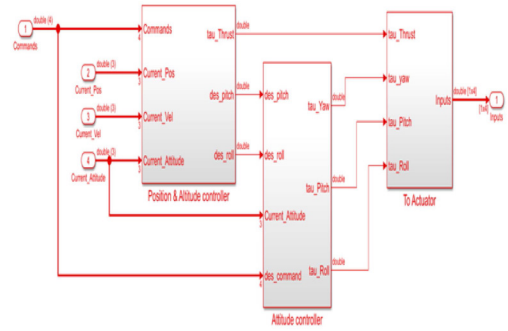
รูปที่ 7 Controller Model

Signal Conditioning Block ทำหน้าที่แปลงข้อมูลจากเซนเซอร์หรือ estimator ให้อยู่ในรูปที่สะดวกต่อการใช้งานในระบบควบคุม เช่น ตำแหน่ง ความเร็ว และมุมท่าทาง อินพุตประกอบไปด้วย uORB_vehicle_local_position เป็นข้อมูลสถานะของตำแหน่งจากระบบ PX4 ที่ได้จากการประเมินของ Estimator ประกอบด้วยข้อมูลตำแหน่งในแกน x, y, z และความเร็ว dx, dy, dz และ uORB_vehicle_attitude คือ ข้อมูลท่าทางของยานพาหนะในรูปควอเทอร์เนียน หรือ Euler angles (Yaw, Pitch, Roll) ดังรูปที่ 8



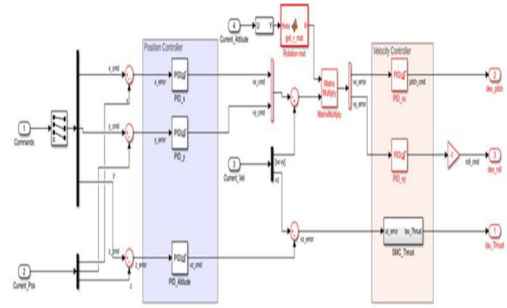
รูปที่ 8 Signal Conditioning Block

Controller Block ใน Flight Control ประกอบไปด้วย อินพุต Commands ตำแหน่งที่ต้องการ (เช่น x, y, z) Current_Pos, Current_Vel และ Current_Attitude Position & Attitude Controller Block บล็อกนี้ทำหน้าที่ควบคุมตำแหน่งของยานพาหนะโดยการสร้างคำสั่งสำหรับ attitude ที่ต้องการ (desired attitude) และแรงขับที่ต้องการ (desired thrust) To Actuator Block ทำหน้าที่แปลง Torque และ Thrust ที่ต้องการเหล่านี้ไปเป็นสัญญาณควบคุมสำหรับ Actuators ของยานพาหนะดังรูปที่ 9



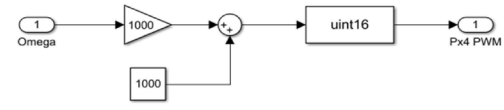
รูปที่ 9 Controller Block ใน Flight Control

Position & Altitude Controller Commands จะมีอินพุต (4 ค่า เช่น x, y, z, yaw) Current_Pos ตำแหน่งปัจจุบัน (x, y, z), Current_Vel ความเร็วปัจจุบัน และ Current_Attitude มุมเอียงปัจจุบัน (Roll, Pitch, Yaw) จะทำการเปรียบเทียบค่าที่ต้องการ (Commands) กับค่าปัจจุบัน tau_Thrust, des_pitch และ des_roll จะส่งค่าไปยัง Attitude Controller เมื่อได้รับค่ามาแล้วจะคำนวณเปรียบเทียบค่า Attitude กับค่าปัจจุบันเพื่อคำนวณ torque ที่ต้องใช้ในแต่ละแกนและส่งค่าที่ได้ไปยัง To Actuator เพื่อแปลง torque แต่ละแกนเป็นค่า angular velocity (Omega) หรือค่า PWM สำหรับแต่ละมอเตอร์ดังรูปที่ 10



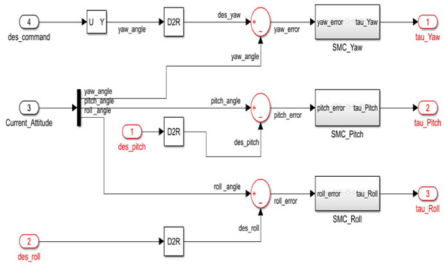
รูปที่ 10 Position & Altitude Controller

Omega to PX4 PWM block แปลงค่าสั่งมอเตอร์เป็น PWM 1000-2000 us ดังรูปที่ 11



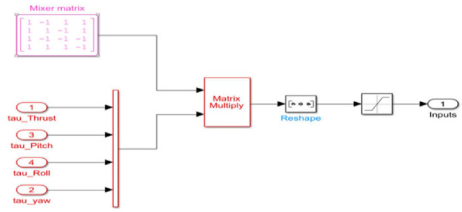
รูปที่ 11 Omega to PX4 PWM Block

Attitude Controller จะแบ่งการทำงานเป็น 3 ส่วน คือ 1) การควบคุมแกน Yaw โดย des_command คำสั่งมุม yaw ถูกแยกค่ามาเฉพาะ yaw_angle ผ่านกล่อง D2R (Degree to Radian) แปลงเป็นเรเดียน des_yaw เทียบกับค่าปัจจุบันจาก Current_Attitude เป็น yaw_angle แล้วคำนวณค่า yaw_error จาก des_yaw-yaw_angle ส่งเข้าไปยัง SMC_Yaw เพื่อคำนวณแรงบิด 2) การควบคุมแกน Pitch โดย Current_Attitude ให้ค่า pitch_angle (มุมปัจจุบันของ pitch) des_pitch มาจาก Position Controller ซึ่งเป็นค่าที่ต้องการแปลง des_pitch เป็นเรเดียน เทียบกับ pitch_angle แล้วคำนวณ pitch_error แล้วส่งไปยัง SMC_Pitch และ 3) การควบคุมแกน Roll des_roll แปลงเป็นเรเดียนเทียบกับ roll_angle จาก Current_Attitude จากนั้นคำนวณ error แล้วส่งไปยัง SMC_Roll ดังรูปที่ 12



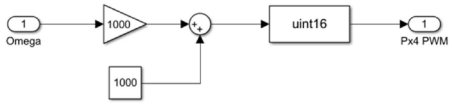
รูปที่ 12 Attitude Controller

To Actuator Block จะแปลงค่า control actuators ไปเป็นค่าเพื่อสั่งมอเตอร์ชุดขับเคลื่อนดังรูปที่ 13



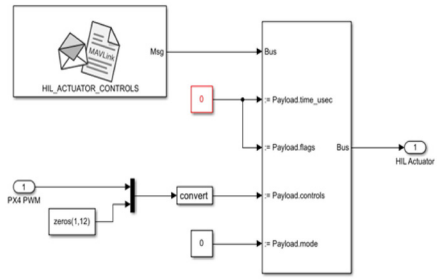
รูปที่ 13 To Actuator Block

Omega to PX4 PWM Block แปลงค่าสั่งมอเตอร์เป็น PWM 1000-2000 us ดังรูปที่ 14



รูปที่ 14 Omega to PX4 PWM Block

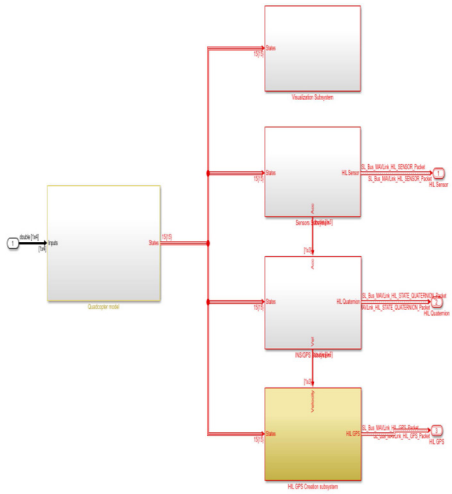
Simulator module-Create HIL Actuator Bus Subsystem เป็นการแปลงค่า PWM ไปเป็นค่าจำลองส่วนขับเคลื่อน ดังรูปที่ 15



รูปที่ 15 Simulator module - Create HIL Actuator Bus Subsystem

ในส่วนของ Plant and Visualization จะประกอบไปด้วย 1) Quadcopter model รับอินพุต: PWM (จาก PX4 เป็นคำสั่งควบคุมมอเตอร์) เอาต์พุต states ถูกส่งต่อไปยัง Subsystems เพื่อแปลงเป็น sensor data 2) Visualization Subsystem แปลงเป็นภาพเคลื่อนไหวของโดรน 3) Sensors Subsystem รับ States แปลงข้อมูลเป็นความเร่ง (accelerometer), ความเร็วเชิงมุม (gyroscope), สนามแม่เหล็ก (magnetometer) และความดันบรรยากาศ (barometer) ส่งไปยัง PX4 ผ่าน MAVLink, 4) IMU/Quaternion Subsystem ส่งออก HIL Quaternion ใช้ quaternion จาก States เพื่อสร้างข้อมูล orientation สร้าง MAVLink HIL_ATTITUDE_QUATERNION message เอาต์พุตเป็น HIL Quaternion แล้วส่งให้ PX4 ใช้แทน IMU attitude sensing และ 5) HIL GPS Creation Subsystem ส่งออก HIL GPS ใช้ตำแหน่ง (x, y, z) จาก States แปลงเป็นพิกัด GPS เอาต์พุต HIL GPS เป็นข้อมูล GPS จำลองที่ส่งเข้า PX4 ผ่าน MAVLink

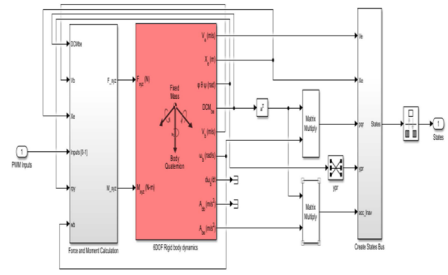
กระบวนการทำงานโดยรวม คือ PX4 ส่ง PWM ไปยัง Quadcopter model แล้วจะทำการคำนวณ States จากนั้น States จะส่งค่าไปยัง Subsystem ต่าง ๆ แต่ละ Subsystem สร้างข้อมูล Sensor จำลอง และส่งกลับไปยัง PX4 โดย PX4 ทำงานเหมือนอยู่ในสถานการณ์จริง (Real Hardware) ทั้งที่อยู่ในการจำลอง (HIL) ดังรูปที่ 16



รูปที่ 16 Plant and Visualization

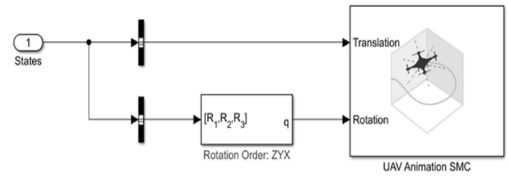
Quadcopter model ประกอบไปด้วย Force and Moment Calculation ซึ่งคำนวณให้ค่าแรงและโมเมนต์ เพื่อส่งไปยังแบบจำลองไดนามิกของโดรน (6DoF Rigid body dynamics) แบบจำลองจะคำนวณหาค่าควอเทอร์เนียนเพื่อเข้าสู่ Altitude > 0 Constraint and Create State Bus Blocks เพื่อสร้างเวกเตอร์ข้อมูลแบบจำลองขึ้นดังรูปที่ 17 การอัปเดตควอเทอร์เนียนสามารถคำนวณได้จากอัตราการหมุน $\otimes$ หมายถึงการคูณควอเทอร์เนียน $\omega_q = [0, \omega_x, \omega_y, \omega_z]$ คือควอเทอร์เนียนที่ได้จากอัตราการหมุน (18)

$$\dot{q} = \frac{1}{2} q \otimes \omega_q \qquad (18)$$



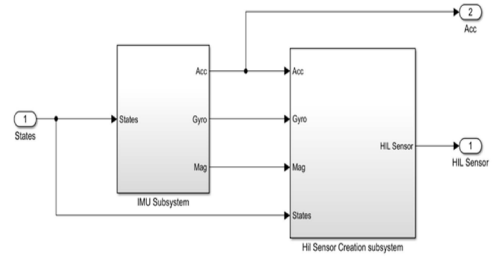
รูปที่ 17 Quadcopter model

Visualization Subsystem นำค่า state จาก Quadcopter model มาเพื่อสร้างแบบจำลองเส้นทางการบินในรูปแบบ 3 มิติ ดังรูปที่ 18



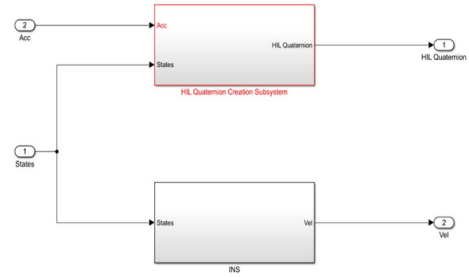
รูปที่ 18 Visualization Subsystem

Sensors Subsystem ทำหน้าที่จำลองค่าเซนเซอร์ เช่น Accelerometer, Gyroscope และ Magnetometer เพื่อประมวลผลของระบบนำทางและควบคุมการบิน ดังรูปที่ 19



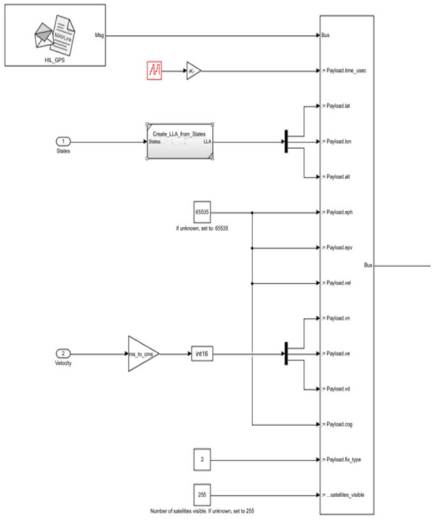
รูปที่ 19 Sensors Subsystem

INS/GPS Subsystem ทำหน้าที่จำลองค่าที่ได้จาก Instrument (INS: Inertial Navigation System) และค่าความเร็วจาก GPS เพื่อคำนวณค่าตำแหน่งและทิศทาง โดยใช้ข้อมูลจาก IMU (Inertial Measurement Unit) ดังรูปที่ 20



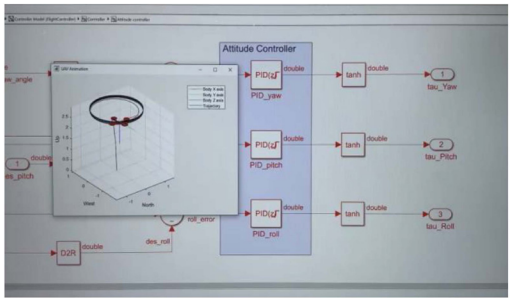
รูปที่ 20 INS/GPS Subsystem

HIL GPS Creation Subsystem ทำหน้าที่จำลองค่าตำแหน่ง GPS และความสูงของยานพาหนะ ข้อมูล GPS และความสูงถูกสร้างขึ้นจาก Quadcopter Model เพื่อให้ตรงกับพฤติกรรมของยานพาหนะจริง ข้อมูลนี้ใช้ในการทดสอบระบบนำทางและการควบคุมของ PX4 Autopilot ดังรูปที่ 21



รูปที่ 21 HIL GPS Creation Subsystem

Subsystem ใน HITL Simulink Plant Model มีหน้าที่ในการสร้างข้อมูลที่จำเป็นสำหรับการจำลองและทดสอบระบบการบิน โดยใช้ข้อมูลจาก Quadcopter Model เพื่อให้ได้ค่าที่ใกล้เคียงกับสถานการณ์จริงมากที่สุด ดังรูปที่ 22



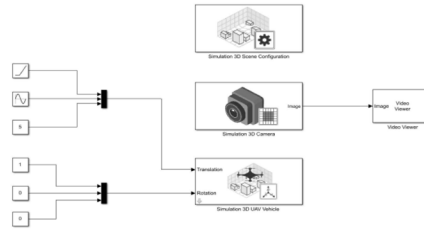
รูปที่ 22 ผลลัพธ์การจำลอง Simulink

3. การจำลอง 3 มิติโดยใช้ MATLAB-Simulink

MATLAB-Simulink เป็นโปรแกรมที่มีประสิทธิภาพสูงในการจำลองระบบไดนามิกและการควบคุม ซึ่งสามารถนำมาใช้สร้างแบบจำลองโดรนในรูปแบบ 3 มิติ ได้อย่างแม่นยำ โดย Simulink ช่วยให้สามารถออกแบบทดสอบ และปรับแต่งตัวควบคุม (Controller) สำหรับโดรนได้อย่างสะดวก ไม่ว่าจะเป็นระบบควบคุมเสถียรภาพ (Stabilization), ระบบนำทาง (Navigation) หรือระบบหลีกเลี่ยงสิ่งกีดขวาง (Obstacle Avoidance)

3.1 กำหนดอุปกรณ์ที่ใช้งานในการจำลอง

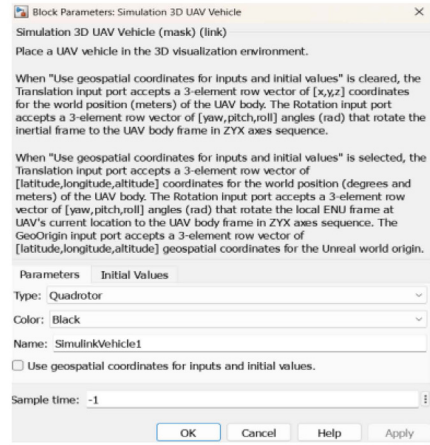
อุปกรณ์หลักสำหรับการจำลองการบิน ประกอบด้วย Simulation 3D Scene Configuration, Simulation 3D Camera, Simulation 3D UAV Vehicle และ Video Viewer โดยอุปกรณ์แต่ละอุปกรณ์จะทำหน้าที่ กำหนดลักษณะของฉาก, กำหนดการเคลื่อนที่ของ UAV ในพิกัด X, Y, Z รองรับการหมุนรอบแกน (Roll, Pitch, Yaw) และโดยการเชื่อมต่อของบล็อกจะเป็นไปตามรูปที่ 23



รูปที่ 23 Quadcopter 3D simulation

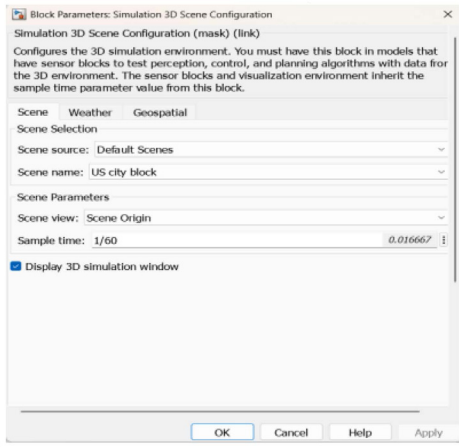
Simulation 3D UAV Vehicle ใช้สำหรับจำลองและแสดงผล อากาศยานไร้คนขับ (UAV: Unmanned Aerial Vehicle) ในสภาพแวดล้อม 3 มิติของ Simulink 3D Simulation Environment โดยรองรับการตั้งค่าต่าง ๆ เช่น ประเภท UAV, สี, พิกัด เป็นต้น

ค่า Translation input เป็นเวกเตอร์ 3 มิติ ซึ่งกำหนดตำแหน่ง UAV ในระบบพิกัดคาร์ทีเซียน (หน่วยเป็นเมตร) ค่า Rotation input เป็นเวกเตอร์ 3 มิติ [yaw, pitch, roll] (หน่วยเป็นเรเดียน) [8] ซึ่งใช้กำหนดการหมุนของ UAV ดังรูปที่ 24



รูปที่ 24 Simulation 3D UAV Vehicle

บล็อกสำหรับตั้งค่าสภาพแวดล้อมการจำลอง 3D เพื่อใช้ทดสอบระบบที่มีเซนเซอร์ เช่น การรับรู้ (perception) การควบคุม (control) และอัลกอริทึมวางแผน (planning algorithms) เป็นต้น โดยอาศัยข้อมูลจากสภาพแวดล้อม 3D Scene source เลือกใช้ฉากจาก Default Scenes และ Scene name เลือกฉากเป็น US city block ตามรูปที่ 25



รูปที่ 25 Simulation 3D Scene Configuration

3.2 การจำลองแบบติดตามวงโคจรและการนำทางตามจุดอ้างอิง (Orbit follower and Waypoint follower navigation)

หนึ่งในแนวทางหลักในการควบคุมเส้นทางของ UAV คือ การติดตามวงโคจร (Orbit Following) เป็นเทคนิคเคลื่อนที่รอบจุดศูนย์กลางในลักษณะของวงกลมหรือวงรี โดยคงระยะห่างจากจุดศูนย์กลางให้คงที่ เทคนิคนี้มักใช้ในงานเฝ้าระวัง ตรวจสอบพื้นที่ และการบินลาดตระเวน และการนำทางตามจุดอ้างอิง (Waypoint Following) เป็นแนวทางที่ UAV จะเคลื่อนที่ไปยังชุดของจุดที่กำหนดไว้ล่วงหน้า จะปรับทิศทางและความเร็วเพื่อไปถึงแต่ละจุดตามลำดับ เทคนิคนี้เหมาะสำหรับภารกิจที่ต้องมีการกำหนดเส้นทางที่แน่นอน เช่น การขนส่งอัตโนมัติ การตรวจสอบโครงสร้างพื้นฐาน และการลาดตระเวนพื้นที่ สามารถดำเนินการผ่านแพลตฟอร์ม เช่น MATLAB-Simulink ซึ่งมีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจำลองได้อย่างแม่นยำ

3.2.1 หลักการทำงาน Orbit Following

ต้องรักษาระยะห่างจากศูนย์กลางของวงโคจรให้คงที่ ใช้ Lookahead Point เพื่อช่วยกำหนดเส้นทางและปรับมุมหันการปรับค่าทิศทาง (Desired Course) และมุมส่าย (Desired Yaw) เพื่อช่วยให้อยู่ในวงโคจร

อินพุตหลัก ประกอบด้วย ตำแหน่งของ UAV (Pose) [x, y, z และ course], ตำแหน่งศูนย์กลางวงโคจร (Center) [x, y, z], รัศมีของวงโคจร (Radius) และทิศทางการหมุน (Turn Direction)

เอาต์พุตหลัก ประกอบด้วย จุดอ้างอิงเป้าหมาย (Lookahead Point) ทิศทางที่ต้องการ (Desired Course) มุมที่ต้องปรับ (Desired Yaw) และค่าความผิดพลาดระหว่างตำแหน่งปัจจุบันกับเส้นทาง (Cross-Track Error)

3.2.2 หลักการทำงาน Waypoint Follower

การนำทางตามจุดอ้างอิงเป็นกระบวนการที่บินผ่านชุดของพิกัดที่กำหนดไว้ ซึ่งใช้ในภารกิจที่ต้องมีการกำหนดเส้นทางล่วงหน้า เช่น การส่งของหรือการลาดตระเวน เป็นต้น

อินพุตหลัก ประกอบด้วย ตำแหน่งของ UAV (Pose) [x, y, z และ course], Waypoints เมทริกซ์ [x, y, z], ระยะมองล่วงหน้าเพื่อช่วยกำหนดเป้าหมาย (Lookahead Distance), Close Loop ระบุว่า UAV จะวนกลับไปยังจุดเริ่มต้นหรือไม่

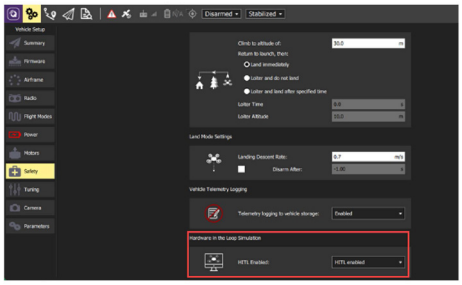
เอาต์พุตหลัก ประกอบด้วย จุดอ้างอิงเป้าหมาย (Lookahead Point), ทิศทางที่ต้องการ (Desired Course), มุมที่ต้องปรับ (Desired Yaw) และ Waypoint Index แสดงจุดอ้างอิงที่ UAV กำลังมุ่งหน้าไป

4. QGroundControl

QGroundControl (QGC) เป็นซอฟต์แวร์ที่ใช้ควบคุมและตรวจสอบอากาศยานไร้คนขับ (UAV) ที่ใช้ PX4 หรือ ArduPilot เป็น Flight Controller ใช้ MATLAB ร่วมกับ QGroundControl เพื่อควบคุมหรือวิเคราะห์ข้อมูลจาก UAV รองรับ MAVLink ซึ่งเป็นโปรโตคอลสำหรับสื่อสารกับ UAV MATLAB ก็สามารถรับ-ส่งข้อมูลผ่าน MAVLink ได้โดยใช้ UAV Toolbox หรือ Robotics System Toolbox [9]

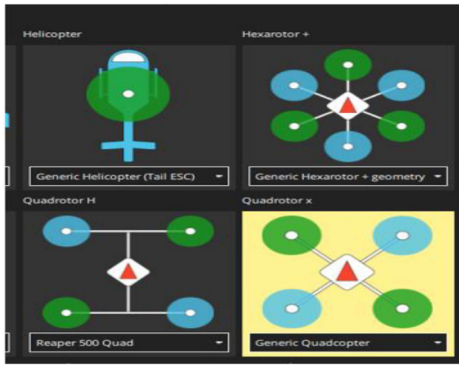
การตั้งค่า PX4 Autopilot ในโหมด Hardware -in-the-Loop (HITL) ผ่าน QGroundControl (QGC)

เริ่มต้นโดยการเชื่อมต่ออุปกรณ์ Autopilot เข้ากับ QGC ผ่านสาย USB ไปที่เมนู 'Setup' แล้วเลือก 'Safety' ในตัวเลือก 'HITL Enabled' ให้เลือก 'Enabled' เพื่อเปิดการใช้งานโหมด (HITL) ดังรูปที่ 26



รูปที่ 26 การเปิดใช้งานโหมด Hardware-in-the-Loop

ไปที่เมนู 'Setup' แล้วเลือก 'Airframes' เลือก 'Generic Quadcoptor' คลิก 'Apply and Restart' ที่มุมขวาบนของหน้า Airframe Setup ดังรูปที่ 27

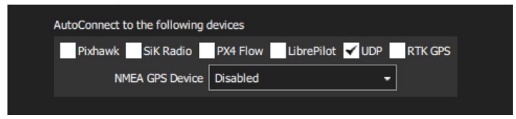


รูปที่ 27 ตั้งค่า Airframes

กำหนดค่ามอเตอร์ผ่าน PWM MAIN ตัวเลือกในการกำหนดค่ามอเตอร์สี่ตัวจะปรากฏขึ้น สามารถกำหนดค่ามอเตอร์สี่ตัวโดยใช้ช่องสัญญาณหลัก 1-4 PWM จำนวนช่องสัญญาณหลักที่ใช้ควรตรงกับจำนวนมอเตอร์ที่ต้องการใช้เฉพาะช่องสัญญาณหลัก PWM สำหรับมอเตอร์ที่เชื่อมต่อผ่านอินเทอร์เฟซ PWM ดังรูปที่ 28

รูปที่ 28 การกำหนดค่า Actuator

ตั้งค่า AutoConnect ไปที่แท็บ General ในเมนูการตั้งค่า เลือกใช้ UDP การตั้งค่านี้จะหยุดการสื่อสารระหว่าง QGC และ PX4 Autopilot ผ่าน USB และทำให้สื่อสารผ่าน UDP แทน ดังรูปที่ 29

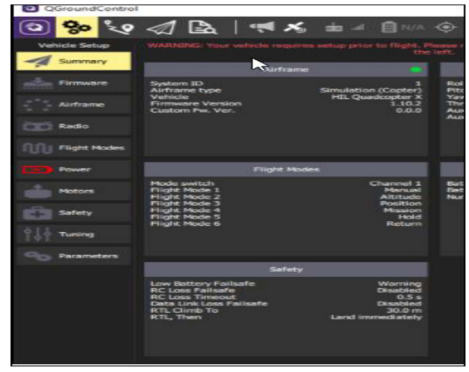


รูปที่ 29 ตั้งค่า AutoConnect

เปิดใช้งาน Virtual Joystick ในแท็บ General ของเมนูการตั้งค่า ไปที่ส่วน Fly View เลือกตัวเลือก Virtual Joystick เพื่อใช้จอยสติกเสมือนแทนรีโมตคอนโทรล ตั้งค่าจอยสติกและ Failsafe

ตั้งค่าพารามิเตอร์ COM_RC_IN_MODE เป็น Joystick only เพื่ออนุญาตให้ใช้จอยสติกและปิดการตรวจสอบ RC ตั้งค่าพารามิเตอร์ COM_DISARM_LAND เป็น -1 เพื่อปิดการทำงานอัตโนมัติของการปลดอาวุธเมื่อ QGC ตรวจจับการลงจอด ตั้งค่าพารามิเตอร์ NAV_ACC_RAD เป็น 10

ตรวจสอบการตั้งค่าใน QGC ตรวจสอบว่าไม่มีการแจ้งเตือนในแท็บ Vehicle Settings ของ QGC ยกเว้นแท็บ Power ที่อาจมีการแจ้งเตือนได้หากมีการแจ้งเตือนในแท็บ Flight Modes ให้ตั้งค่าโหมดการบินสำหรับแต่ละช่องสัญญาณ ตรวจสอบว่ามีการใส่ SD card ในอุปกรณ์เพื่อให้สามารถอัปโหลดภารกิจจาก QGC ได้ ตามรูปที่ 30



รูปที่ 30 ตรวจสอบการตั้งค่าใน QGC

ตรวจสอบการตั้งค่า MAVLink วางแผนที่จะใช้ฟังก์ชัน Monitor & Tune ผ่านพอร์ต TELEM1, TELEM2 หรือ TELEM3 ของฮาร์ดแวร์ Pixhawk ให้ตรวจสอบว่าพอร์ตเหล่านี้

ไม่ได้เปิดใช้งาน MAVLink ตรวจสอบได้โดยดูค่าของพารามิเตอร์ MAV_0_CONFIG และ MAV_1_CONFIG ซึ่งควรถูกตั้งค่าเป็น Disabled เมื่อใช้งาน Monitor & Tune หลังจากตั้งค่าทั้งหมดเสร็จสิ้น ให้ปิดโปรแกรม QGC เพื่อให้การตั้งค่ามีผล

5. บทสรุป

ระบบควบคุมการบินอัตโนมัติของอากาศยานไร้คนขับ (UAV) โดยใช้แนวคิด Model-Based Design ผ่าน Simulink ร่วมกับเฟิร์มแวร์ PX4 และการทดสอบด้วย Hardware-in-the-Loop (HITL) ที่พัฒนาขึ้นและอธิบายได้อย่างละเอียดในบทความวิจัยฉบับนี้ เพื่อก่อให้เกิดความมั่นใจถึงความเสถียรและประสิทธิภาพของระบบควบคุมก่อนนำไปใช้งานจริง

กระบวนการพัฒนาเริ่มจากการออกแบบอัลกอริทึมควบคุมใน Simulink และทำการจำลองการบินโดยใช้ HITL ซึ่งช่วยให้สามารถทดสอบการตอบสนองของซอฟต์แวร์และฮาร์ดแวร์ในสภาพแวดล้อมที่ใกล้เคียงกับสถานการณ์จริง อัลกอริทึมที่ผ่านการตรวจสอบจะถูกติดตั้งบนชุดควบคุมการบินที่ใช้ PX4 Autopilot โดยใช้ QGroundControl (QGC) เป็นเครื่องมือในการกำหนดค่าปรับแต่งพารามิเตอร์แบบเรียลไทม์ และวางแผนภารกิจ

ระบบที่พัฒนาขึ้นเพื่อรองรับการนำทางอัตโนมัติผ่านจุดนำทาง (Waypoint-Based Navigation) จึงทำให้ UAV สามารถปฏิบัติภารกิจได้อย่างแม่นยำ และสามารถควบคุมเส้นทางบินผ่านแผนที่ดิจิทัลได้อย่างมีประสิทธิภาพ ช่วยให้สามารถนำไปประยุกต์ใช้กับงานด้านการสำรวจทางอากาศ การตรวจการณ์ และภารกิจทางยุทธวิธีได้อย่างมีประสิทธิภาพ

6. กิตติกรรมประกาศ

ขอขอบคุณ สถาบันเทคโนโลยีป้องกันประเทศ ที่ได้ให้การสนับสนุนทุนตามสัญญาที่ 670201 รวมถึงคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ที่ได้อำนวยความสะดวกทั้งในเรื่องสถานที่และสิ่งสนับสนุนต่าง ๆ และสาขาเทคโนโลยีเพื่อการพัฒนาที่ยั่งยืนที่ได้ช่วยสนับสนุนเครื่องมือ อุปกรณ์ และสิ่งอำนวยความสะดวกในการทำวิจัยในครั้งนี้

7. เอกสารอ้างอิง

[1] B. O. Fereshte, “Programming of an Indoor Autonomous Drone and Metrological Characterization,” M.Sc. thesis, Politecnico di Milano, Milan, Italy, 2020. [Online]. Available: https://hdl.handle.net/10589/166702

[2] A. A. Elgohary, A. M. Ashry, A. M. Kaoud, M. M. Gomaa, M. H. Darwish, and H. E. Taha, “Hardware-in-the-Loop Simulation of UAV Altitude Hold Autopilot,” in *2022 AIAA SciTech Forum*, p. 1520, doi: 10.2514/6.2022-1520.

[3] J. Iqbal, R. ul Islam, and H. Khan, “Modeling and Analysis of a 6 DOF Robotic Arm Manipulator,” *Canadian J. Elect. Electron. Eng.*, vol. 3, no. 6, pp. 300 - 306, Jul. 2012. [Online]. Available: https://www.researchgate.net/publication/280643085_Modeling_and_analysis_of_a_6_DOF_robotic_arm_manipulator

[4] C. Cömert and C. Kasnakoğlu, “Comparing and Developing PID and Sliding Mode Controllers for Quadrotor,” *Int. J. Mech. Eng. Robot. Res.*, vol. 6, no. 3, pp. 194–199, 2017, doi: 10.18178/ijmerr.6.3.194-199.

[5] J. Ackermann, *Robust Control: Systems with Uncertain Physical Parameters,* London, UK: Springer-Verlag, 1993.

[6] R. Isermann, *Mechatronic Systems: Fundamentals.* Berlin, Germany: Springer, 2005.

[7] MathWorks. “HITL Simulink Plant Example.” MATHWORKS.com. https://www.mathworks.com/help/uav/px4/ref/hitl-simulink-plant-example.html (accessed Mar. 19, 2025).

[8] MathWorks. “Simulator Plant Model Example.” MATHWORKS.com. https://www.mathworks.com/help/releases/R2021b/supportpkg/px4/ref/simulator-plant-model-example.html (accessed Mar. 19, 2025).

[9] MathWorks. “Configure Actuators Using QGround Control.” MATHWORKS.cn. https://ww2.mathworks.cn/help/uav/px4/ug/configure-actuator-qgc.html (accessed: Mar. 20, 2025).