

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Інформаційної безпеки та електронних комунікацій

(повне найменування інституту, факультету)

Інформаційних технологій та телекомунікацій

(повне найменування кафедри )

## Пояснювальна записка

до дипломного проекту (роботи)

Магістр

(ступінь вищої освіти)

на тему ОПТИМІЗАЦІЯ АЛГОРИТМУ СТАБІЛІЗАЦІЇ БПЛА ПО КРЕНУ

Виконав: студент(ка) \_6\_ курсу, групи РТ-512м

Спеціальності 172 Телекомунікації та радіотехніка

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інтелектуальні технології мікросистемної  
радіоелектронної техніки

Керівник Огренич Є.В.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Факультет інформаційної безпеки та електронних комунікацій

Кафедра Інформаційних технологій електронних засобів

Ступінь вищої освіти Магістр

Спеціальність 172 Телекомунікації та радіотехніка  
 (код і найменування)

Освітня програма (спеціалізація): Інтелектуальні технології мікросистемної радіоелектронної техніки  
 (назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТЕЗ \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

Шейко Олексій Анатолійович

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Оптимізація алгоритму стабілізації БПЛА по крену

керівник проєкту (роботи) Огренич Євген Вікторович, к.т.н, доцент каф.ІТЕЗ

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом закладу вищої освіти від «21» листопада 2023 року № 448

2. Строк подання студентом проєкту (роботи) 21 грудня

3. Вихідні дані до проєкту (роботи) структура системи керування двигунами БПЛА

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Конструкція БПЛА літакового типу, структура стабілізації крену, компоненти системи стабілізації крену, алгоритм фільтрації сигналів, запропонований алгоритм стабілізації крену, рТехнічна реалізація системи стабілізації, висновки, перелік посилань, додаток А, додаток Б

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
23 слайди

## 6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
<b>1</b>	<b>Євген ОГРЕНИЧ</b>	<b>03.10.23</b>	
<b>2</b>	<b>Євген ОГРЕНИЧ</b>	<b>17.10.23</b>	
<b>3</b>	<b>Євген ОГРЕНИЧ</b>	<b>07.11.23</b>	
<b>4</b>	<b>Євген ОГРЕНИЧ</b>	<b>21.11.23</b>	
<b>нормоконтроль</b>	<b>Ірина ПОСПЕСВА</b>	<b>01.12.23</b>	

7. Дата видачі завдання «04» \_\_вересня\_\_\_\_2023\_року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
<b>1</b>	<b>Структура системи стабілізації крену БПЛА</b>	<b>9.10.23</b>	
<b>2</b>	<b>Аналіз алгоритмів фільтрації сигналів</b>	<b>11.10.23</b>	
<b>3</b>	<b>Розробку алгоритму керування кутом крену</b>	<b>12.10.23</b>	
<b>4</b>	<b>Валідація запропонованого алгоритму у MATLAB</b>	<b>20.10.23</b>	
<b>5</b>	<b>тримання кута крену за допомогою MEMS датчиків</b>	<b>15.11.23</b>	
<b>6</b>	<b>изначення передавальної функції між кутом відхилення елеронів та кутом крену</b>	<b>25.11.23</b>	
<b>7</b>	<b>Випробування системи</b>	<b>02.12.23</b>	
<b>8</b>	<b>Оформлення ПЗ та захист дипломного проєкту</b>	<b>16.12.23</b>	

Студент(ка)

( підпис )

Шейко О.А.  
(прізвище та ініціали)

Керівник проєкту (роботи)

( підпис )

Огренич Є.В.  
(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до магістерської роботи: 70 сторінок, 39 рисунків, 9 таблиць, 18 джерел, 2 додатки.

БЕЗПЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, ПОЛЬОТНИЙ КОНТРОЛЕР, ПРОГРАМА, СТАБІЛІЗАЦІЯ, КУТ КРЕНУ, ГІРОСКОП, ПІД-РЕГУЛЯТОР, STM32, КЕРУВАННЯ, УТРИМАННЯ

Метою даної є оптимізація алгоритму стабілізації крену БПЛА та розробка системи стабілізації безпілотного літального апарату літакового типу по крену і проведення випробувань отриманої системи.

Об'єктом дослідження є розробка методів оптимізації алгоритму по крену на базі мікроконтролера STM32.

Результатом дослідження є робочий прототип системи.

Значимість роботи полягає у створенні системи, яка забезпечуватиме горизонтальне положення літального апарату протягом усього польоту, тим самим значно полегшуючи роботу оператора БПЛА на базі доступного мікроконтролера сімейства STM32F3.

У майбутньому планується подальший розвиток системи: реалізація системи стабілізації за іншими контурами регулювання, а саме по ризику та тангажу, а також оптимізація наявного коду.

## ЗМІСТ

Реферат.....	4
Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ .....	8
1 Проектування системи стабілізації безпілотного літального апарату літакового типу за креном .....	10
1.1 Управління кутом крену .....	10
1.1.1 Конструкція БПЛА літакового типу.....	11
1.2 Структура системи стабілізації крену БПЛА .....	16
1.3 Компоненти системи стабілізації кута крену .....	19
1.3.1 Мікроконтролер STM32F303VCT6 .....	19
1.3.2 MEMS датчики L3GD30 та LSM303DLHC .....	21
1.4 Вибір програмного забезпечення.....	22
1.4.1 Keil uVision середовище розробки для мікроконтролерів .....	22
1.4.2 Графічний генератор проектів STM32CubeMX .....	24
1.5 Інтерфейси передачі даних, що використовуються для обміну даними з датчиками.....	25
1.5.1 Послідовний інтерфейс SPI .....	26
1.5.2 Послідовний інтерфейс I2C.....	27
2. Опис та оптимізація алгоритму стабілізації кута крену БПЛА .....	28
2.1 Алгоритми фільтрації сигналів .....	28
2.1.1 Фільтр ковзного середнього .....	28
2.1.2 Медіанний фільтр .....	29
2.1.3 Комплементарний фільтр .....	29
2.2 Запропонований алгоритм керування кутом крену .....	31

2.2.1 Розширений фільтр Калмана.....	33
2.3 Валідація запропонованого алгоритму у MATLAB .....	34
3 Технічна реалізація системи стабілізації безпілотного літального апарату літакового типу за креном .....	39
3.1 Отримання кута крену за допомогою MEMS датчиків .....	39
3.1.1 Підключення гіроскопа L3GD20 .....	39
3.1.2 Підключення мікросхеми LSM303DLHC .....	47
3.1.3 Об'єднання показань двох датчиків.....	52
3.2 Визначення передавальної функції між кутом відхилення елеронів та кутом крену.....	53
3.3 Налаштування параметрів регулятора в Simulink Matlab .....	57
3.4 Реалізація алгоритму дискретного ПД-регулятора мовою програмування Сі.....	58
3.5 Випробування системи .....	62
Висновки.....	64
Перелік джерел посилання .....	66
Додаток А – Код бібліотеки, що реалізує алгоритм ПД-регулятора .....	68
Додаток Б – Презентація.....	70

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БПЛА – Безпілотний літальний апарат

ІНС – Інерційна навігаційна система

НДР – науково-дослідна робота;

ПД – пропорційно-інтегрально-диференціальний;

ПЗ – програмне забезпечення;

ТЗ – технічне завдання;

ШІМ - широтно-імпульсна модуляція;

Крен – поворот об'єкта (судна, літака, фундаменту) навколо його поздовжньої осі;

Пропорційний інтегральний диференціальний регулятор: Пристрій, який використовується в системах автоматичного керування для підтримки заданого значення параметра, що вимірюється. ПД-регулятор вимірює відхилення стабілізованої величини від заданого значення (уставки) і видає керуючий сигнал, що є сумою трьох доданків, перше з яких пропорційно цьому відхиленню, друге пропорційно інтегралу відхилення і третє пропорційно похідної відхилення;

Стабілізація: Підтримка заданого постійного в часі значення однієї (або кількох) регульованої величини незалежно від зовнішніх (стосовно об'єкту) та внутрішніх обурювальних (дестабілізуючих) впливів, що прагнуть відхилити регульовану величину від заданого значення;

Широтно-імпульсна модуляція: Метод перетворення сигналу, у якому змінюється тривалість імпульсу (шпаруватість), а частота залишається константою.

## ВСТУП

Безпілотні літальні апарати стають все більш затребуваними для потреб промисловості, особливо ті, що мають високий запас ходу. Це пов'язано з тим, що останні десять років вартість придбання та обслуговування БПЛА значно знизилася, при одночасному збільшенні функціоналу та часу напрацювання на відмову.

Практично будь-яке завдання, яке раніше вирішувалося за допомогою залучення літака з екіпажем на борту, сьогодні може бути вирішене за допомогою дронів. При цьому сучасні БПЛА в порівнянні з класичними літаками не вимагають для запуску спеціальної злітної смуги, здатні працювати в автономному режимі, мають більшу маневреність, є більш екологічно безпечними, мають більш низьку вартість як обслуговування, так і придбання. У промислових підприємствах дрони використовуються для моніторингу територіальних володінь, таких, як магістральні лінії нафтопроводів.

У горизонтальному польоті (без зміни висоти і швидкості польоту) будь-яке відхилення крену від нейтрального положення призводить до таких небажаних наслідків, як зміна напрямку курсу польоту в бік крену і зниження висоти польоту. Чим більше крен, тим сильніше виражені обидва ці явища. Тому утримання крену в нульовому положенні є найважливішим завданням, безпосередньо пов'язаним з безпекою польоту, а також з їх комфортністю та економічністю. Політ літаком без автопілота крену вимагає від оператора більшого досвіду пілотування, а політ у негоду стає практично неможливим [1].

Метою роботи є розробка алгоритму стабілізації безпілотного літального апарату літакового типу по крену. Для досягнення поставленої мети необхідно вирішити такі завдання:

- визначення передавальної функції між кутом відхилення елеронів та кутом крену наявного безпілотного літального апарату літакового типу, використовуючи математичний пакет MATLAB;

- фільтрація даних MEMS датчиків;



- розробка алгоритму отримання кутового положення БПЛА літакового типу використовуючи гіроскоп і акселерометр;
- перевірка алгоритму стабілізації крену БПЛА літакового типу шляхом налаштування параметрів регулятора в математичному пакеті MATLAB;
- реалізація регулятора крену на базі мікроконтролера STM32F303VCT6.

# 1 ПРОЕКТУВАННЯ СИСТЕМИ СТАБІЛІЗАЦІЇ БЕЗПІЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ ЛІТАКОВОГО ТИПУ ЗА КРЕНОМ

## 1.1 Управління кутом крену

Для розуміння процесів, що виникають у польоті, необхідно ознайомитися з конструкцією безпілотного літального апарату. Оскільки у роботі було прийняте рішення обмежитися БПЛА літакового типу, через брак часу для досліджень в рамках виконання магістерської дипломної роботи, було проведено аналіз конструкції та особливостей керування кутом крену для саме цих БПЛА.

В останні роки спостерігається величезне зростання досліджень, спрямованих на стабілізацію безпілотного літального апарату та керування ним. Ця дослідницька стаття присвячена розробці системи керування для стабілізації кута крену БПЛА з нерухомим крилом. На ринку є кілька різних типів БПЛА, але БПЛА з нерухомим крилом є найпоширенішим, який використовується для спостереження та рятування. Як і інші БПЛА, БПЛА з нерухомим крилом також нелінійні за своєю природою, і їх стабілізація під час польоту є найважливішим завданням. У бічному керуванні автопілотом фіксуючим БПЛА стабілізація кута крену відіграє дуже важливу роль, і її правильне керування має важливе значення для стабільності БПЛА. У цій дослідницькій роботі пропонується алгоритм управління PID з розширеним фільтром Калмана для стабілізації кута крену БПЛА в умовах невизначеності. Параметри PID налаштовані таким чином, щоб крен стабілізувався якомога швидше та в межах обмеженого перерегулювання, а додавання розширеного фільтра Калмана (ЕКФ) робить БПЛА більш стабільним, усуваючи шуми, створювані системою та датчиками. Результати моделювання довели, що запропонована методика керування працює добре та дає ефективні результати для стабілізації кута крену БПЛА.

### 1.1.1 Конструкція БПЛА літакового типу

Безпілотні літальні апарати (БПЛА) нещодавно з великим успіхом використовувалися у військовій розвідці, забезпечуючи життєздатну альтернативу пілотованим літальним апаратам завдяки їх меншому розміру, зниженому ризику для життя та меншій вартості. Збройні сили використовують БПЛА в таких цілях, як патрулювання кордонів, розвідка безпеки, спостереження, захоплення цілей і охорона. Окрім військового застосування БПЛА, він також використовується в багатьох цивільних програмах, таких як пошуково-рятувальні місії, дослідження, охорона та обстеження нафтопроводів, горіння лісів, застосування в сільському господарстві та інспекція енергетичних і атомних станцій.

Літак із нерухомим крилом здатний літати за допомогою нерухомих крил, які створюють підйомну силу, викликану тягою/швидкістю транспортного засобу вперед і формою крил. Для динаміки польоту необхідно вказати інерціальну систему відліку, оскільки закони руху Ньютона дійсні лише тоді, коли прискорення вимірюється відносно інерціальної системи, яка фактично перебуває у спокої у Всесвіті.

Для опису просторового стану літака застосовується нормальна система координат. Ця система координат є три взаємоперпендикулярні осі, початок яких збігається з центром мас БПЛА (рис.1.1) [3].

Ось  $OY$  завжди спрямована вертикально вгору від центру землі, проходячи через центр мас літального апарату, а осі  $OZ$  і  $OX$  утворюють строго горизонтальну площину. Для опису положення у просторі прийнято використовувати кути тангажу, крену та нишпорення. Крен є поворотом літака щодо поздовжньої осі  $OX$ . Тангаж – поворот щодо поперечної осі  $OZ$ . Ризик – поворот щодо вертикальної осі  $OY$  [4].

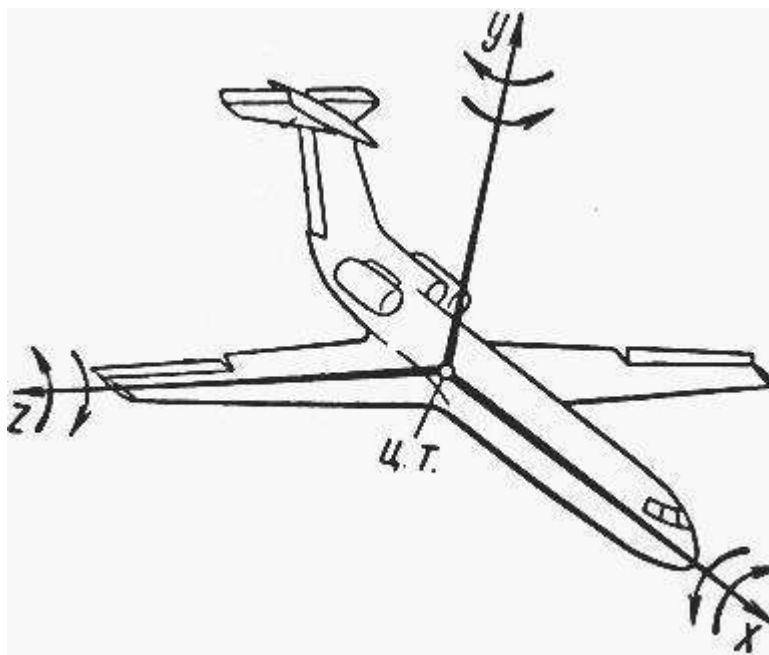


Рисунок 1.1 – Пов'язана система координат

Дрон є літальним апаратом з аеродинамічним принципом польоту. Двигун постійного струму, що обертає лопаті гвинта, створює повітряний потік, що набігає, на аеродинамічні поверхні БПЛА. Підйомна та керуючі сили утворюються за рахунок обтікання таких несучих поверхонь літака, як крило та хвостове оперення.

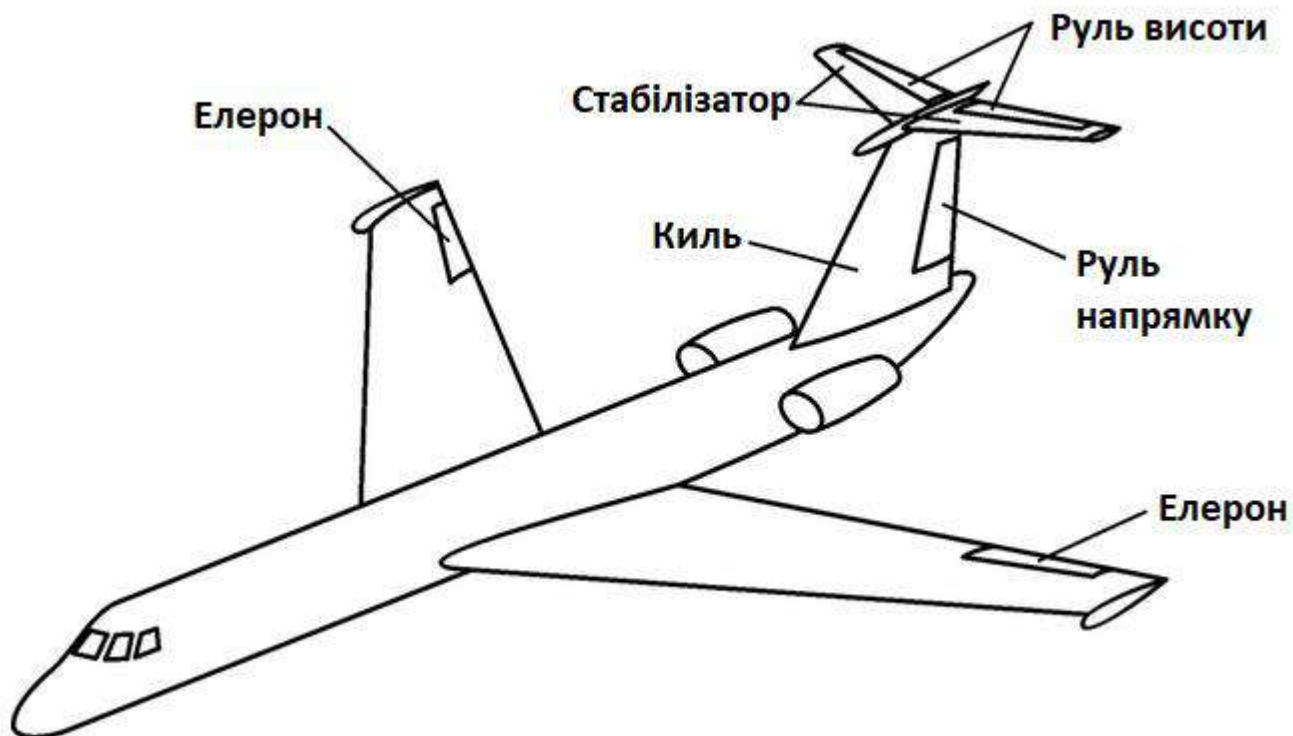


Рисунок 1.2 – Складові частини літака (БПЛА літакового типу)

Головною несучою частиною літака є фюзеляж. Він служить кріпленням для решти конструктивних елементів, таких як: крило, двигун, хвостове оперення, бортова камера. Усередині фюзеляжу розташовані: польотний контролер, радіоапаратура керування та літій-іонні акумулятори. Корпус літака виконаний з пінополістиролу, з подальшим посиленням конструкції за допомогою плівкової стрічки з клейовим покриттям. Матеріали, що застосовуються, дозволяють створити легку і при цьому досить міцну конструкцію.

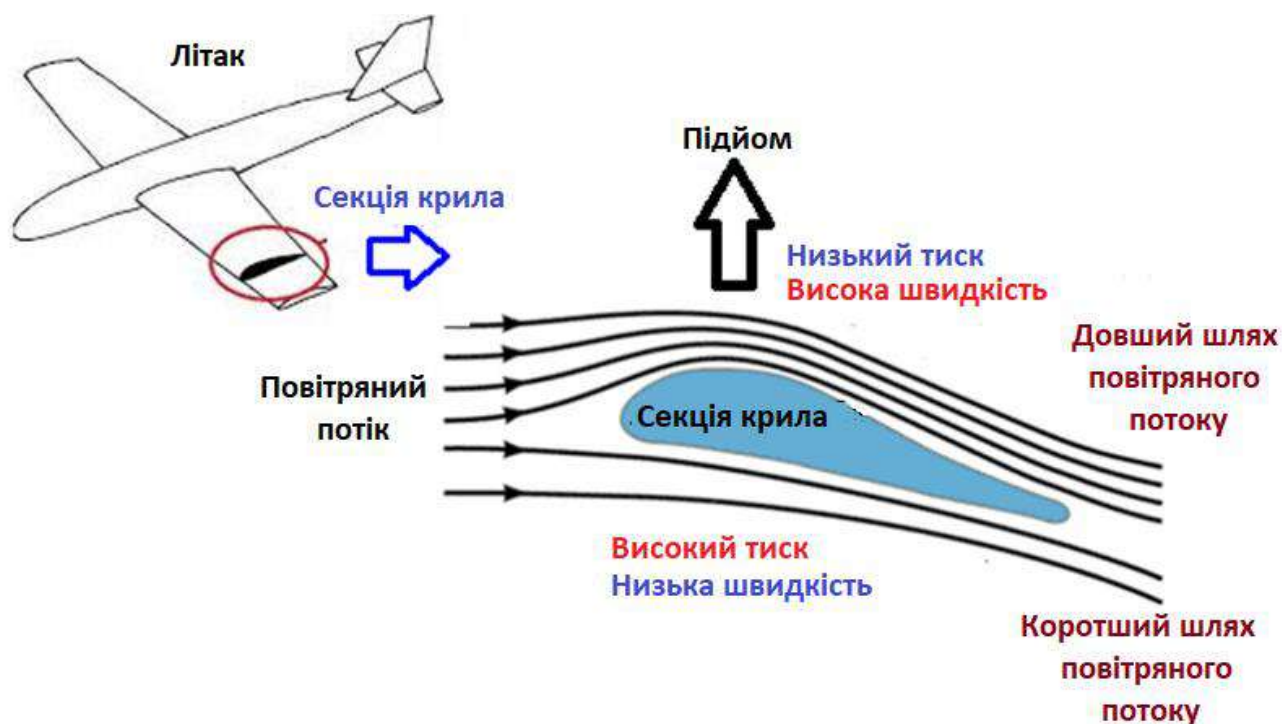


Рисунок 1.3 – Принцип аеродинамічної підйомної сили

Утворення підйомної сили на крилі є різниця швидкостей течії повітря (рідини) над крилом і під крилом, і внаслідок цього виникає перепад тиску згідно закону Бернуллі. При цьому однозначно пов'язують через закон Бернуллі розрахункову швидкість потоку на поверхні крила з тиском, що інструментально реєструється на крило, ігноруючи інші можливі пояснення на основі не менш базових законів фізики.

Як було зазначено вище, підйомна сила утворюється з допомогою обтікання крила повітряними масами. Профіль крила спроектований так, щоб розділяти повітряний потік, що набігає на нього. Таким чином утворюється два потоки розташованих один вище, а інший нижче крила. Потік над кромкою крила має

більшу швидкість ніж під ним. За рахунок чого за рівнянням Бернуллі утворюється область низького тиску зверху та надмірного знизу поверхні крила. Зрештою різниця цих тисків створює підйомну силу крила. Аеродинамічні властивості крила залежать від його геометричних розмірів, таких як розмах крила та його площа.

На крилі встановлені аеродинамічні органи управління, які називаються елеронами. Відхиляючись диференціально (тобто у протилежні сторони), вони здатні керувати кутовою швидкістю обертання літака навколо поздовжньої осі. Так, наприклад, якщо на лівому півкрилі відхилити елерон вгору, а на правому відповідно донизу, то на лівій половині крила підйомна сила зменшиться, а на правій – збільшиться.

В результаті виникне момент навколо поздовжньої осі літака  $\Delta M_x$  і літак нахилиться на ліве напівкрило (рис.1.4) [5].

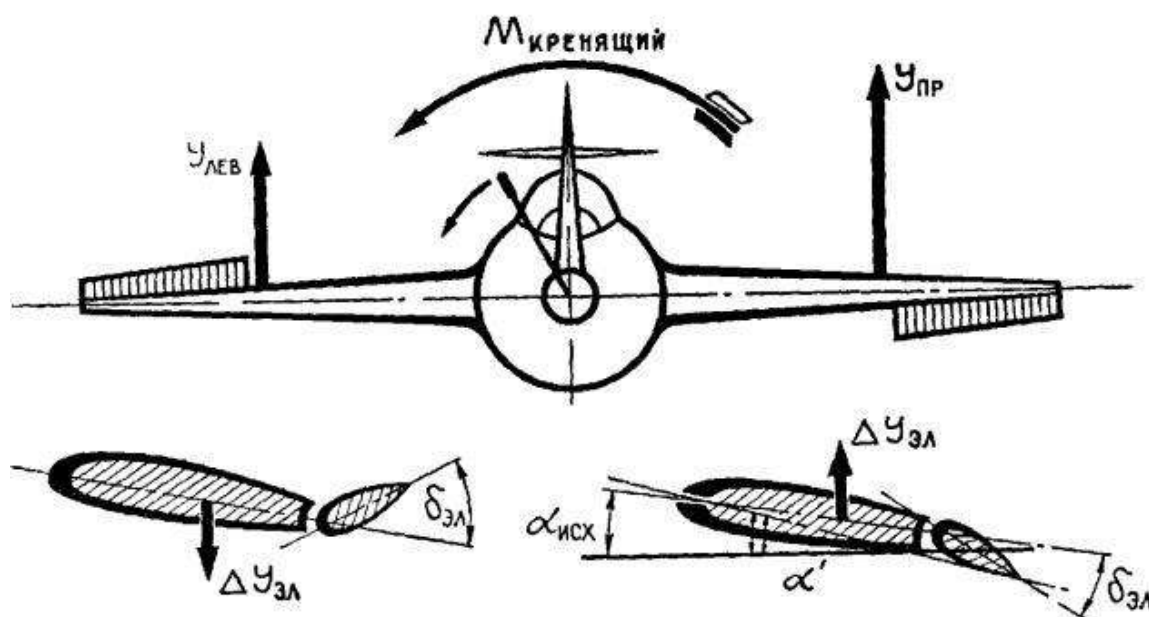


Рисунок 1.4 – Принцип дії елеронів

Хвостове оперення складається з вертикально спрямованого кіля та горизонтального стабілізатора. Кіль служить задля забезпечення стійкості літака у ході його руху, а стабілізатор — кутом тангажу [2]. Управління кутами тангажу та нишпорення здійснюється за допомогою розташованих у хвостовій частині літального апарату кермів висоти та нишпорення.

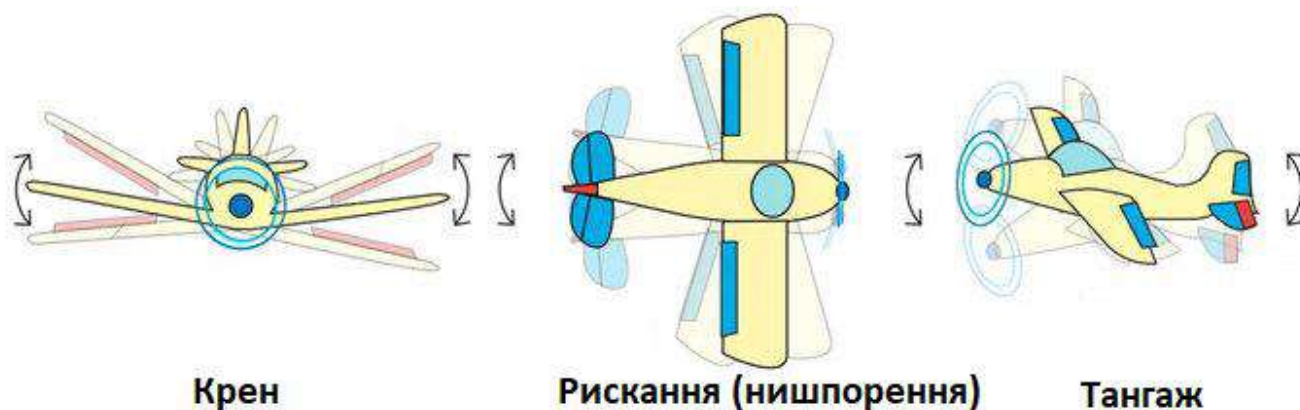


Рисунок 1.5 – Відхилення літака по осях

Щоб мати можливість пов'язати обертання та рух БПЛА літаково типу використовується матриця:

$$R_B^n = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (1.1)$$

Вектор швидкості в системі Body розкладається в системі NED як:

Шість ступенів свободи (6 DOF) жорстке тіло. Сили та моменти тіла в рамі BODY визначаються як:

$$\bar{t}_{RB} = \begin{bmatrix} f_0^b \\ m_0^b \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \text{forward force} \\ \text{Sideways force} \\ \text{Vertical force} \\ \text{Roll moment} \\ \text{Pitch moment} \\ \text{yaw moment} \end{bmatrix} \quad (1.2)$$

Де  $f_0^b$  і  $m_0^b$  – сили і моменти, які є сумою аеродинамічних (зі збуренням), гравітаційних і приводних сил і моментів.

Рівняння (1.3) є виразом, який допомагає визначити шість ступенів свободи (6 DOF) для літального апарату з нерухомим крилом:

$$M_{RB} \dot{v} + C_{RB}(v) v + g(\eta) = I + \omega \quad (1.3)$$

де,

$M$  – інерційна матриця системи;

$C(v)$  – доцентрова матриця Коріоліса;

$D(v)$  – матриця демпфування;

$g(\eta)$  – вектор сил і моментів гравітації;

$i$  – вектор керуючих входів;

$g$  – вектор для попереднього обрізання;

$\omega$  – вектор збурень/

Вираз рівняння (1.3) оцінюється, як показано в рівнянні (1.4). Форма компонента може бути виражена як:

$$\left. \begin{aligned} X &= m (\dot{U} + QW - RV + g \sin \theta) \\ Y &= m (\dot{V} + UR - PW - g \cos \theta \sin \theta) \\ Z &= m (\dot{W} + PV - QU - g \cos \theta \cos \theta) \\ L &= I_x \dot{P} - I_{xz} (\dot{R} + PQ) + (I_z - I_y) QR \\ M &= I_y \dot{Q} - I_{xz} (P^2 - R^2) + (I_x - I_z) PR \\ N &= I_z \dot{R} - I_{xz} \dot{P} + (I_y - I_x) PQ + I_{xz} QR \end{aligned} \right\} \quad (1.4)$$

Рівняння (1.4) представляє сили, а рівняння моменту керує рухом літака відносно інерціальної системи.

## 1.2 Структура системи стабілізації крену БПЛА

Автопілоти крену виконують стабілізацію обертання БПЛА навколо поздовжньої осі на траєкторії його проходження, тим самим приводячи літак у горизонтальний політ. Управління креном є одним із найпростіших випадків



управління обертанням центру мас літального апарату. Так як при такому русі величина моментів і сил, що діють на літак, залежить від значення кута крену [6].

Контур стабілізації складається з обчислювального пристрою у вигляді мікроконтролера, виконавчого пристрою – цифровий сервопривід, безпілотного літального апарату, гіроскопа та акселерометра, які розташовані в ланці зворотного зв'язку регулятора. Регулювання кута нахилу літака здійснюватиметься за ПІД-законом управління. Структурна схема системи стабілізації кута крену представлена на рис. 1.6.



Рисунок 1.6 – Структурна схема системи стабілізації кута крену ЛА

Принцип роботи: кут уставки крену задається один раз при ініціалізації параметрів регулятора і протягом польоту не змінюється, дані про поточний кут крену виходить за допомогою показань гіроскопа та акселерометра. Сигнал неузгодженості між поточним кутом і уставкою надходить на вхід ПІД-регулятора, який у свою чергу видає сигнал на сервоприводи.

Крен – це рух крил літака вгору та вниз, при крені літак обертається навколо свого центру ваги, точки, яка є середнім розташуванням маси літака. Поворотний рух викликається відхиленням елерона літака. Для БПЛА з нерухомим крилом, що летить на постійній висоті, для керування курсом часто використовується команда кута крену. Тому для контролю швидкості руху необхідно контролювати кут крену. Бічне рівняння руху для управління елероном задано рівняннями нижче.

$$\begin{aligned}
(m_1 s - b C_{y\beta} s - C_{y\beta}) \Delta \bar{\beta}(s) - (b C_{yp} s + C_{y\dot{\Phi}}) \Delta \bar{\Phi}(s) + (m_1 - b C_{yr}) s \Delta \bar{\Psi}(s) &= \\
C_{y\delta_a} \Delta \bar{\delta}_a(s) & \\
(-C_{l\beta} - b C_{l\beta} s) \Delta \bar{\beta}(s) + (-C_{lp} b s - I_{xl} s^2) \Delta \bar{\Phi}(s) - (C_{lr} b s + I_{xz} s^2) \Delta \bar{\Psi}(s) &= \\
C_{l\delta_a} \Delta \bar{\delta}_a(s) & \\
(-C_{n\beta} - b C_{n\beta} s) \Delta \bar{\beta}(s) + (-C_{np} b s - I_{xz} s^2) \Delta \bar{\Phi}(s) + (-C_{nr} b s + I_{zl} s^2) \Delta \bar{\Psi}(s) &= \\
C_{n\delta_a} \Delta \bar{\delta}_a(s) &
\end{aligned}$$

де,

$C_x, C_z, C_m$  аеродинамічні коефіцієнти;

$\beta, \Phi, \Psi$  поздовжня змінна;

$s$  – контрольна область;

$c$  – аеродинамічна хорда;

$b$  – розмах крил;

$C_l, C_n$  – використовується для визначення коефіцієнтів моменту тангажу та нишпорення.

$C_m$  – аеродинамічний коефіцієнт хорди.

$C_{y\{\Phi, \beta, p, r\}}, C_{l\{r, \beta, p\}}, C_{n\{r, \beta, p\}}$  – похідні стійкості.

$C_{m\delta_e}, C_{mq}$  – похідні контролю.

Припускаючи, що елерон є єдиним боковим керуванням, це можна виразити у рівнянні (1.6).

Розв'язавши вищевказані рівняння, передатну функцію системи кренів.

$$\frac{\Delta \bar{\Phi}(s)}{\Delta \bar{\delta}_a(s)} = \frac{\begin{vmatrix} (m_1 s - b C_{y\beta} s - C_{y\beta}) & C_{y\delta_a} & (m_1 - b C_{yr}) s \\ (-C_{l\beta} - b C_{l\beta} s) & C_{l\delta_a} & -(C_{lr} b s + I_{xz} s^2) \\ (-C_{n\beta} - b C_{n\beta} s) & C_{n\delta_a} & -(C_{nr} b s - I_{zl} s^2) \end{vmatrix}}{\Delta_{lat}(s)} \quad (1.6)$$

$$\Delta_{lat}(s) = \begin{vmatrix} (m_1 s - b C_{y\beta} s - C_{y\beta}) & - (b C_{yp} s + C_{y\dot{\phi}}) & s(m_1 - b C_{yr}) \\ (-C_{l\beta} - b C_{l\beta} s) & - (C_{lp} b s + I_{xl} s^2) & - (C_{lr} b s + I_{xz} s^2) \\ (-C_{n\beta} - b C_{n\beta} s) & - (C_{np} b s + I_{xz} s^2) & - (C_{nr} b s - I_{zl} s^2) \end{vmatrix}$$

Після розв'язання та перетворення рівняння (1.6) у рівняння Лапласа ми отримуємо

$$\frac{\Delta \bar{\Phi}(s)}{\Delta \bar{\delta}_a(s)} = \frac{k_1 s^3 - k_2 s^2 + k_3 s}{k_4 s^5 + k_5 s^4 + k_6 s^3 + k_7 s^2 + k_{10} s} \quad (1.7)$$

де  $k$  – коефіцієнти БПЛА.

### 1.3 Компоненти системи стабілізації кута крену

Після того як було визначено структурну схему системи стабілізації кута крену необхідно підібрати комплектуючі системи. Обладнання вибиралося виходячи з його функціональних та цінових характеристик. Як основний обчислювальний пристрій був обраний мікроконтролер STM32F303VCT6. Як кутові датчики використовується акселерометр LSM303DLHC, а як гіроскоп L3GD20 виробництва компанії ST Microelectronics. Власне, сам безпілотний літак виготовили власними силами.

#### 1.3.1 Мікроконтролер STM32F303VCT6

Як польотний контролер використовується мікроконтролер STM32F303VCT6, виконаний на базі налагоджувальної плати STM32 Discovery F3. Цей вибір був зроблений в першу чергу через те, що дане сімейство мікроконтролерів оснащується вбудованим блоком FPU (Floating Point Unit) для роботи з числами з точкою, що плаває. Це значно збільшує швидкість роботи з даними типу float або double.

Мікроконтролер побудований на базі 32-розрядного RISK ядра Arm Cortex - M4, що працює на частоті 72 МГц. На борту контролера є власна пам'ять (ПЗП) об'ємом 256 кілобайт, а також 40 кілобайт оперативної пам'яті.

Також важливу роль у виборі саме цієї плати відіграла наявність на ній двох MEMS-мікросхем: триосьового гіроскопа (L3GD20) та акселерометра LSM303DLHC. Мала вага, яка становить 32 грами, дозволяє без проблем розмістити плату на борту літака. Зовнішній вигляд плати представлено на рис.1.7.

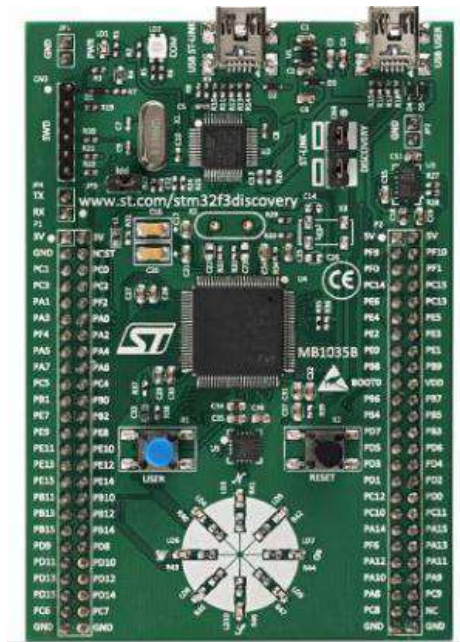


Рисунок 1.7 – Зовнішній вигляд плати розробника Discovery F3 для МК STM32F303VCT6

Крім іншого мікроконтролер STM32F303VCT має наступні переваги:

- робочі температури у діапазоні від мінус 40 до 80°C;
- невисока вартість виробу;
- висока продуктивність;
- мале енергоспоживання.

### 1.3.2 MEMS датчики L3GD30 та LSM303DLHC

Мікросхема L3GD20 – це енергоефективний триосьовий датчик швидкості кутових обертань, виконаний у корпусі LGA-16. Гіроскоп має такі характеристики [8]:

- наявність трьох шкал вимірювання;
- підтримка інтерфейсів передачі даних I2C та SPI;
- вихідна величина в 16 бітному форматі;
- наявність вбудованого низько-і високочастотного фільтра, з можливістю налаштування смуги пропускання;
- напруга живлення в діапазоні від 2,4 до 3,6 В;
- вбудований режим нікого енергоспоживання та сплячий режим
- діапазон робочих температур від мінус 40 до 85 °С.

У цій роботі реалізовуватиметься підключення гіроскопа за інтерфейсом SPI, оскільки воно розпаяно на платі STM32 Discovery.

Мікросхема LSM303DLHC - являє собою систему з триосьового датчика лінійних прискорень і терхосевого датчика магнітних полів, виконаних в єдиному корпусі LGA-14. Для вимірювання кута крену використовується лише акселерометр. Він має наступні характеристики [7]:

- вимірювання прискорення в діапазонах  $\pm 2$  g;  $\pm 4$  g;  $\pm 8$  g;  $\pm 16$  g;
- вихідна величина в 16 бітному форматі;
- використання послідовного інтерфейсу I2C;
- наявність енергозберігаючого режиму роботи;
- наявність вбудованого датчика температури;
- напруга живлення в діапазоні від 2,16 до 3,6 В;
- датчик може бути налаштований на генерацію переривання від руху пристрою;
- діапазон робочих температур від мінус 40 до 85°С.

## **1.4 Вибір програмного забезпечення**

Програмування та налаштування вибраного обладнання здійснюється з використанням наступного програмного забезпечення:

- Keil  $\mu$ Vision 5 – це середовище використовується для програмування мікроконтролера через вбудований програматор ST link v2.
- STM32CubeMX – це програмне забезпечення застосовується для конфігурації мікроконтролера, автоматично генеруючи необхідні файли для проекту Keil.

### **1.4.1 Keil uVision середовище розробки для мікроконтролерів**

Keil  $\mu$ Vision – це інтегроване середовище розробки, що є повним комплексом програмного забезпечення необхідним для програмування та налагодження мікроконтролерів серії STM32 [10].

Keil  $\mu$ Vision дозволяє працювати з проектами високого ступеня складності, дозволяючи як вносити зміни у вихідний текст, так і виконувати внутрішньосхемне налагодження програми та програмувати ПЗП мікроконтролера. Приховування більшості другорядних функцій від розробника дозволяє значно розвантажити інтерфейс і зробити його інтуїтивним. Однак за необхідності розв'язання складних завдань можна задіяти весь потенціал модулів, що працюють під єдиною оболонкою. Нижче наведено основні програмні засоби Keil  $\mu$ Vision [9]:

- вбудована база даних мікроконтролерів із вмістом інформації про всі підтримувані пристрої. Вона зберігає посилання джерела інформації та дані конфігурацій (При додаванні нового пристрою в поле Project Target всі його опції додаються автоматично);

- менеджер проектів, який служить для об'єднання окремих вихідних та заголовних файлів у групи, що обробляються за єдиними правилами (Таке групування дозволяє легше орієнтуватися серед множини файлів);

- вбудований редактор коду полегшує процедуру набору вихідного тексту за рахунок виділення синтаксичних помилок кольором (редагування залишається доступним у режимі налагодження коду, дозволяючи виділяти або одразу виправляти проблемні ділянки коду);

- засоби автоматичної компіляції, асемблювання та компоновання, які служать для створення завантажувального модуля програми;

- відладчик-симулятор, призначений для налагодження скомпільованої програми на віртуальній моделі мікроконтролера (Моделювання роботи ядра здійснюється на досить високому рівні достовірності).

Знімок екрана програми Keil  $\mu$ Vision представлений на рис. 1.8.

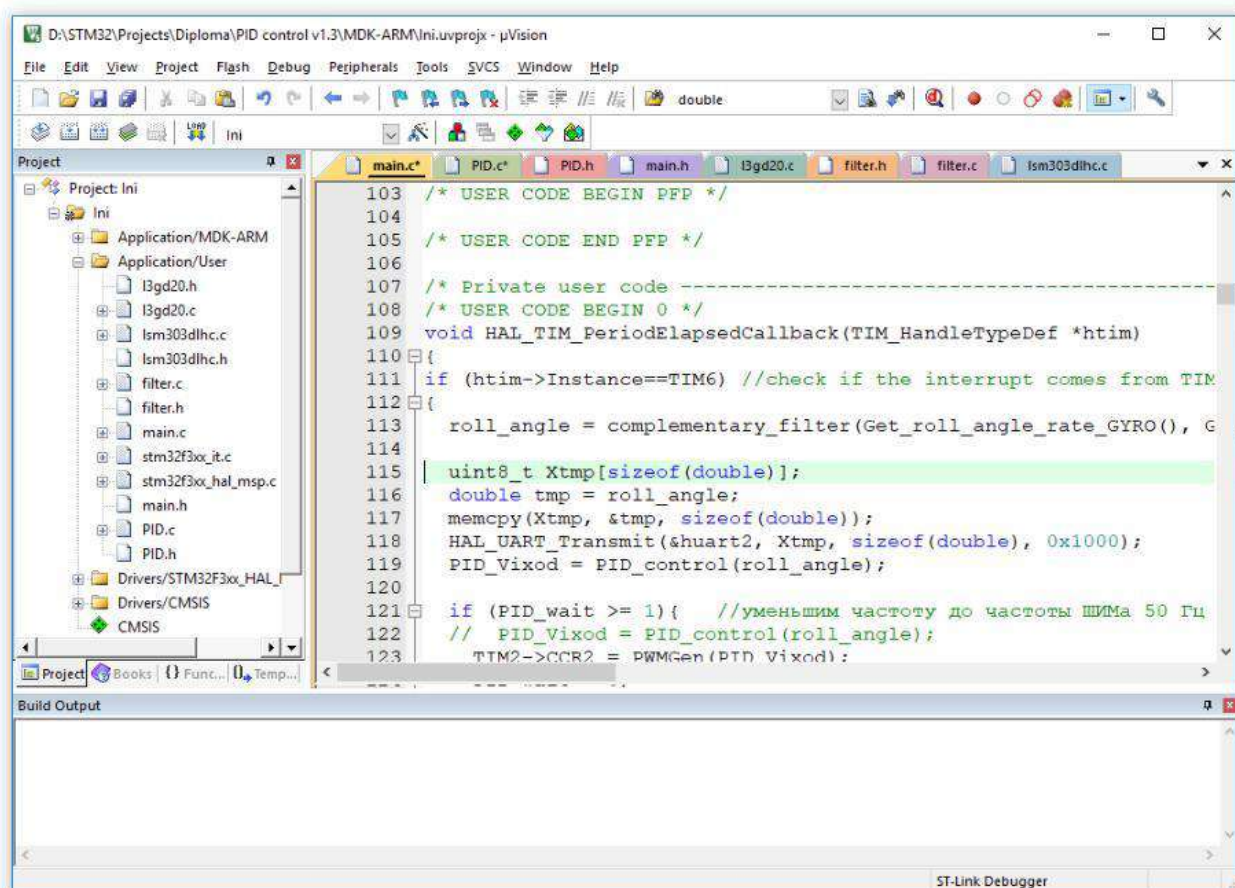


Рисунок 1.8 – Знімок екрану Keil  $\mu$ Vision

## 1.4.2 Графічний генератор проектів STM32CubeMX

STM32CubeMX – це графічний інструмент, який дозволяє здійснювати конфігурацію мікроконтролерів та мікропроцесорів серії STM32, а також генерувати відповідний Сі код для ядра Arm® Cortex®-M [12].

Візуальний редактор є складовою інфраструктури STM32Cube, яка містить спеціальні бібліотеки для забезпечення працездатності з різними лінійками мікроконтролерів STM32. Встановлювати необхідні бібліотеки можна безпосередньо в редакторі через вбудований завантажувач, при цьому комп'ютер має бути підключений до Інтернету. Також є можливість відстеження актуальних версій бібліотек, які вже встановлені на комп'ютер.

Пакет програмного забезпечення STM32CubeMX включає наступні основні компоненти:

- бібліотеку CMSIS (Cortex Microcontroller Software Interface Standard). Незалежний від вендора рівень абстракції для серії процесорних ядер Cortex®-M, що визначає інтерфейси для ядра і його периферію;

- бібліотеку HAL (Hardware Abstraction Layer) (Є бібліотекою для роботи з периферійним обладнанням. Вона дозволяє позбавитися роботи безпосередньо з регістрами мікроконтролера, а працювати безпосередньо з необхідними об'єктами. Таким чином, код, написаний з використанням бібліотеки HAL, може бути легко переписаний для використання на іншому мікроконтролері. Саме за допомогою цієї бібліотеки було написано основну керуючу програму);

- бібліотеки, що підтримують популярні плати, такі як STM32 discovery;

- бібліотеки вищого прикладного рівня, що реалізують підтримку файлової системи FAT, протоколи передачі TCP/IP, операційних систем реального часу (FreeRTOS), інтерфейсу USB.

Основні завдання, які можуть бути вирішені з використанням цього програмного забезпечення, представлені нижче:

- налаштування тактування, портів, таймерів. Також можливий вибір способу тактування мікроконтролера;



- налаштування режимів роботи портів;
- налаштування режимів роботи таймерів;
- розрахунок споживання енергії.

Знімок робочого екрану програми STM32CubeMX представлено на рис.1.9.

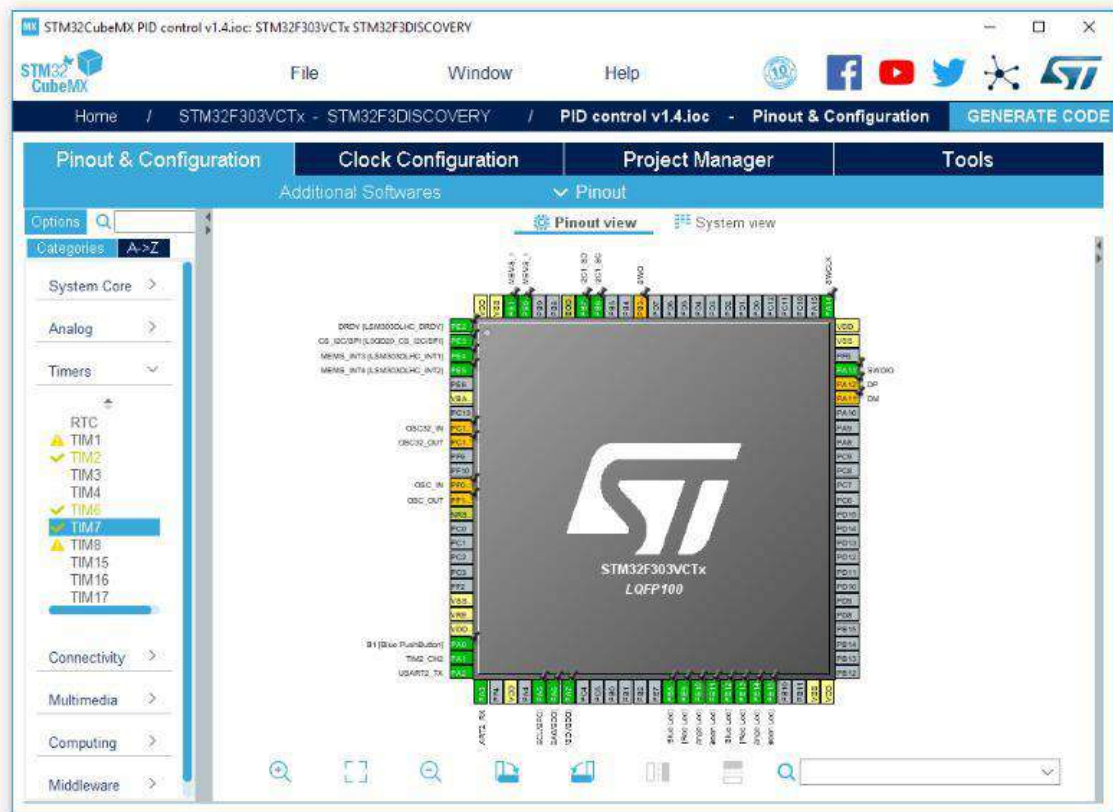


Рисунок 1.9 – Знімок екрану програми STM32CubeMX

## 1.5 Інтерфейси передачі даних, що використовуються для обміну даними з датчиками

Підключення гіроскопа L3GD20 здійснюється за допомогою послідовного SPI інтерфейсу, а мікросхема LSM303DLHC за допомогою I2C.

### 1.5.1 Послідовний інтерфейс SPI

Serial Peripheral Interface (spi) – інтерфейс для послідовного обміну даними між мікроконтролером і периферією. У SPI є один провідний пристрій, один або кілька ведених. Для передачі даних використовуються 3 лінії [13]:

- MISO (Master Input Slave Output) – цією лінією здійснюється передача даних від веденого (Slave) пристрою до ведучого (Master).

- MOSI (Master Output Slave Input) – по цій лінії провідний пристрій (Master) надсилає дані веденому (Slave).

- SCK (Serial Clock) – дана лінія служить передачі тактируючого сигналу веденому пристрою.

- за наявності кількох Slave пристроїв на шині необхідно використовувати лінію SS (Slave Select), що дозволяє вибрати керований пристрій.

Протокол запису та читання датчика кутових швидкостей L3GD20 представлений на рис. 1.10 [8].

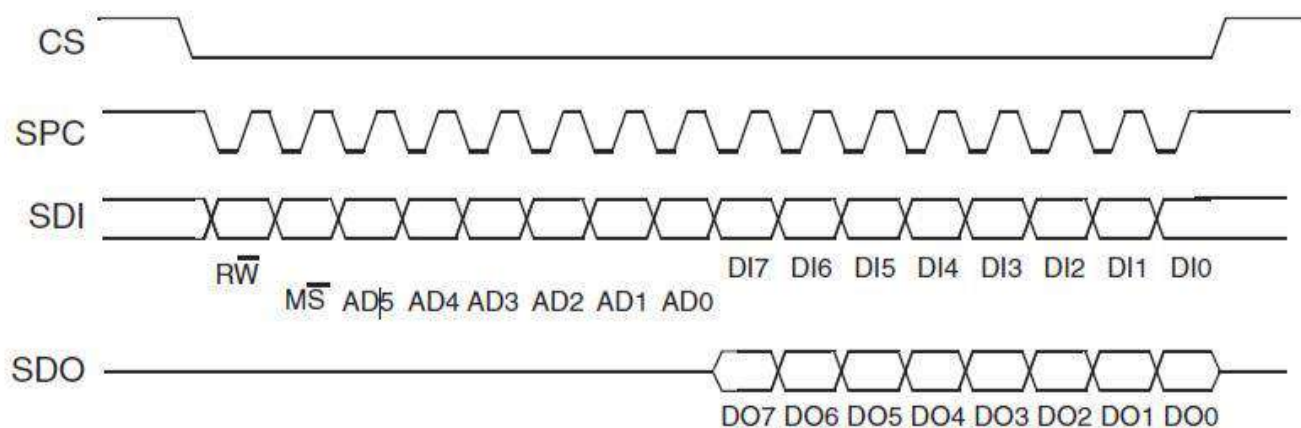


Рисунок 1.10 – Протокол запису та читання L3GD20

У цьому випадку реалізовано чотири дротове підключення: CS, SPC, SDI, SDO. CS - це лінія вибору Slave пристрою, при передачі даних необхідно щоб на лінії був сигнал, що відповідає логічному нулю. По лінії SPC здійснюється тактування. А лінії SDI та SDO служать для прийому та передачі даних відповідно.

## 1.5.2 Послідовний інтерфейс I2C

Inter-Integrated Circuits (I2C) – послідовний протокол обміну даними, який використовує передачі двох двонаправлених ліній зв'язку [14].

У мережі може бути кілька провідних (Master) пристроїв, які генерують сигнал і ініціалізують передачу даних. Також у мережі може бути кілька ведених (Slave) пристроїв, що мають кожен свою унікальну адресу. До однієї шини може бути підключено до 127 пристроїв. На рис. 1.11 представлено підключення пристроїв до шини [14].

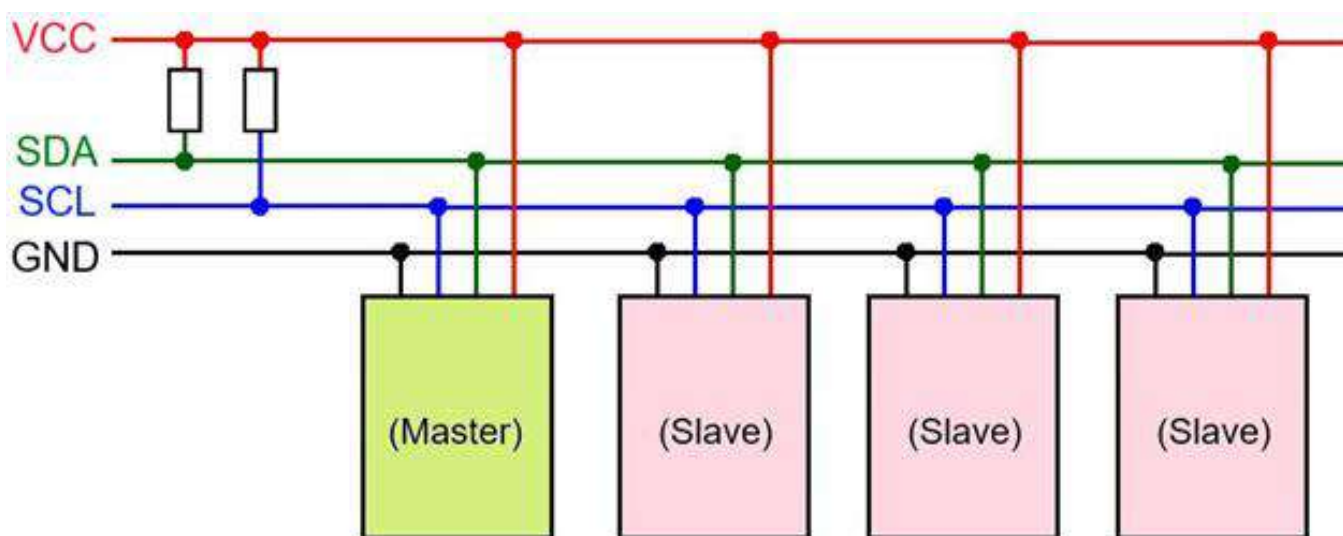


Рисунок 1.11 – Підключення пристроїв до шини I2C

Передача даних ініціалізується провідним пристроєм. Для цього він повинен почати передавати тактуючі імпульси лінії SCL. Одночасно по лінії SDA надсилається адреса пристрою, якого виконується звернення, що складається з 7 біт. Далі по цій же лінії надсилається код операції (запис або читання) і ще один біт, званий бітом підтвердження, що сигналізує про те, що ведений пристрій прийняв запит. У разі відсутності біта підтвердження провідний пристрій припиняє виконання запиту. Після встановлення зв'язку здійснюється читання чи запис.

## 2. ОПИС ТА ОПТИМІЗАЦІЯ АЛГОРИТМУ СТАБІЛІЗАЦІЇ КУТА КРЕНУ БПЛА

### 2.1 Алгоритми фільтрації сигналів

У процесі зняття сигналів МЕМС-датчиків отримана вихідна величина виходить зашумленою. Для отримання корисного сигналу із зашумленого вихідного необхідно зробити його фільтрацію. У роботі використовується фільтр ковзного середнього, медіанний фільтр та комплементарний фільтр.

#### 2.1.1 Фільтр ковзного середнього

Фільтр ковзного середнього - фільтр, значення якого в кожній точці дорівнює середньому арифметичному за певний проміжок часу до поточного значення. Ковзаючі середні часто використовуються для усунення коливань щодо корисного сигналу, дозволяючи виявити тенденції або цикли. Значення фільтра на кроці розраховується за формулою (2.1).

$$Y[n] = \frac{\sum_{i=n-N+1}^n X[i]}{N} \quad (2.1)$$

де  $X[i]$  – зашумлений сигнал на  $i$ -му такті роботи;

$Y[n]$  – відфільтрований сигнал на  $n$ -му такті;

$N$  – параметр настроювання.

З формули (2.1) видно, що на відфільтроване значення впливають лише  $N$  попередніх вимірювань, тобто даний фільтр має кінцеву імпульсну характеристику. Цей фільтр простий у реалізації, не вимагає великих обчислювальних ресурсів мікроконтролера, його можна легко налаштувати, але при цьому при великому згладжуванні  $N$  має велику затримку щодо сигналу, що фільтрується. Крім того, цей тип фільтра погано підходить для фільтрації імпульсної перешкоди.

### 2.1.2 Медіанний фільтр

Медіанний фільтр – один із видів цифрових фільтрів. Даний фільтр набув широкого поширення в цифровій обробці сигналів зменшення рівня шумів. Він також здатний з високою ефективністю обробляти сигнали, схильні до впливу імпульсних перешкод.

Фільтрація відбувається так: вихідні значення сортуються в порядку зростання значення; і значення, що знаходиться всередині упорядкованого списку, надходить на вихід фільтра. У разі парного числа відліків у вікні вихідне значення фільтра дорівнює середньому значенню двох відліків у середині упорядкованого списку. Вікно переміщається вздовж сигналу, що фільтрується, і обчислення повторюються на кожній ітерації.

Відфільтрована величина обчислюється за такою формулою:

$$Y[n]=med(sort(X[(n+1-M)..n])) \quad (2.1)$$

де  $X[(n+1-M)..n]$  – масив значень зашумленого сигналу;

$sort(X[(n+1-M)..n])$  – відсортований масив значень зашумленого сигналу;

$med(sort(X[(n+1-M)..n]))$  – значення зашумленого сигналу, що знаходиться на центральній позиції у відсортованому масиві (якщо кількість значення парна, то напівсума середніх);

$Y[n]$  – відфільтрований сигнал на n-му такті;

$M$  – настроювальний параметр фільтра.

### 2.1.3 Комплементарний фільтр

Як відомо, гіроскоп вимірює кутову швидкість обертання щодо трьох осей. Щоб отримати значення кута орієнтації датчика необхідно виконати процедуру інтегрування сигналу гіроскопа з часом. Точні гіроскопи, такі як кільцевий лазер, занадто дорогі і має занадто велику вагу для застосування на борту використовуваного БПЛА. З іншого боку, є менш точний напівпровідникові МЕМС-

гіроскопи, які позбавлені недоліків лазерних гіроскопів. Однак, інтегрування даних з таких датчиків призводить до накопичення помилки у обчисленні кута орієнтації та з часом помилка зростає. Таким чином, використовуючи лише гіроскоп не вийде виміряти кутове положення пристрою.

Акселерометр, у свою чергу, здатний вимірювати гравітаційне поле планети, і відповідно до розкладання гравітаційної сили на три осі стає можливим визначити абсолютне значення орієнтації в просторі. Тим не менш, і він дуже сильно піддається впливу перешкод. Наприклад, додаткове прискорення, викликане рухом датчика, неодмінно призведе до зсуву контрольованого напрямку сили тяжіння, що у свою чергу призведе до неправильних значень кутового положення. Для вирішення цієї проблеми одержувані показання акселерометра фільтрують низькочастотним фільтром. Даний фільтр здатний усунути високочастотний шум, проте при цьому відбувається збільшення затримки сигналу щодо відфільтрованого. Для компенсації цієї затримки використовують показання гіроскопа.

Комплементарний фільтр поєднує високочастотний фільтр для гіроскопа і низькочастотний фільтр для акселерометра і магнітометра. На відміну від робіт, де використовується повна матриця кутових станів, у цій роботі використовується лише одне значення, яке визначає орієнтацію пристрою щодо горизонтальної осі. Тому комплементарний фільтр має дещо змінений вигляд, представлений на рис.2.1, проте принцип його роботи залишається незмінним.

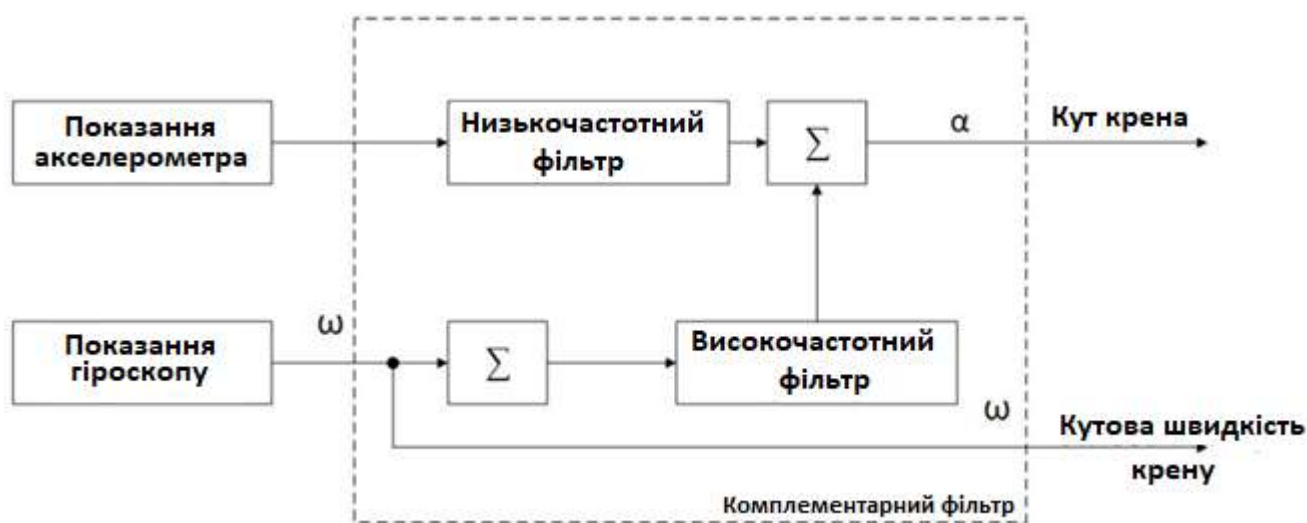


Рисунок 2.1 – Структурна схема комплементарного фільтра

Спочатку кожної ітерації роботи фільтра з кожного сенсора знімаються поточні показання, які разом з результатами, отриманими на попередній ітерації, надходять на вхід фільтра. Нижче наведено формулу для обчислення цього кута:

$$\varphi(i + 1) = (1 - \alpha) * (\varphi(i) + \omega_{GYRO}(i + 1) * dt) + \alpha * \varphi_{ACCEL}(i + 1) \quad (2.2)$$

де  $\varphi(i+1)$  – значення фільтрованого кута крену в момент часу  $t=i+1$ ;

$\varphi(i)$  – значення фільтрованого кута крену в момент часу  $t=i$ ;

$\varphi_{GYRO}(i+1)$  – кутова швидкість крену, отримана з гіроскопа в момент часу  $t=i+1$ ;

$\varphi_{ACCEL}$  – значення кута крену, отримане з акселерометра в момент часу  $t=i+1$ ;

$\alpha$  – коефіцієнт фільтра (значення знаходиться в діапазоні від 0 до 1);

$dt$  – періодичність опитування гіроскопа.

Регулюючи коефіцієнт настроювання фільтра, можна змінювати співвідношення між впливом на вихідну величину значень гіроскопа і акселерометра.

## 2.2 Запропонований алгоритм керування кутом крену

Мета цього дослідження полягає в розробці покращеного алгоритму та контролера, який підтримує стабільність руху безпілотного літального апарату по крену відповідно до бажаного вхідного сигналу. ПД-регулятор дає хороші результати щодо керованості руху крену, але він має певні обмеження, оскільки алгоритм керування недостатньо хороший, щоб працювати в умовах шуму, спричинених датчиками або системою, тому для усунення шумів використовується розширений фільтр Калмана (ЕКФ) пропонується з ПД-регулятором у даній дослідницькій роботі. На рис. 2.2 показана загальна блок-схема запропонованого алгоритму керування для керування рухом кута крену. На цій діаграмі ЕКФ використовується для усунення шумів датчика зворотного зв'язку, а автоматично

налаштований ПІД-регулятор використовується для отримання бажаної реакції на форму кута крену моделі системи БПЛА з нерухомим крилом.

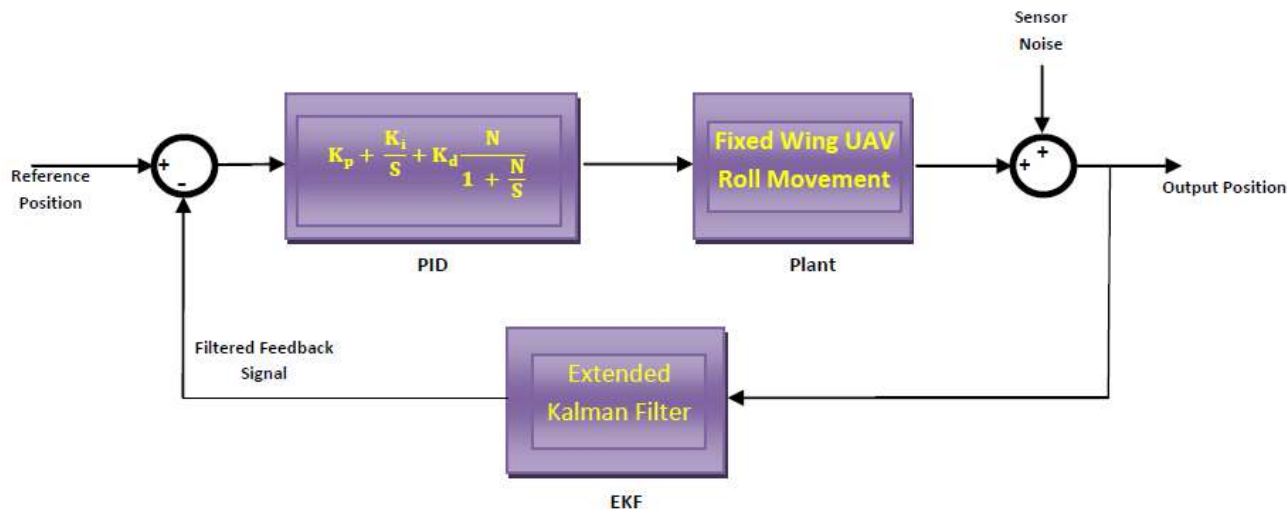


Рисунок 2.2 – Блок-схема системи керування зворотним зв'язком із ЕКФ

Щоб отримати потрібний кут повороту, сигнали помилки «e» для ПІД-регулювання:

$$e_{\bar{\phi}} = \bar{\Phi}_{\text{ref}} - \bar{\Phi}$$

А рівняння ПІД-регулятора можна записати так:

$$U(s) = K_p e(s) + \frac{K_i}{s} e(s) + K_d \frac{N}{1 + \frac{N}{s}} e(s) \quad (2.3)$$

Параметри ПІД-регулятора вибираються методом автоналаштування з оптимальним налаштуванням.

Застосування ПІД-регулятора до системної моделі кута крену, добуток рівняння (1.7) і рівняння (2.3) дає систему з відкритим контуром, показано в рівнянні (2.4)

$$\frac{\text{Output}}{\text{Input}} = \frac{k_1 s^3 - k_2 s^2 + k_3 s}{k_4 s^5 + k_5 s^4 + k_6 s^3 + k_7 s^2 + k_8 s} \cdot K_p e(s) + \frac{K_i}{s} e(s) + K_d \frac{N}{1 + \frac{N}{s}} e(s) \quad (2.4)$$



Нарешті, рівняння системи розімкненої системи заводу можна легко перетворити на систему зворотного зв'язку з замкнутим контуром  $\left(\frac{G}{1+GH}\right)$

### 2.2.1 Розширений фільтр Калмана

Системи безпілотних літальних апаратів із нерухомим крилом завжди видають шумовий вихідний сигнал через шумовий зворотний зв'язок, тому для оцінки початкового вихідного сигналу та усунення ефекту шуму використовується метод ЕКФ (розширений фільтр Калмана). Рівняння (2.5) є стандартною моделлю простору станів нелінійної системи,

$$\left. \begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ \bar{y}_k &= h(x_k) + v_k \end{aligned} \right\} \quad (2.5)$$

де  $x_k$  — вектор стану;

$\bar{y}_k$  — вимірний вихідний сигнал процесу;

$w_k$  і  $v_k$  — шум процесу та вимірювання відповідно;

$f(\cdot)$  і  $g(\cdot)$  є загальними нелінійними функціями.

Стандартний фільтр Калмана, ЕКФ, також використовує двоетапний алгоритм передбачення та коригування. Оновлені рівняння ЕКФ показані в рівнянні (2.6).

$$\left. \begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, u_k) \\ P_{k+1} &= A_k P_k A_k^T + Q \end{aligned} \right\} \quad (2.6)$$

де  $\hat{x}_{k+1}$  — оцінка стану оновлення.

Рівняння оновлення часу проектують оцінку стану та коваріації  $P_{k+1}$  від попереднього кроку часу  $k$  до поточного кроку часу  $k+1$ . Рівняння оновлення вимірювань ЕКФ такі:

$$K_k = P_k C_k^T (C_k P_k C_k^T + R)^{-1}$$

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k) + K_k [\bar{y}_k - h(\hat{x}_k)]$$

$$P_{k+1} = A_k (1 - K_k C_k) P_k$$

де  $K$  — це вектор посилення Калмана;

$A_k$  і  $C_k$  не можна використовувати безпосередньо.

При цьому типі обмеження використовується ряд Тейлора або якобіан. Якобіанці визначаються як.

$$f'(x) = \begin{bmatrix} \frac{df_1}{dx_1} & \dots & \dots & \dots & \dots & \dots & \frac{df_1}{dx_n} \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ & & & & \cdot & & \\ \frac{df_n}{dx_1} & \dots & \dots & \dots & \dots & \dots & \frac{df_n}{dx_n} \end{bmatrix} \quad (2.7)$$

де  $f(\dots)$  можна обчислити як у рівнянні (11).

### 2.3 Валідація запропонованого алгоритму у MATLAB

Для валідації запропонованого PID з технікою керування EKF у стабілізації кута крену БПЛА було обрано форму пластини MATLAB. Параметри ПД-регулятора вибираються методом автоналаштування таким чином, щоб крен стабілізувався якомога швидше. Ці значення  $K_p = 0,4593$ ,  $K_i = 6,4651$ ,  $K_d = -0,0123$  з  $N = 37,3396$ . Параметри моделі стабілізації кута крену БПЛА Fix-wing вибрано, як зазначено в таблиці 2.1.

На рис. 2.3 показано ступінчасту реакцію розімкнутого циклу руху БПЛА на крен, яка показує, що його амплітуда абсолютно неконтрольована. На рис. 2.4

показана реакція замкнутого циклу на рух крену з ПД-регулятором за наявності шумів датчика, які додаються в його контур зворотного зв'язку. Ця цифра підтверджує ефективність ПД-регулятора, він швидко підтримував потрібне значення кута крену, але через наявність шуму в його контурі зворотного зв'язку вихід шумить і не дуже хороший. На рис. 2.5 показано шумовий сигнал, який додається до системи з дисперсією 0,5 м/с та амплітудою 0,1. На рис. 2.6 показана повна ефективність запропонованого PID з алгоритмом ЕКФ, який дає бажану вихідну характеристику з усуненням небажаних шумів для стабілізації кута крену БПЛА Fix-wing.

Таблиця 2.1 – Коефіцієнти системи

K1	0.0046
K2	0.0046
K3	0.0211
K4	1.5586e-4
K5	0.0015
K6	0.0022
K7	0.0076
K8	6.6266e-5

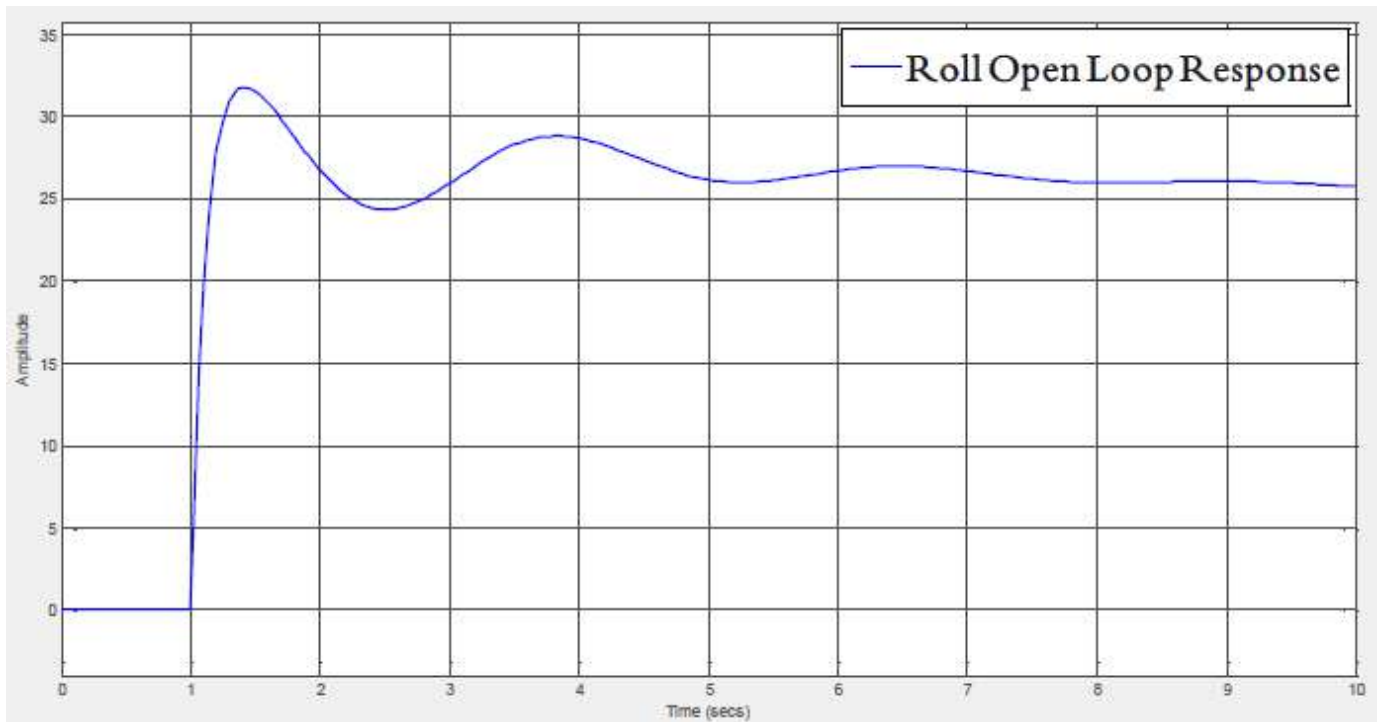


Рисунок 2.3 – Ступінчаста реакцію розімкнутого циклу руху БПЛА на крен

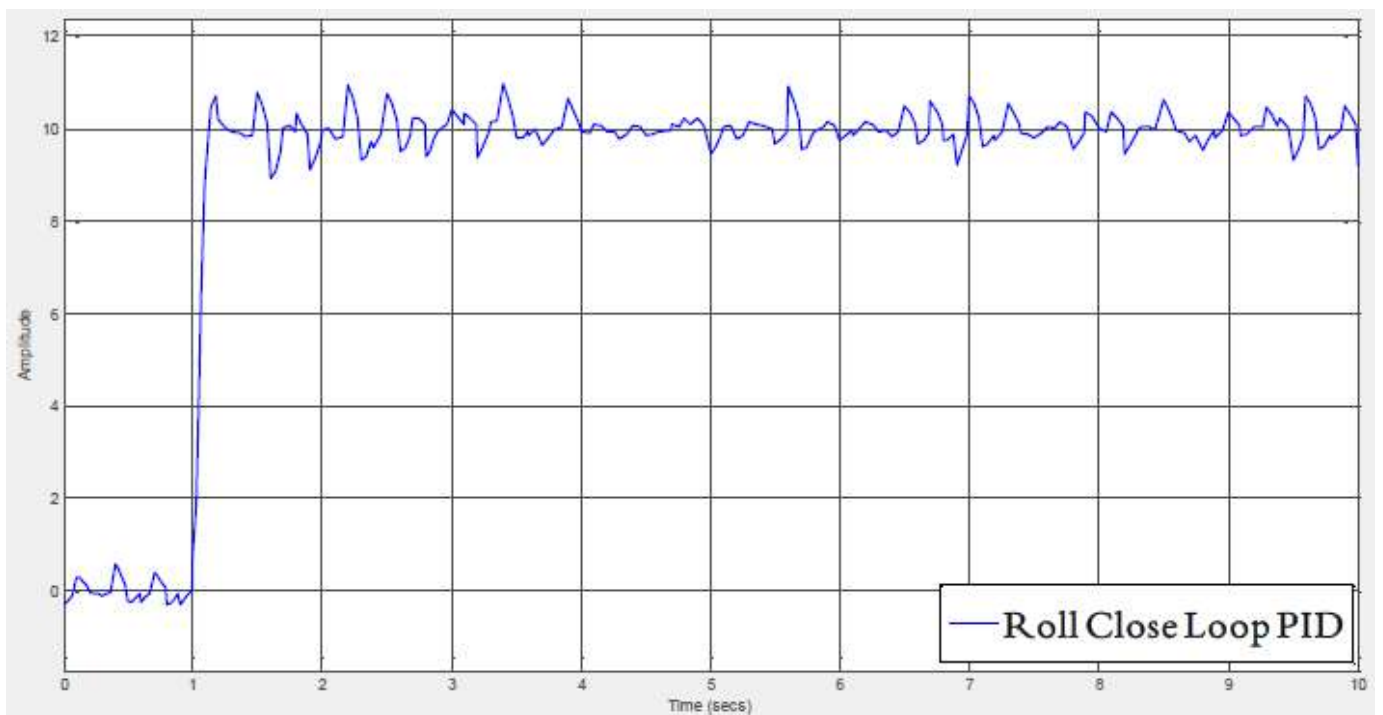


Рисунок 2.4 – Реакція замкнутого циклу на рух крену з ПІД-регулятором за наявності шумів датчика

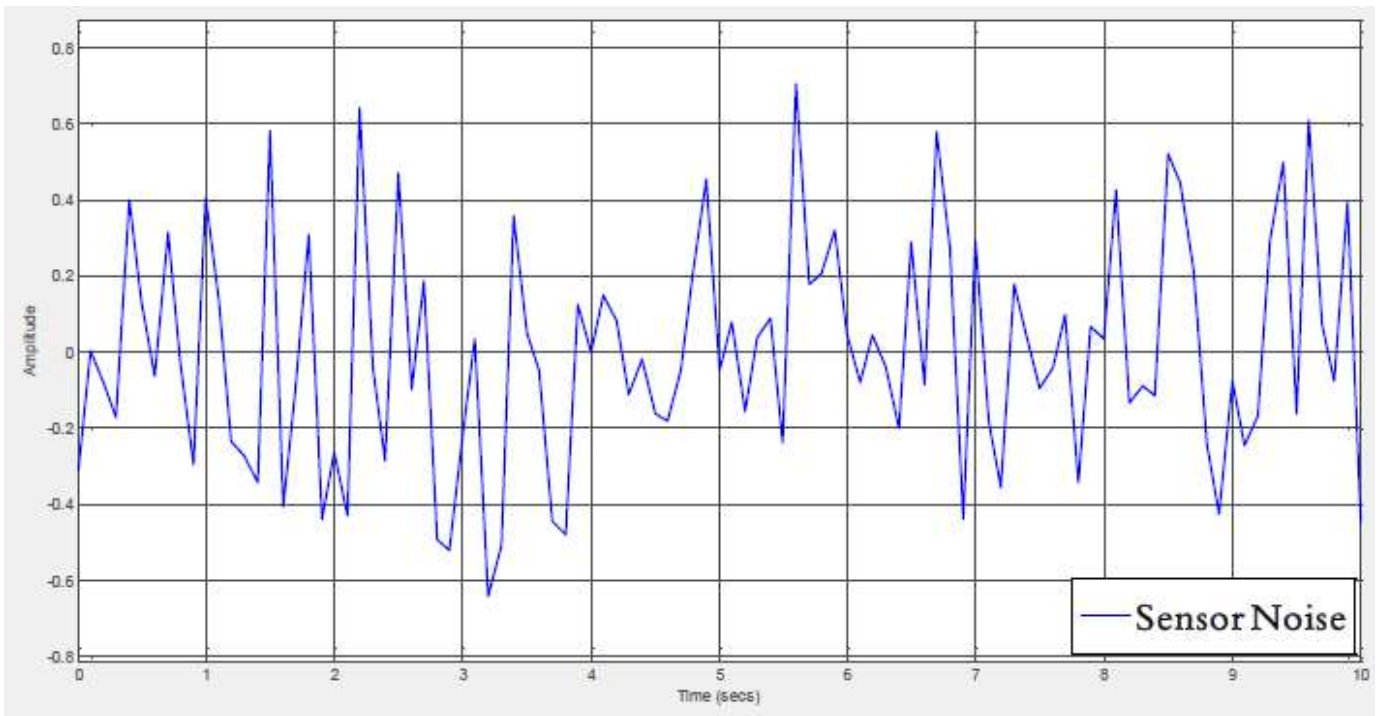


Рисунок 2.5 – Шумовий сигнал, який додається до системи з дисперсією 0,5 м/с та амплітудою 0,1

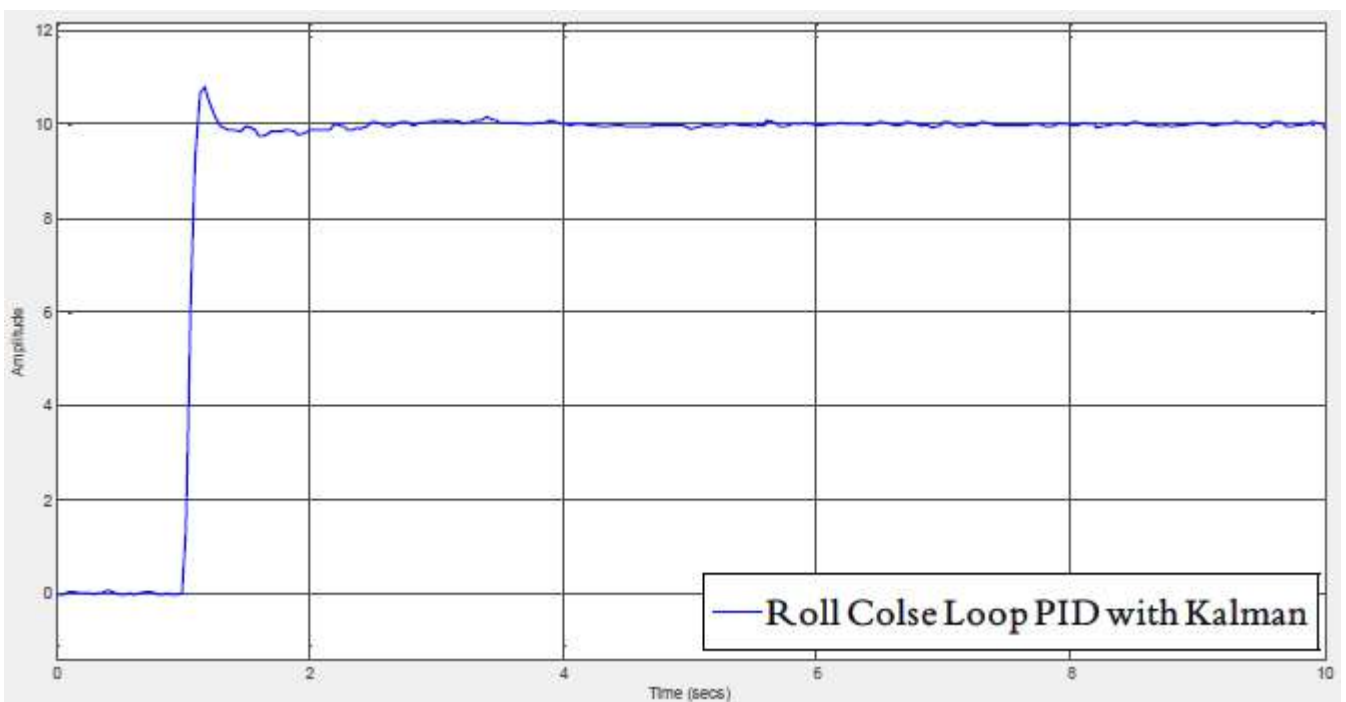


Рисунок 2.6 – Повна ефективність запропонованого PID з алгоритмом ЕКФ

Відповідно до результатів моделювання спостерігається, що ПІД-регулятор стабілізує рух крену, але за шумних умов його вихід порушується, але після

поєднання ПД-регулятора з ЕКФ він дає дуже хороші результати. З результатів моделювання ми можемо зробити висновок, що запропонований ПД з методологією керування ЕКФ є достатньо хорошим для забезпечення стабільності руху крену в умовах шуму та перешкод.

## 3 ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ СТАБІЛІЗАЦІЇ БЕЗПЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ ЛІТАКОВОГО ТИПУ ЗА КРЕНОМ

### 3.1 Отримання кута крену за допомогою MEMS датчиків

Для отримання даних щодо кутового положення плати необхідно здійснити підключення та налаштування мікросхем гіроскопа та акселерометра.

#### 3.1.1 Підключення гіроскопа L3GD20

Схема підключення гіроскопа L3GD20 до мікроконтролера STM32F3 реалізована на налагоджувальній платі Discovery F3 представлена на рис.2.1.

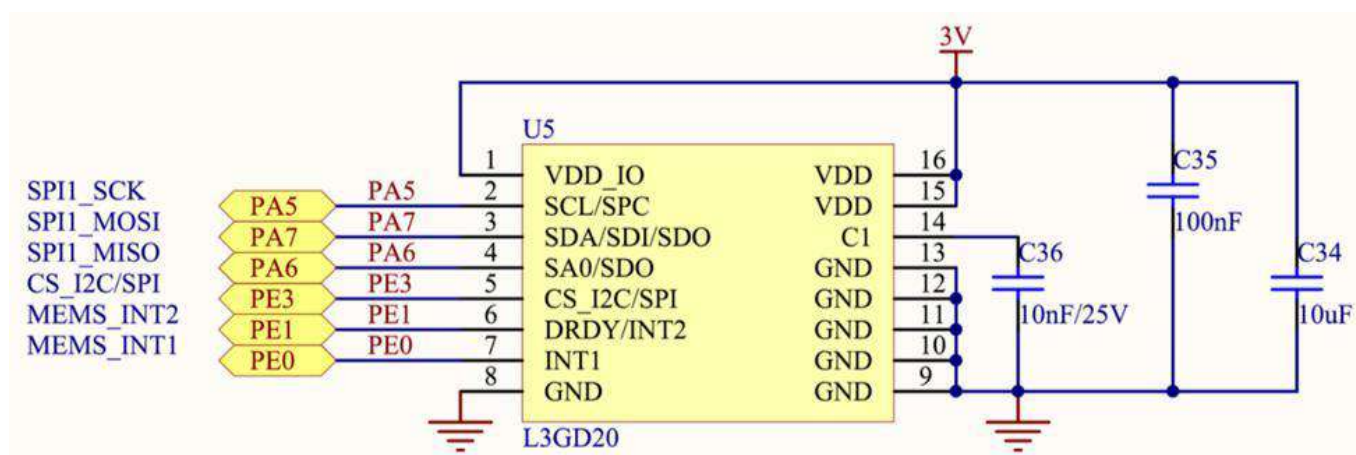


Рисунок 3.1 – Схема підключення гіроскопа до плати

Для читання даних біт  $R\bar{W}$  (рисунок 3.2), що передається по лінії SDI, встановлюється в логічну одиницю, а для запису в нуль. А біти AD5 – AD0 – служать передачі адреси регістру для запису чи читання. Установка біта  $M\bar{S}$  в логічну одиницю переводить режими зчитування або запису в множинний вигляд, тобто відбувається автоматична інкрементація адреси регістра.

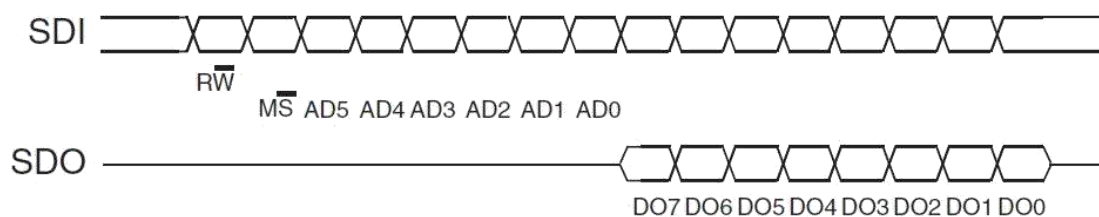


Рисунок 3.2 – Протокол читання/запису L3GD20

Адреси вбудованих у мікросхему L3GD20 регістрів представлені таблиці 3.1.

Таблиця 3.1 – Адреси регістрів гіроскопа L3GD20

Назва	Тип	Адреса регістру		За замовчуванням
		HEX	BIN	
WHO_AM_I	r	0F	0001111	11010100
OUT_TEMP	r	26	0100110	Вихід
STATUS_REG	r	27	0100111	Вихід
OUT_X_L	r	28	0101000	Вихід
OUT_X_H	r	29	0101001	Вихід
OUT_Y_L	r	2A	0101010	Вихід
OUT_Y_H	r	2B	0101011	Вихід
OUT_Z_L	r	2C	0101100	Вихід
OUT_Z_H	r	2D	0101101	Вихід
FIFO_CTRL_REG	rw	2E	0101110	00000000
FIFO_SRC_REG	r	2F	0101111	Вихід

Для того, щоб налаштувати одиниці, в яких приходять дані гіроскопа, необхідно налаштувати чутливість гіроскопа. За це відповідає параметр Sensitivity (табл. 3.2). На підставі даних таблиці можна зробити висновок про те, що для того, щоб конвертувати сирі дані з гіроскопа в градуси необхідно примножити вихідне значення в dps  $(8.75; 17.50; 70) * 10^{-3}$  в залежності від обраного значення чутливості приладу. DPS(Differential-phase-shift) – диференціальний фазовий зсув.



Таблиця 3.2 – Значення параметрів чутливості

Символ	Параметр	Тестове значення	Значення	Од.вим.
So	Чутливість	FS=250 dps	8,75	mdps/digit
		FS=500 dps	17,50	
		FS=2000 dps	70	

Біт CS встановлюється в нуль для початку роботи з мікросхемою. Сама ж прочитана інформація знаходиться в бітах D00 – D07. Використовуючи стандартну бібліотеку HAL від компанії ST, була написана власна бібліотека для роботи з мікросхемою.

Зрештою після запуску мікросхеми на чутливості = 500 вийшли показання, представлені на рис. 3.3. За даними на графіку можна дійти невтішного висновку про сильної зашумленості сигналу гіроскопа, що у спокої.

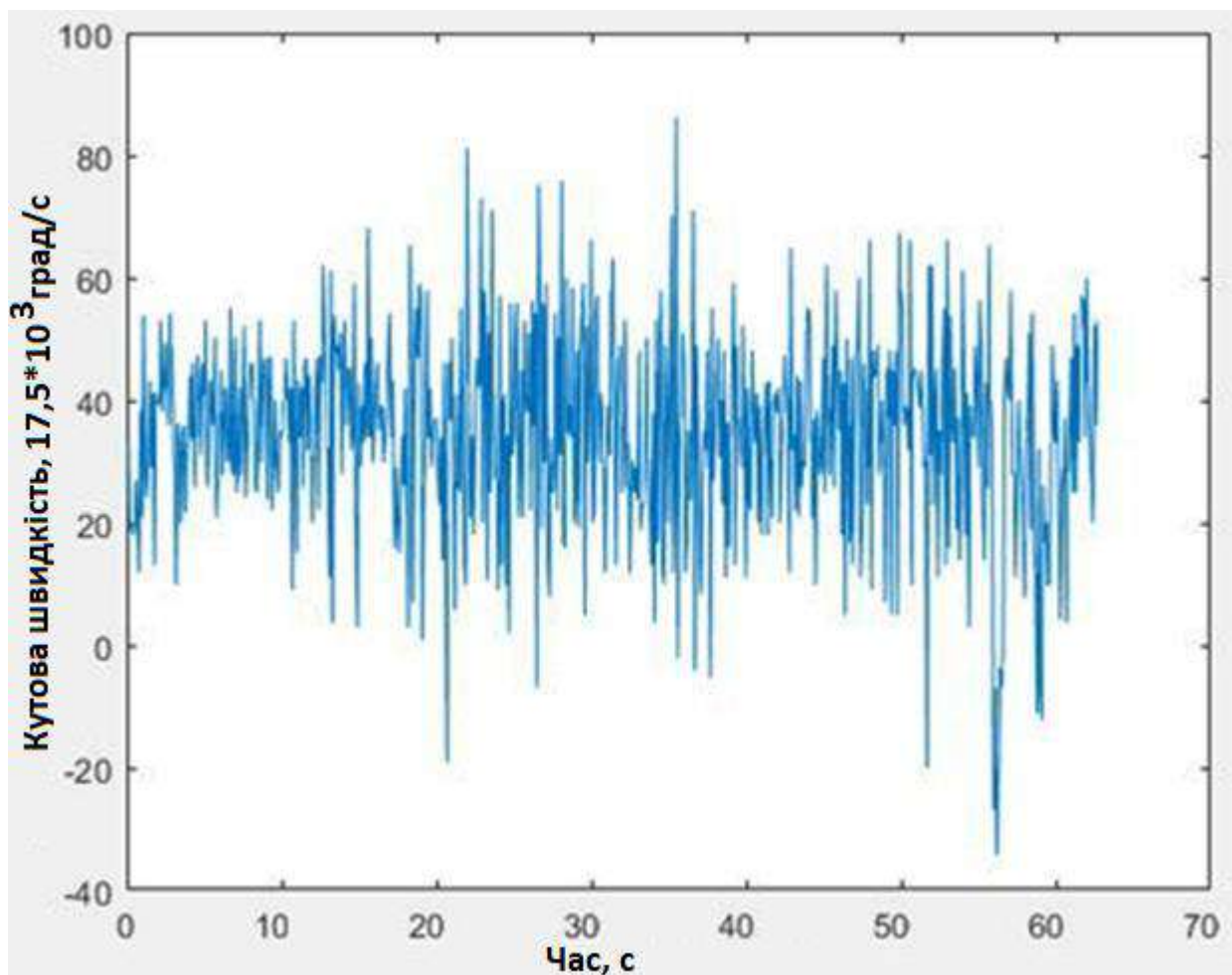


Рисунок 3.3 – Дані гіроскопа у спокої

Як відомо, гіроскоп вимірює значення кутових швидкостей, з чого випливає, що для отримання абсолютного значення кутового положення необхідно провести інтегрування за формулою (3.1).

$$\theta(T) = \theta_0 + \int_0^T \omega(t) * dt \quad (3.1)$$

де  $\theta(t)$  – значення кута в момент часу  $t = T$ ;

$\theta_0$  – значення кута в початковий момент часу;

$\omega(t)$  – функція кутової швидкості.

Формула (3.1) призначена для безперервної функції кутової швидкості. У роботі вимірювання даних гіроскопа відбувається у дискретні моменти часу, з частотою дискретизації 100 Гц, тобто. свідчення зчитуються з періодом 10 мс. Оскільки вимірювання вхідного сигналу здійснюється з малим кроком, то формулі (3.1) інтеграл можна замінити сумою за способом лівих прямокутників, заснованому на заміні підінтегрального виразу дискретною функцією. Формула, що вийшла, представлена нижче:

$$\theta(T) = \theta_0 + \sum_{t=0}^T \omega(t) * dt \quad (3.2)$$

У формулі (3.2) значення  $dt$  відповідає 10 мс. Використовуючи формулу (3.2) для розрахунку кута гіроскопа, отримаємо дані, що представлені на графіку на рис.3.4.

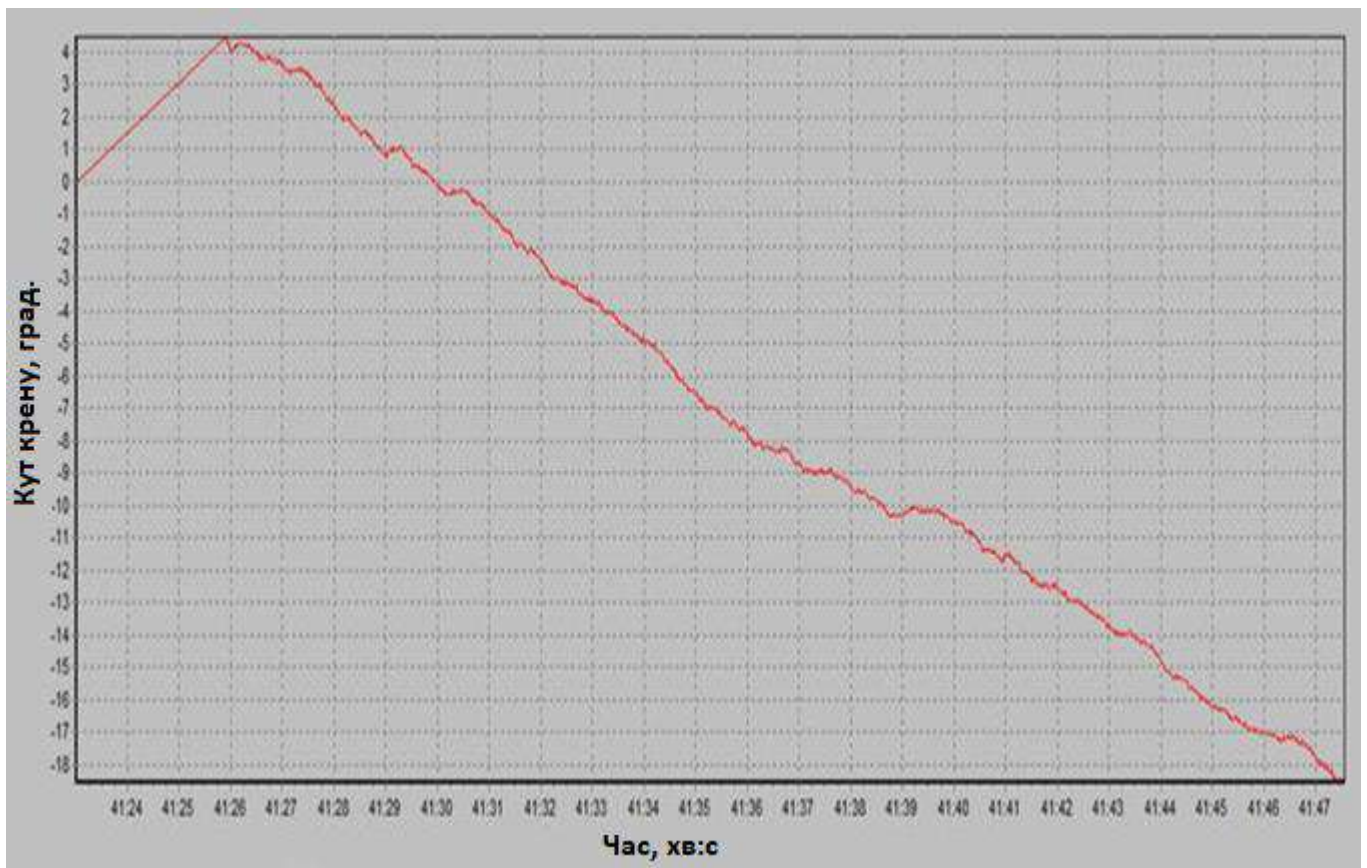


Рисунок 3.4 – Значення кута крену при інтегруванні даних гіроскопа

За графіком на рис.3.4 видно, що кут крену постійно зменшується, проте плата лежала у спокої. Це свідчить у тому, що з інтеграції відбувається накопичення помилки виміру і це помилка значна. Для того, щоб позбавитися постійної складової помилки інтегрування необхідно провести первинну обробку вихідної величини.

Для того щоб оцінити якість фільтрації був поставлений експеримент, в ході якого налагоджувальна плата STM32 F3 Discovery була встановлена на платформу, що обертається, яка імітує крен літака. Платформа є сервопривід з прикріпленням до нього будівельним кутом. Фотографія стенду представлена на рис. 3.5.

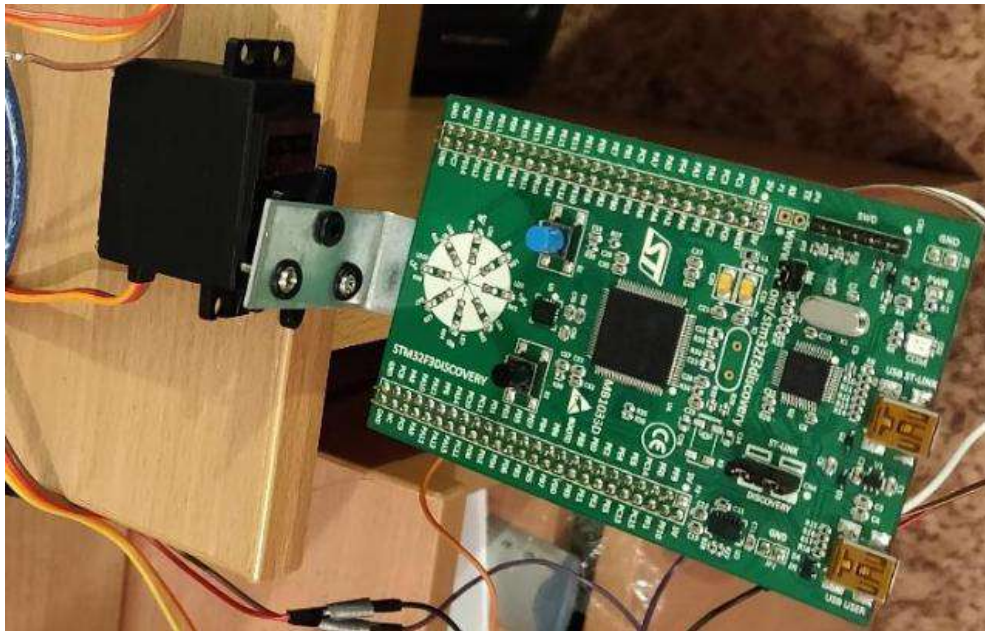


Рисунок 3.5 – Експериментальний стенд

Для керування сервоприводом було складено алгоритм на мікроконтролерній платі Arduino Nano, що виконує розворот серво на 90 градусів у кожну зі сторін з часовим проміжком 2 секунди. Ознайомившись із технічною документацією на даний сервопривід, було визначено його операційну швидкість, яка становить  $\omega = \frac{60}{0,17} = 352,941$  градус/с. Це означає, що для одного повороту на 90 градусів сервоприводу знадобиться  $t = \frac{\varphi}{\omega} = 0,255$  с. Для порівняння алгоритмів фільтрації між собою пропонується вимірювати відносний час перехідного процесу, що дорівнює відношенню отриманого часу перехідного процесу до теоретично правильного. Це допоможе оцінити часові затримки фільтра.

Так як вихідний сигнал гіроскопа є досить сильно зашумлений сигнал з переважанням коливань щодо справжнього значення, його необхідно відфільтрувати.

Результат фільтрації ковзним середнім, використовувана ширина вікна дорівнює 10 значенням, представлений на рис. 3.6. А в таблицю 3.3 зведені результати фільтрації під час використання різного значення параметра N.

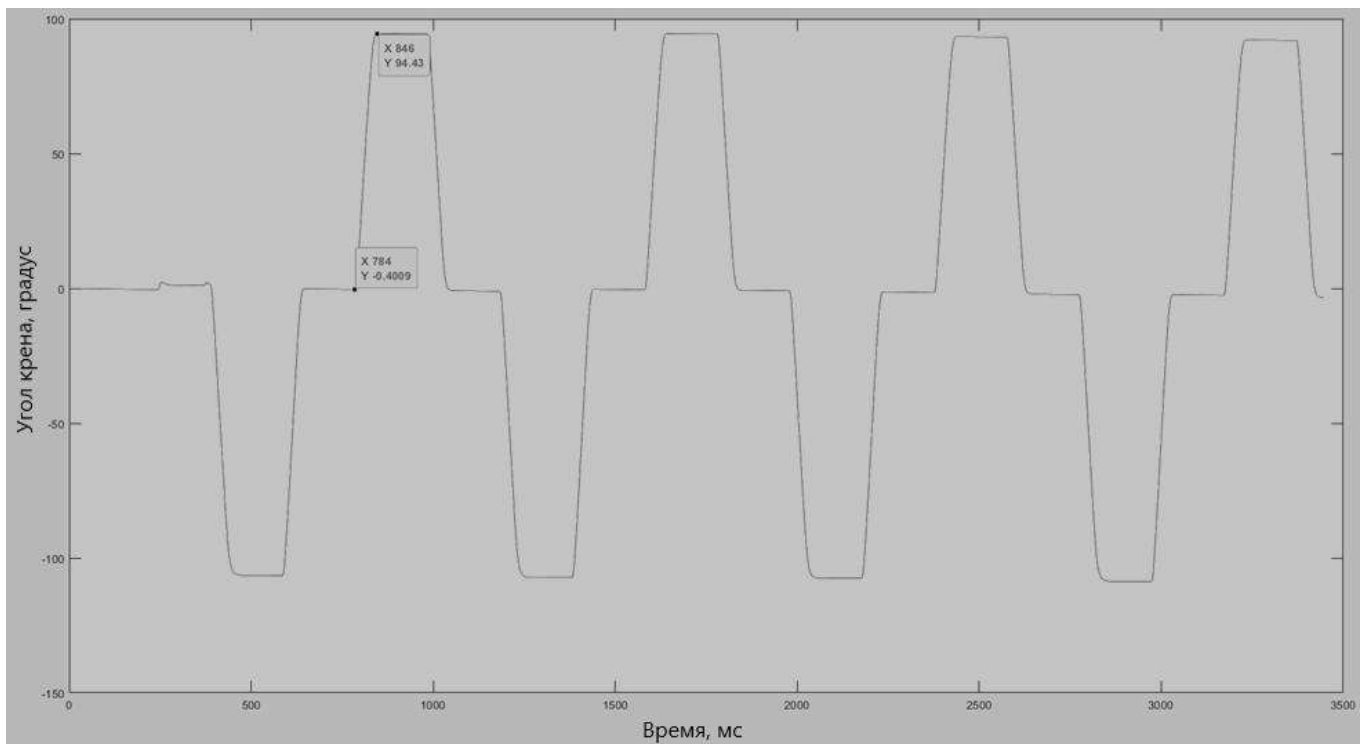


Рисунок 3.6 – Результат фільтрації фільтром ковзного середнього при ширині діапазону  $N = 10$

Таблиця 3.3 – Результати експерименту

N	$t_{\text{перехід}}, \text{с}$	$t_{\text{перехід}}/t_{\text{теор}}, \%$	Зміщення нуля за період
10	0,62	243	0,124
20	0,72	282	1,288
100	1,09	427	1,340

За представленими в таблиці 3.3 результатами можна дійти невтішного висновку у тому, що первинна обробка сигналу методом ковзного середнього дозволяє значно зменшити зміщення нуля гіроскопа. Однак при дуже великому значенні  $N$  відбувається затримка сигналу щодо не відфільтрованого.

Результат фільтрації медіанним фільтром представлений на рис.3.7, у своїй ширині вікна  $N = 10$ . На таблиці 3.4 зведені результати фільтрації з використанням різного значення параметра  $N$ .

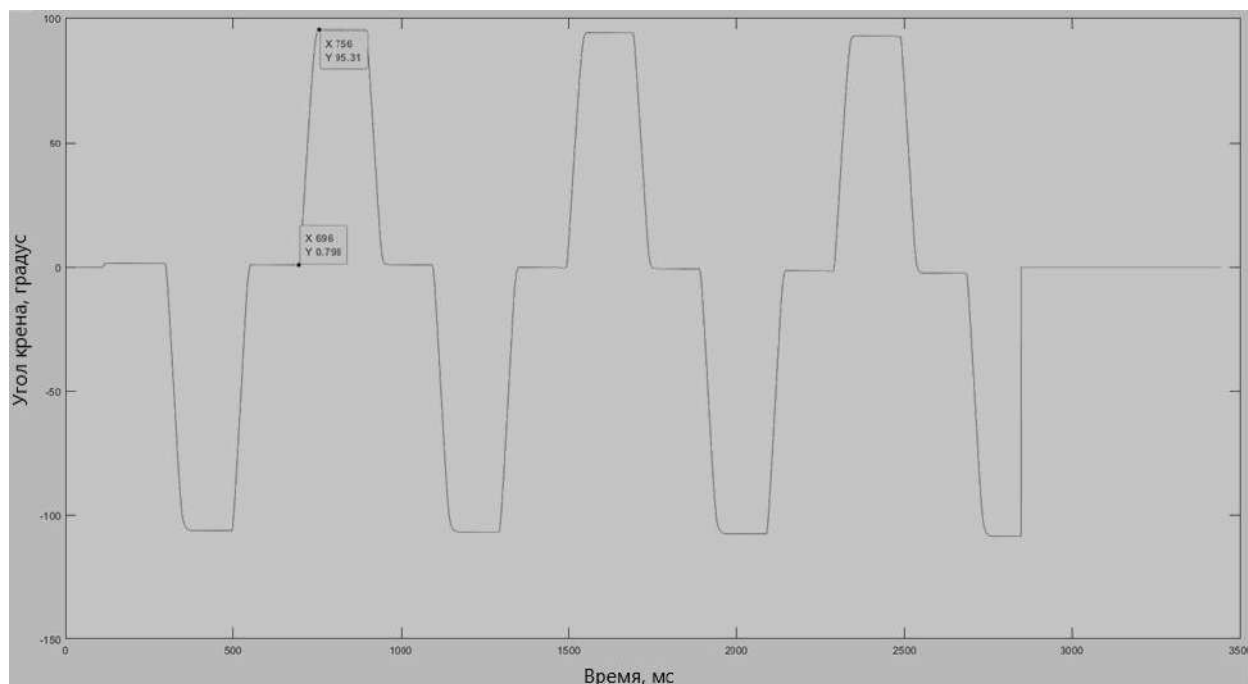


Рисунок 3.7 – Результат фільтрації медіанним фільтром при ширині діапазону  
N = 10

Таблиця 3.4 – Результати експерименту

N	$t_{\text{перехід}}, \text{с}$	$t_{\text{перехід}}/t_{\text{теор}}, \%$	Зміщення нуля за період
10	0,62	235	0.5462
20	0,75	294	0.2754
100	0.76	298	0.2749

За даними, представленими в таблиці 3.4 можна дійти невтішного висновку у тому, що результати не сильно відрізняються від результатів фільтрації ковзним середньому (таблиця 3.3). Цей фільтр дозволяє так само, як і фільтр ковзного середнього мінімізувати помилку інтегрування. Однак, медіанний фільтр більш складний для обчислення мікроконтролера. Тому саме в цьому випадку не доцільно використовувати його, оскільки вихідний сигнал гіроскопа не містить імпульсних перешкод. У результаті було прийнято рішення використовувати фільтрацію ковзним середнім з оптимально-мінімальною шириною вікна N = 10.

### 3.1.2 Підключення мікросхеми LSM303DLHC

Схема підключення акселерометру LSM303DLHC до мікроконтролера STM32F303VCT6 представлена на рис.3.8. Цей акселерометр підтримує підключення за інтерфейсом I2C

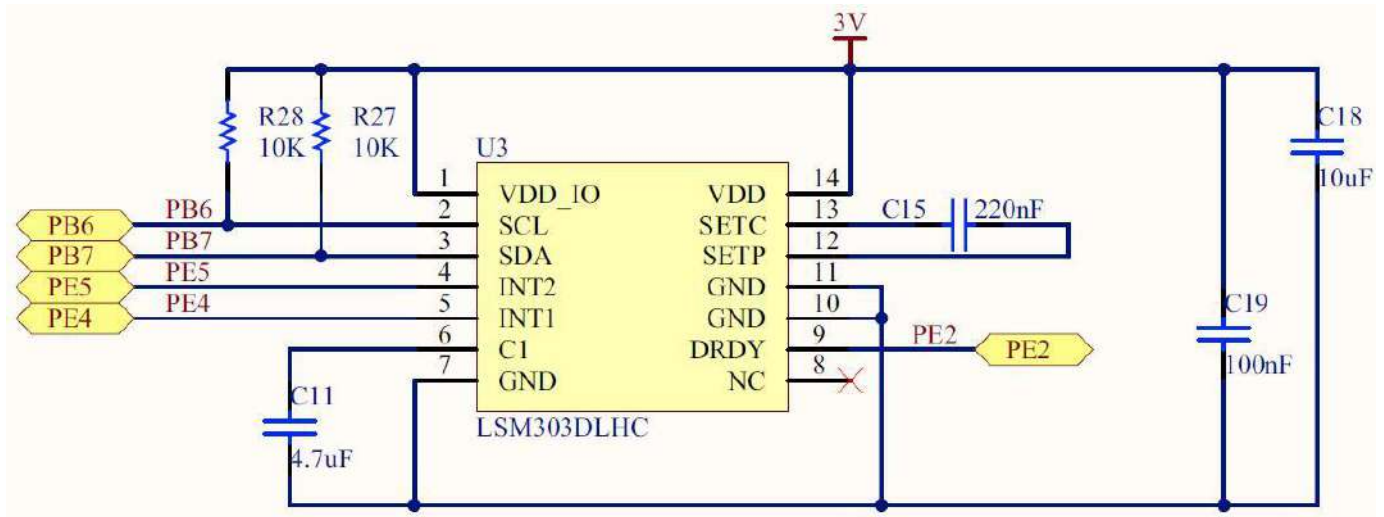


Рисунок 3.8 – Схема підключення датчика до мікроконтролера

Адреси регістрів мікросхеми LSM303DLHC представлені у табл.3.5.

Таблиця 3.5 – Регістри мікросхеми LSM303DLHC

Назва	Тип	Адреса регістру		Значення за замовчуванням
		Hex	Bin	
OUT_X_H_M	r	03	00000011	вихід
OUT_X_L_M	r	04	00000100	вихід
OUT_Z_H_M	r	05	00000101	вихід
OUT_Z_L_M	r	06	00000110	вихід
OUT_Y_H_M	r	07	00000111	вихід
OUT_Y_L_M	r	08	00001000	вихід



За даними табл.3.5 видно, що вихідні значення гіроскопа зберігаються в регістрах OUT\_X\_H\_M – OUT\_Y\_L\_M. З зв'язку з тим, що вихідне значення є 16 бітове значення, то на зберігання однієї змінної виділяється 2 байти інформації. Так, наприклад, проекція прискорення на вісь X мікросхеми зберігається в регістрах OUT\_X\_H\_M і OUT\_X\_L\_M

Так як акселерометр вимірює лише прискорення, то в статичні моменти часу він вимірюватиме прискорення вільного падіння, розкладеного на ортогональні осі мікросхеми. Напрямок осей мікросхеми представлено на рис. 3.9.

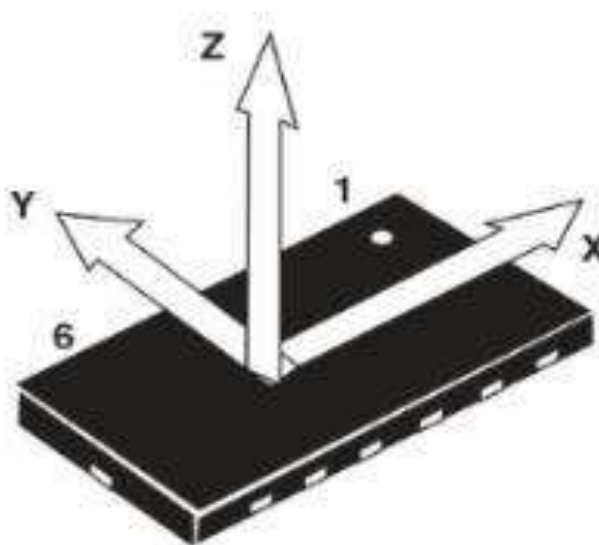


Рисунок 3.9 – Напрямок осей мікросхеми LSM303DLHC

Так як на літаку мікросхема розміщена віссю спрямованої протилежно до польоту БПЛА, то для отримання значення кута крену слід використовувати проекцію сили тяжіння на осі Z і Y. Отже, кут крену літака може бути визначений за формулою (3.3).

$$\theta = \arctg\left(\frac{g_y}{g_z}\right) \quad (3.3)$$

де  $\theta$  – значення кута крену;

$g_y$  – значення проекції прискорення вільного падіння на вісь Y мікросхеми;

$g_z$  – значення проекції прискорення вільного падіння на вісь Z мікросхеми.



Оскільки розрахунок кута крену здійснюється щодо проекцій сил на відповідні осі, стає обов'язковим враховувати одиниці виміру акселерометра.

Після запуску мікросхеми LSM303DLHC за формулою (3.3) було отримано значення кута нахилу налагоджувальної плати. Графік у часовій осі представлений на рис. 3.10.

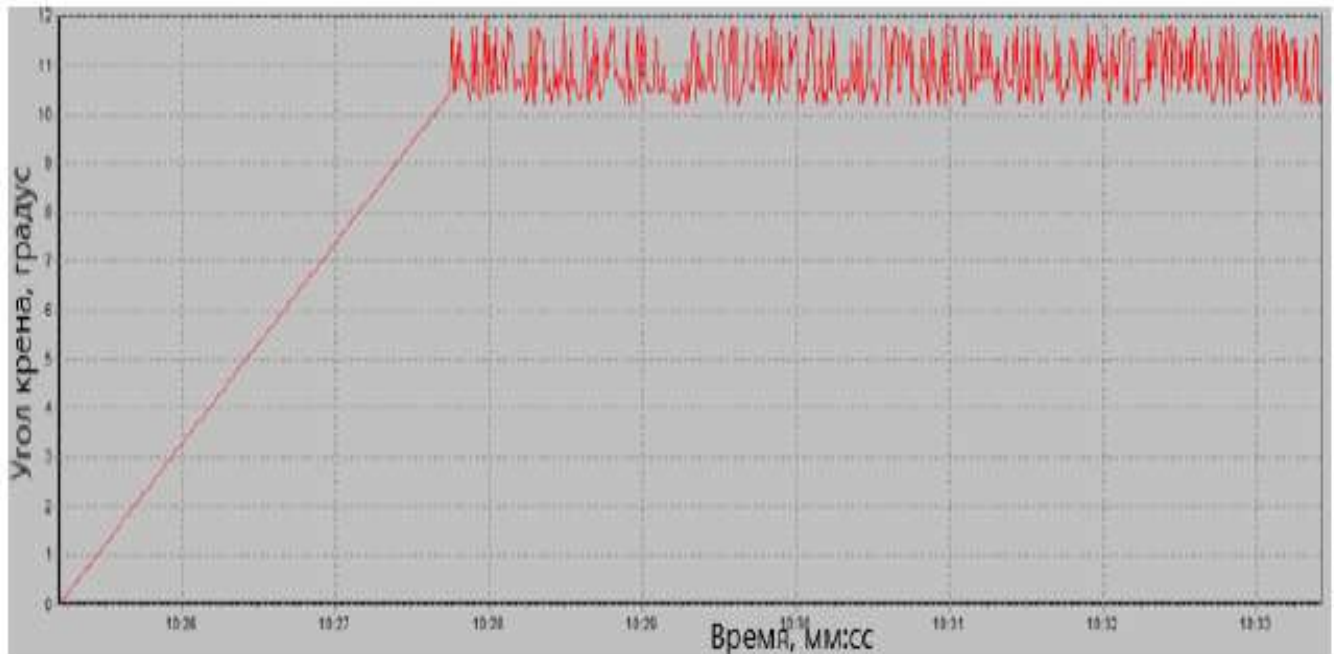


Рисунок 3.10 – Графік кута крену

За графіком, представленому на рис. 3.10 можна дійти невтішного висновку у тому, що вихідний сигнал акселерометра сильно зашумлений. Для отримання корисного сигналу, необхідно зробити первинну фільтрацію вихідного сигналу.

Результат фільтрації ковзним середнім з використанням вікна шириною 10 значень представлений на рис.3.11. А в таблицю 3.6 зведені результати фільтрації при використанні різного значення параметра N.

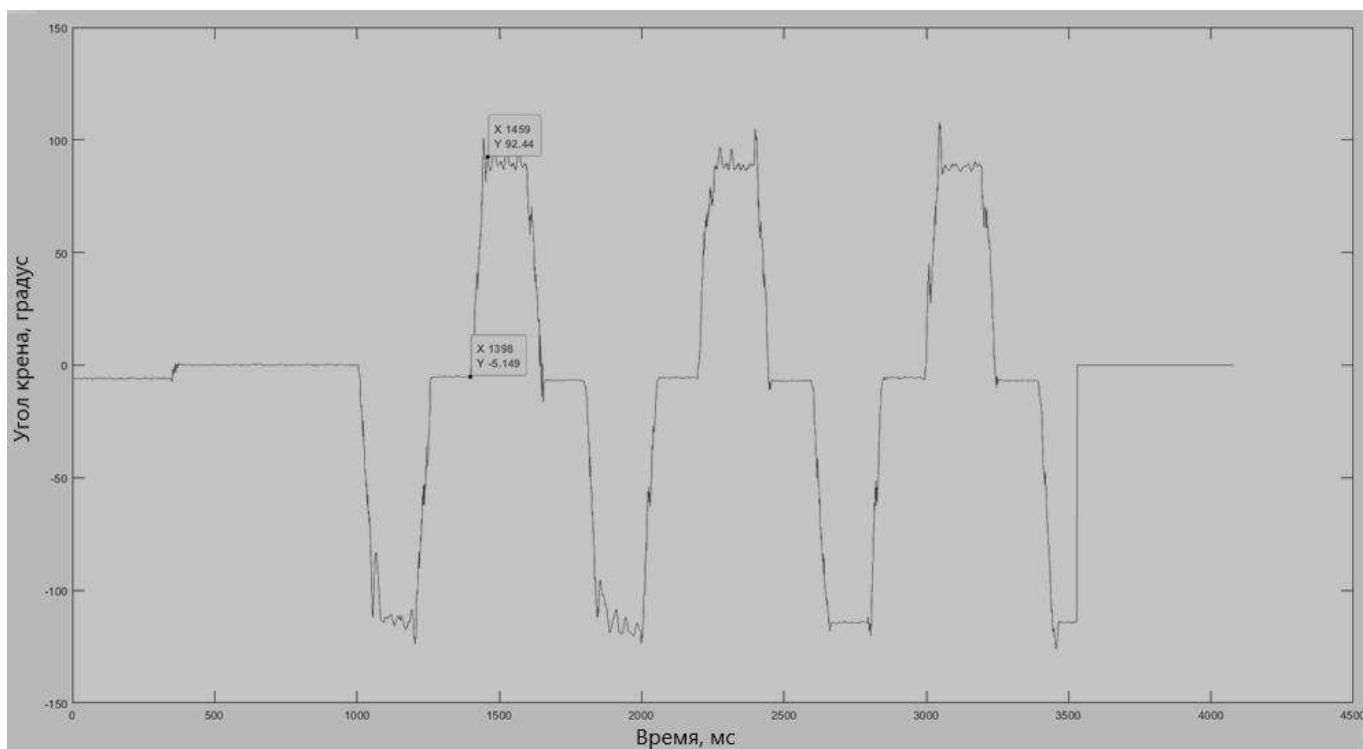


Рисунок 3.11 – Результат фільтрації фільтром ковзного середнього ширина вікна  $N = 10$

Таблиця 3.6 – Результати експерименту

N	$t_{\text{перехід}}, \text{с}$	$t_{\text{перехід}}/t_{\text{теор}}, \%$	Амплітуда коливань
10	0,61	232	5,50
20	0,93	364	2,63
100	1,49	584	0

На підставі даних таблиці 3.6 можна зробити висновок про те, що фільтр ковзного середнього повністю пригнічує імпульсні перешкоди при ширині вікна 100 і більше. Однак, при такому великому вікні відбувається значне спотворення сигналу, що є неприйнятним для швидкісної системи літака. В даному випадку буде доцільніше вдатися до фільтрації медіанним фільтром, так як вихідний сигнал акселерометра зашумлений імпульсними перешкодами.

Результат фільтрації медіанним фільтром з використанням вікна шириною 10 значень представлений на рис. 3.12. А в таблиці 3.7 зведені результати фільтрації при використанні різного значення параметра N.

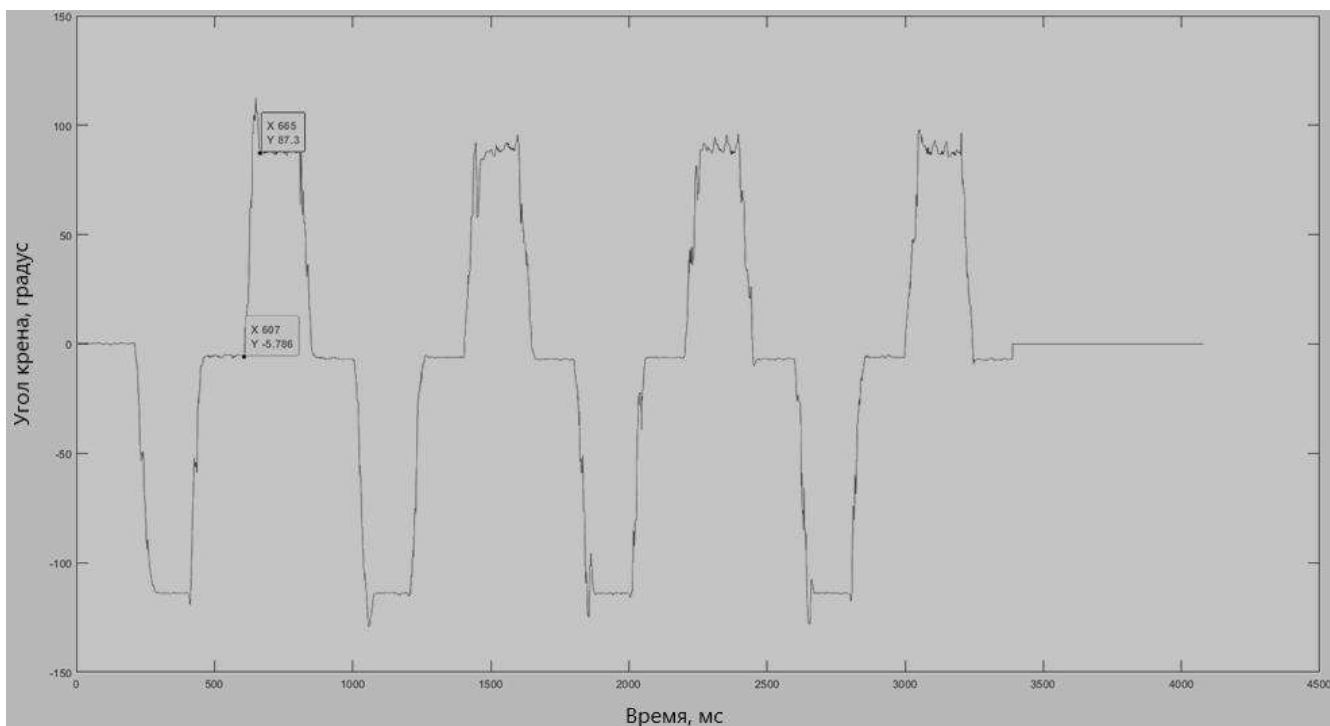


Рисунок 3.12 – Результат фільтрації медіанним фільтром ширина вікна  $N = 10$

Таблиця 3.7 – Результати експерименту

$N$	$t_{\text{перехід}}, \text{с}$	$t_{\text{перехід}}/t_{\text{теор}}, \%$	Амплітуда коливань
10	0,58	227	1,52
20	0,82	321	1,085
100	0,94	368	0

За даними, представленими в таблиці 3.7 можна дійти невтішного висновку у тому, що це фільтр краще придушує імпульсні перешкоди. Це впливає з того, що амплітуда коливань менша ніж у фільтра ковзного середнього при однакових параметрах  $N$ . При цьому даний фільтр вносить меншу затримку в систему. Однак, у медіанного фільтра при повному виключенні перешкоди ( $N$  значеннях більше 100) є значна затримка, це може стати причиною аварії літака через велике перерегулювання майбутньої системи. Отже, слід використовувати мінімальне значення параметра  $N = 10$ .

Зрештою було з'ясовано, що використовувати лише акселерометр або гіроскоп для вимірювання крену літака не можна. Тому було ухвалено рішення об'єднати значення цих двох датчиків.

### 3.1.3 Об'єднання показань двох датчиків

Для поєднання сигналів двох датчиків було вирішено використати комплементарний фільтр. На вхід комплементарного фільтра надходять попередньо відфільтрований сигнал гіроскопа методом ковзного середнього при  $N = 10$  і акселерометра, обробленого медіанним фільтром  $N = 10$ . На малюнку 3.13 представлений графік зміни величини кута крену при обробці комплементарним фільтром для значення параметра  $\alpha$ . У таблицю 3.8 зведено результати фільтрації для різних параметрів  $\alpha$

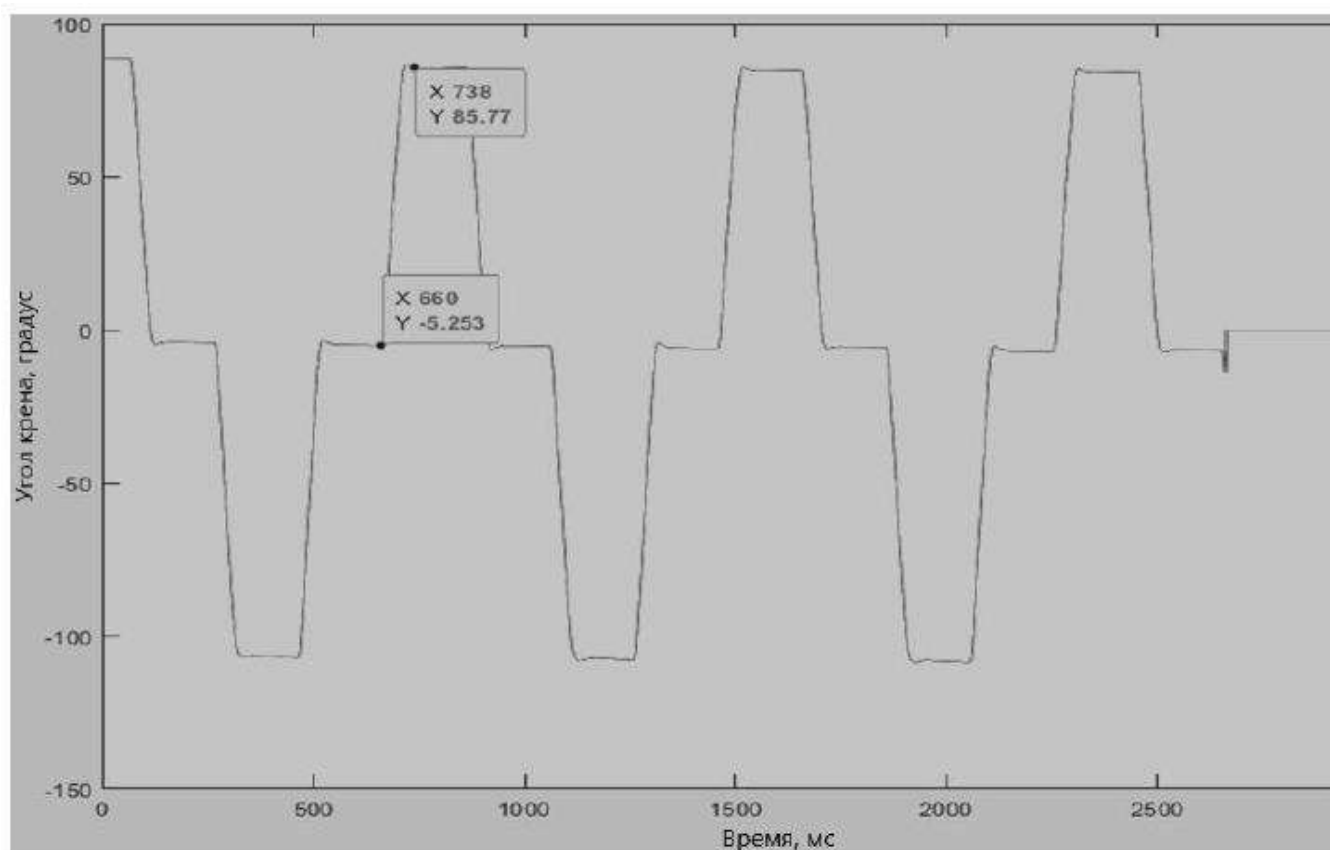


Рисунок 3.13 – Результат фільтрації комплементарним фільтром при параметр  $\alpha=0.1$

Таблиця 3.8 – Результати експерименту

alpha	$t_{\text{перехід}}, \text{с}$	$t_{\text{перехід}}/t_{\text{теор}}, \%$	Зміщення нуля	Амплітуда коливань
0.05	-	-	0	0
0.1	0,78	306	0	0
0.2	1,33	521	0	0,52
0.9	0,8	313	0	1,085

За даними, представленими в таблиці 3.8 можна зробити висновки про те, що при параметрі фільтра  $\alpha = 0.1$  відфільтрована величина має найменшу затримку і при цьому у неї відсутнє зміщення нуля з часом, властиве гіроскопу, а також відсутні імпульсні перешкоди від акселерометра. Використовуючи даний фільтр стає можливим отримати краще від обох сенсорів: точність показань у статичних режимах та відсутність зміщення нуля від акселерометра та точність показань у динаміці від гіроскопа

### 3.2 Визначення передавальної функції між кутом відхилення елеронів та кутом крену

Так як у даній роботі необхідно розробити систему, що стабілізує кут крену літака, необхідно визначити передатну функцію між кутом відхилення елеронів та кутовою швидкістю нахилу літака. Оскільки об'єкт управління є складною системою, яка виготовлена вручну, тобто. з низькою якістю дотримання початкових креслень деталей, а технічні характеристики сервоприводів не збігаються з реальними, математичний розрахунок передавальної функції стає практично неможливим. Тому був використаний принцип проектування чорний ящик, при якому не відома внутрішня будова системи, а відома лише вхідна дія та вихідний сигнал. Тоді для визначення передавальної функції необхідно скористатися математичним пакетом Matlab з встановленим розширенням System Identification

Toolbox. Для визначення передавальної функції у цьому ПЗ необхідно мати значення вхідних та вихідних даних.

Для визначення передавальної функції між кутом відхилення елеронів і кутової швидкості крену, необхідно в розширення System Identification Toolbox подати графік залежності в тимчасовій області кутової швидкості крену від кута відхилення елеронів. Для того, щоб отримати ці дані необхідно провести реальний політ літаком і при цьому виконувати запис кутової швидкості крену і кута відхилення елеронів в лог файли. Для вирішення цього завдання був застосований польотний контролер Pixhawk PX4. Для виключення всіх небажаних факторів політ повинен проходити в умовах, за яких зовнішні впливи на літак мінімальні. Тому експеримент було проведено в день з мінімальним значенням швидкості вітру, яка становила не більше ніж 2 м/с.

В результаті експерименту отримані дані кутової швидкості крену в одиницях рад/с, які представлені на рис.3.14. На жаль, стандартний автопілот записує замість кута відхилення елеронів, тільки ступінь їх відхилення в діапазоні від мінус 4500 до 4500 умовних одиниць. Графік ступеня відхилення елеронів представлений малюнку 26.

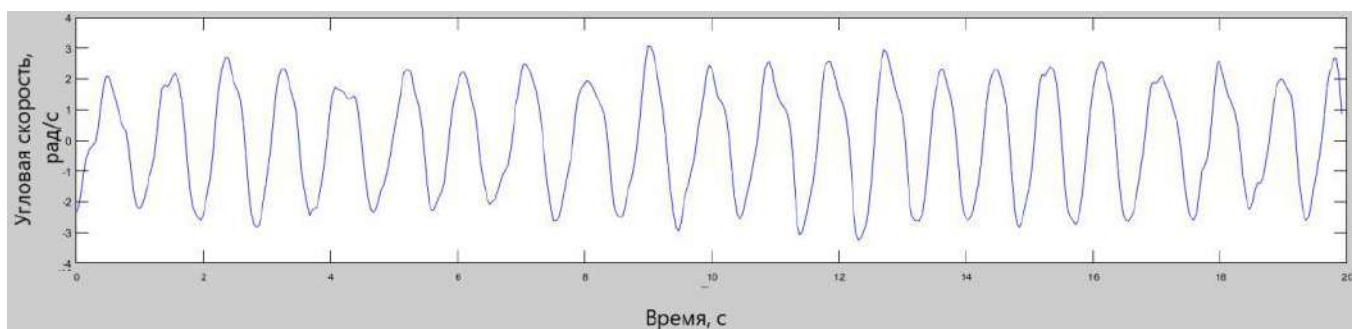


Рисунок 3.14 – Кутова швидкість крену

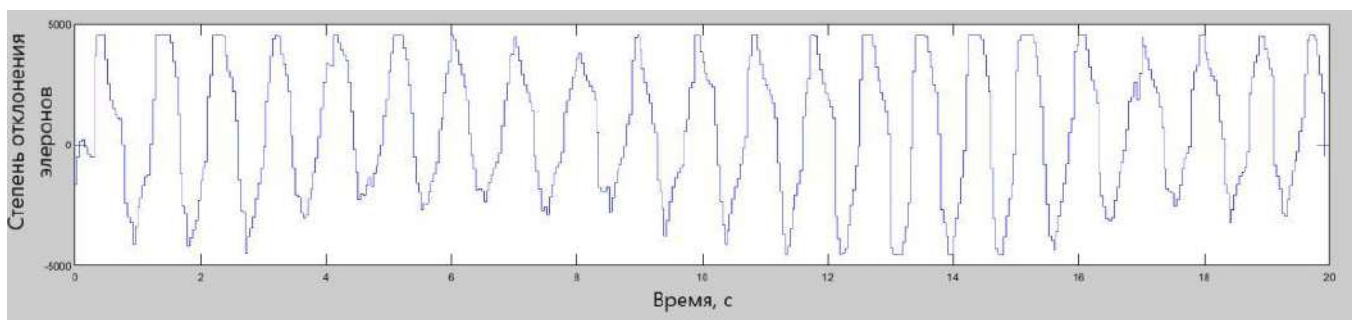


Рисунок 3.15 – Ступінь відхилення елеронів

Після того, як були визначені необхідні дані для визначення передавальної функції, їх необхідно завантажити в блок System Identification Toolbox. Завантаження даних у блок представлено на рис. 3.16. Після того як дані були завантажені можна приступати до процедури ідентифікації передавальної функції. Для цього необхідно визначити вид підсумкової бажаної передавальної. Як стандартна передавальна функція між кутом відхилення елеронів і кутовою швидкістю крену виступатиме аперіодична ланка першого порядку [1].

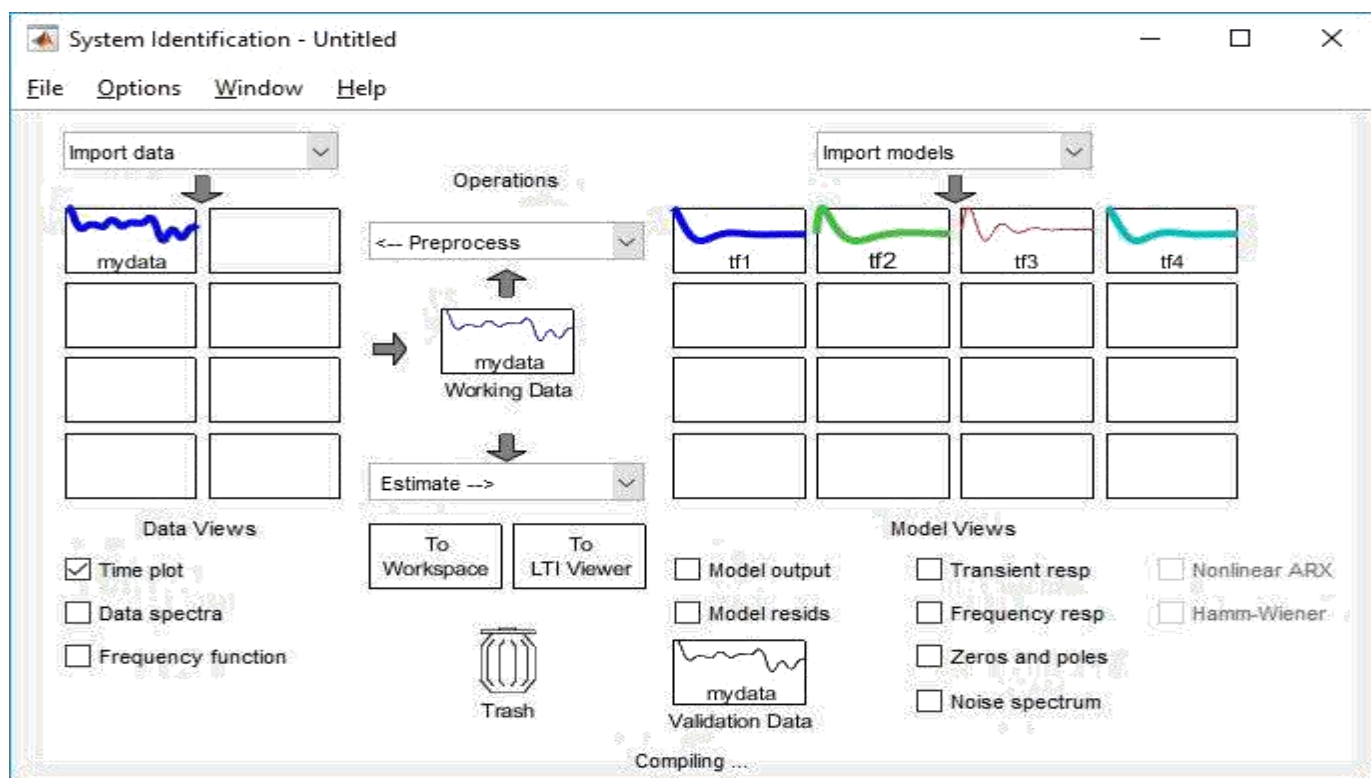


Рисунок 3.16 – Завантаження даних у блок System Identification Toolbox

Після всіх виконаних дій була отримана передатна функція системи. Ступінь відповідності вхідного сигналу становить 67,12%. Графік отриманої функції представлений на рис. 3.17.

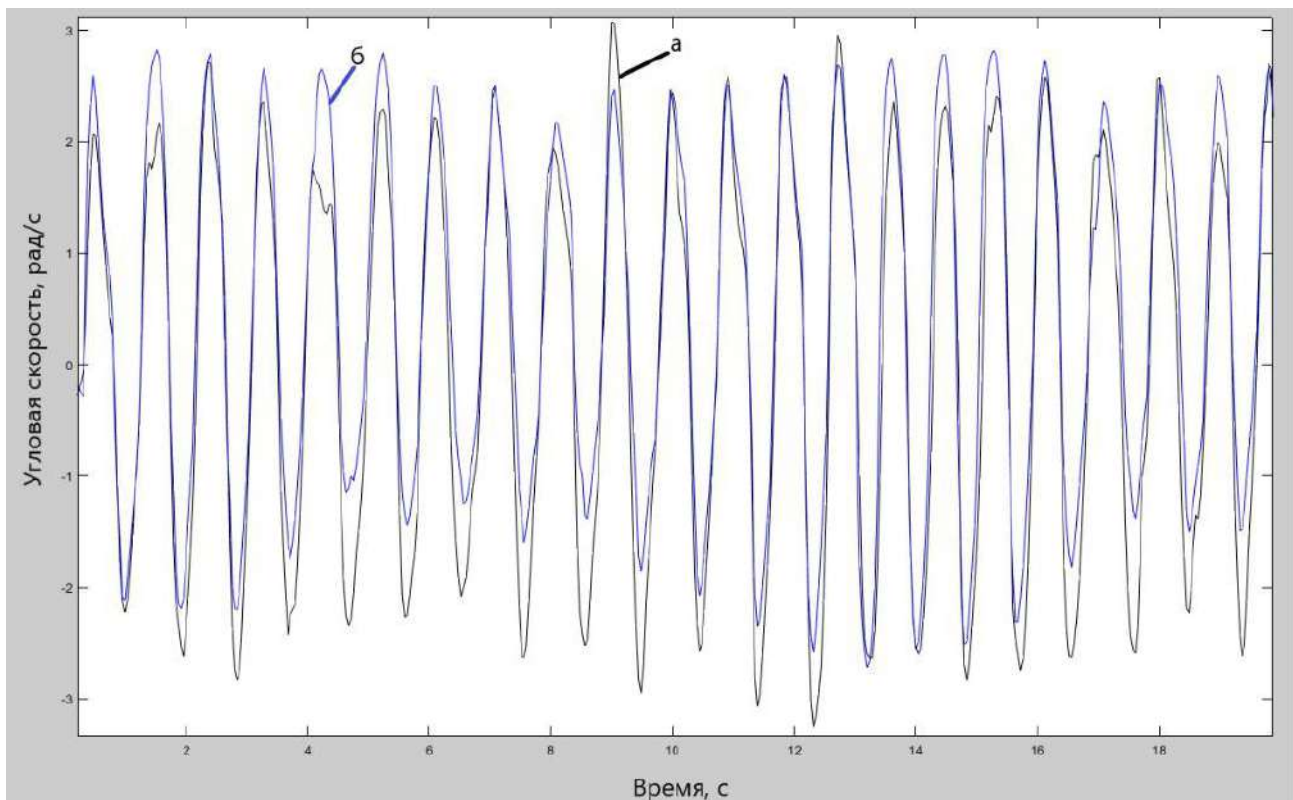


Рисунок 3.17 – Графіки зміни кутової швидкості крену у часовій області  
б – змодельована система; а – вихідна система

Підсумкова передатна функція виражена у формулі (3.4).

$$W(s) = \frac{0.01012}{s+15.8} \quad (3.4)$$

Так як була визначена передатна функція між кутовою швидкістю крену і ступенем відхилення елеронів, а управління буде здійснюватися безпосередньо кутом крену літака, то виникає необхідність отримання передавальної функції між кутом крену і ступенем відхилення елеронів. Як вже було сказано для переходу від кутової швидкості крену до кута крену, необхідно виконати інтегрування кута крену. Отже, остаточна передатна функція буде виглядати наступним чином:

$$W(s) = \frac{0.01012}{s*(s+15.8)} \quad (3.5)$$



### 3.3 Налаштування параметрів регулятора в Simulink Matlab

Так як була отримана математична модель системи, можна приступати до налаштування ПІД-регулятора, скориставшись функцією автоматичного налаштування регулятора в графічному середовищі імітаційного моделювання Simulink. І тому необхідно побудувати модель системи у цьому середовищі. Готова до роботи модель системи представлена на рис. 3.18.

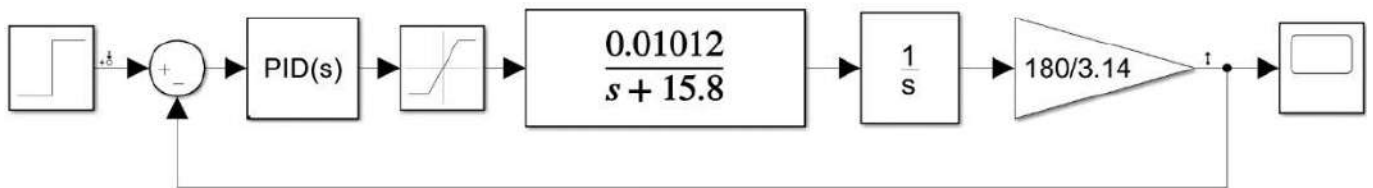


Рисунок 3.18 – Операторно-структурна схема Simulink

Процедура налаштування регулятора представлена малюнку 30. У цьому показники якості перехідного процесу вийшли такими:

- час перехідного процесу = 0,749 с;
- перерегулювання  $\sigma = 11.5\%$

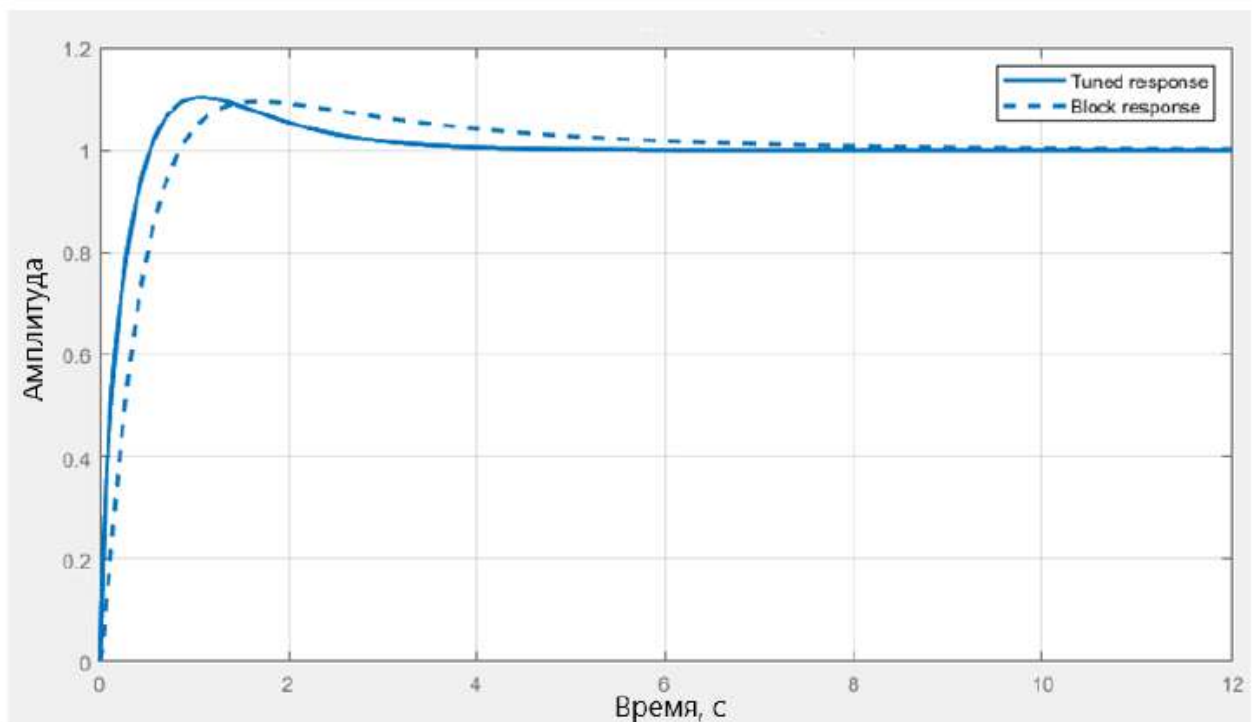


Рисунок 3.19 – Налаштування регулятора

Налаштовані коефіцієнти регулятора представлені на рис. 3.20.

Controller Parameters		
	Tuned	Block
P	132.6312	73.2801
I	108.4795	26.6167
D	12.9667	1.8968
N	558.8634	228.5656
Performance and Robustness		
	Tuned	Block
Rise time	0.37 seconds	0.575 seconds
Settling time	2.86 seconds	5.66 seconds
Overshoot	10.2 %	9.39 %
Peak	1.1	1.09
Gain margin	-Inf dB @ 0 rad/s	-Inf dB @ 0 rad/s
Phase margin	90 deg @ 4.89 rad/s	76.6 deg @ 2.66 rad/s
Closed-loop stability	Stable	Stable

Рисунок 3.20 – Коефіцієнти регулятора

### 3.4 Реалізація алгоритму дискретного ПД-регулятора мовою програмування Сі

Структурна схема ПД-регулятора, що використовується в математичному паку Simulink, представлена на малюнку 3.21.

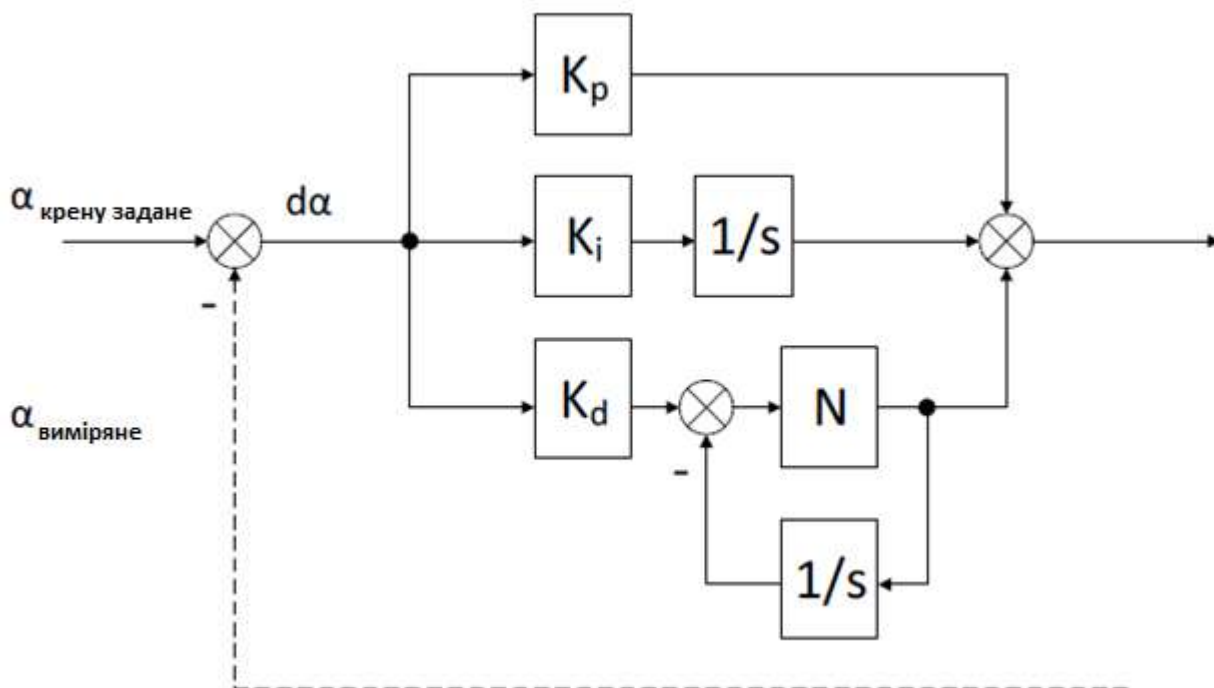


Рисунок 3.21 – Структурна схема ПІД-регулятора

Як відомо з курсу лекцій теорії автоматичного управління, паралельна форма запису рівняння ПІД-регулятора, структурна схема якого представлена на малюнку 3.21, виглядає наступним чином:

$$W(s) = K_p + \frac{K_i}{s} + \frac{K_d N}{1 + \frac{N}{s}} \quad (3.6)$$

де  $K_p$  – пропорційний коефіцієнт;

$K_i$  – коефіцієнт при інтегральній складовій;

$K_d$  – коефіцієнт при диференціальній складовій;

$N$  – коефіцієнт фільтрації диференціальної складової.

Для того щоб перетворити регулятор (3.6) на його дискретний аналог необхідно конвертувати інтегральну та диференціальну частини в їх дискретні аналоги. Це можна зробити трьома різними способами: прямий ейлер, зворотний ейлер і методом трапецій. У цій роботі використовуватиметься перетворення інтегратора шляхом зворотного Ейлера (Backward Euler). Формули перетворення представлені нижче:

$$Z\left(\frac{K_i}{s}\right) = \frac{K_i T_s z}{z-1} \quad (3.7)$$

$$Z\left(\frac{K_d N}{1+\frac{N}{s}}\right) = \frac{N(z-1)}{(1+NT_s)z-1} \quad (3.8)$$

Використовуючи рівність (3.7) і (3.8) отримаємо результуюче рівняння для дискретного ПД-регулятора:

$$C(z) = K_p + \frac{K_i T_s z}{z-1} + \frac{K_d N(z-1)}{(z+NT_s)z-1} \quad (3.9)$$

Так як передатна функція системи є ставленням Лапласового зображення вихідного сигналу до вхідного за нульових початкових умов, маємо наступне вираз:

$$U(z) = \left[ K_p + \frac{K_i T_s}{1-z^{-1}} + \frac{K_d N(1-z^{-1})}{(1+NT_s)-z^{-1}} \right] E(z) \quad (3.10)$$

Приведення до спільного знаменника виразу (3.10) дасть такий вираз

$$U(z) = \left[ \frac{K_p((1+NT_s)-2z^{-1}-NT_s z^{-1}+z^2) + K_i T_s((1+NT_s)-z^{-1}) + K_d N(1-z^{-1})^2}{(1+NT_s)-2z^{-1}-NT_s z^{-1}+z^{-2}} \right] E \quad (3.11)$$

Вираз (3.11) може бути переписаний як:

$$\begin{aligned} U(z) & \left( (1+NT_s) - 2z^{-1} - NT_s z^{-1} + z^{-2} \right) \\ & = (K_p((1+NT_s) - 2z^{-1} - NT_s z^{-1} \\ & + z^2) + K_i T_s((1+NT_s) - z^{-1}) + K_d N(1-z^{-1})^2) E(z) \end{aligned} \quad (3.12)$$

Приведення подібних до рівняння (3.12) дасть такий вираз:

$$\begin{aligned}
U(z) & \left( (1 + NT_s) - (2 + NT_s)z^{-1} + z^{-2} \right) \\
& = (K_p + K_p NT_s + K_i T_s + K_i T_s^2 N + K_d N \\
& + z^{-1} (-2K_p - K_p NT_s - K_i T_s - 2K_d N) \\
& + (K_p + K_d N) z^{-2}) E(z)
\end{aligned} \tag{3.13}$$

$Z[f(k - \alpha)] = F(z)z^{-\alpha}$  по теоремі зміщенні  $F(z) = Z[f(k)]$  слід, що рівняння для ПД-регулятора виглядатиме так:

$$\begin{aligned}
u[k](1 + NT_s) - u[k - 1](2 + NT_s) + u[k - 2] \\
= e[k](K_p + K_p NT_s + K_i T_s + K_i T_s^2 N + K_d N) - e[k - 1](2K_p + K_p NT_s - K_i T_s + 2K_d N) + e[k - 2](K_p + K_d N)
\end{aligned} \tag{3.14}$$

У рівнянні (3.14)  $u[k], u[k - 1], u[k - 2]$  – вихідний сигнал регулятора на  $k$ -му,  $k - 1$  і  $k - 2$  кроках відповідно;  $e[k], e[k - 1]$  та  $e[k - 2]$  – вхідний сигнал неузгодженості на  $k, k - 1$  і  $k - 2$  кроках.

Рівняння (3.14) можна переписати так:

$$\begin{aligned}
u[k] = \frac{2 + NT_s}{1 + NT_s} u[k - 1] - \frac{1}{1 + NT_s} u[k - 2] \\
+ \frac{K_p + K_p NT_s + K_i T_s + K_i T_s^2 N + K_d N}{1 + NT_s} e[k] \\
- \frac{2K_p + K_p NT_s - K_i T_s + 2K_d N}{1 + NT_s} e[k - 1] + \frac{K_p + K_d N}{1 + NT_s} e[k - 2]
\end{aligned} \tag{3.15}$$

Виконавши заміну  $\frac{2 + NT_s}{1 + NT_s}$  на  $k_{u1}$ ,  $\frac{1}{1 + NT_s}$  на  $k_{u2}$ ,  $\frac{K_p + K_p NT_s + K_i T_s + K_i T_s^2 N + K_d N}{1 + NT_s}$  на  $k_e$ ,  $\frac{2K_p + K_p NT_s - K_i T_s + 2K_d N}{1 + NT_s}$  на  $k_{e1}$ ,  $\frac{K_p + K_d N}{1 + NT_s}$  на  $k_{e2}$ , рівняння (3.15) виглядатиме так:

$$\begin{aligned}
u[k] = k_{u1} u[k - 1] - k_{u2} u[k - 2] + k_e e[k] - k_{e1} e[k - 1] + k_{e2} e[k - 2]
\end{aligned} \tag{3.16}$$

Використовуючи рівняння (3.16) було написано алгоритм роботи ПД-регулятора на мікроконтролері, код представлений у «Додаток А».

### 3.5 Випробування системи

Після того, як було написано прошивку автопілота для польотного мікроконтролера STM32F303VCT6, слід встановити його у фюзеляж літака. Розташування плати зображено на рис. 3.22.

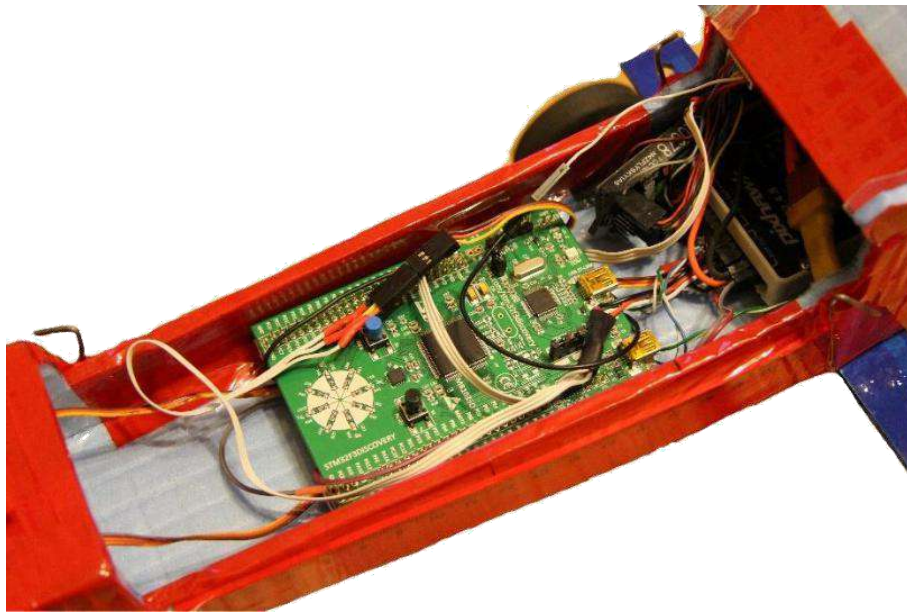


Рисунок 3.22 – Розташування мікроконтролера STM32 на борту літака

На рис.3.23 зображено графік зміни кута крену літака у часовій області. За цим графіком можна визначити якість перехідного процесу. Час перехідного процесу вийшов наступним:

$$T_p = 140 * 10^{-2} = 1.4 \text{с}$$

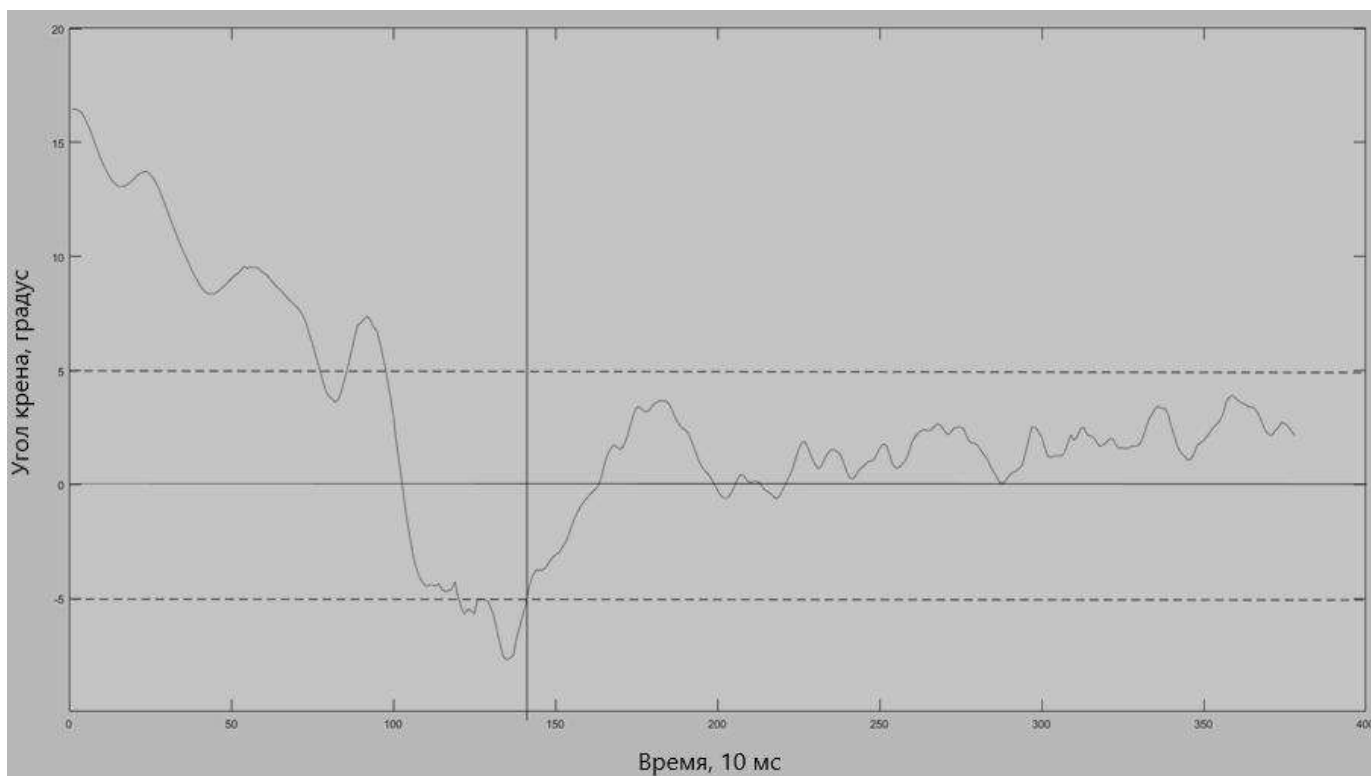


Рисунок 3.23 – Графік зміни крену літака у часовій області

## ВИСНОВКИ

В магістерській роботі:

- розроблено покращений алгоритм компенсації куту крену БПЛА літакового типу;

- спроектована структурна схема системи стабілізації безпілотного літального апарату літакового типу по крену;

- виконано підбір, монтаж та юстування необхідного обладнання для проведення лабораторних та польотних випробувань;

- отримано математичну модель керування кутом крену БПЛА літакового типу;

- використовуючи розроблену математичну модель, було проведено моделювання перехідного процесу визначення оптимальних параметрів ПІД-регулятора;

- було проведено конфігурування мікроконтролера STM32 у графічному середовищі STM32CubeMX, та його програмування мовою Сі у Keil uVision.

У процесі випробувань працездатності системи були виявлені особливості роботи регулятора, такі як несприятливий вплив інтегральної складової за вітряної погоди: внаслідок дії вітру відбувається накопичення інтегральної складової, що тягне за собою збільшення інерційності системи. Було вирішено зменшити інтегральний коефіцієнт, тим більше що система спочатку має астатизм першого порядку. Також було модифіковано алгоритм роботи регулятора, саме введено обмеження диференціальної складової за принципом насичення. Це дозволило зробити систему стійкішою у вітряну погоду. Також були виявлені труднощі в роботі з числами подвійної точності на мікроконтролері STM32F3: вбудований блок FPU має розрядність 32 біти, отже здатний працювати тільки з числами одинарної точності. Тому модифіковано алгоритм. В результаті доробок система показала свою стабільну роботу у процесі польоту літака.

У цій роботі було досягнуто завдання створення системи кутової стабілізації безпілотного літального апарату літакового типу по крену. Була вивчена



документація і по ній виконано підключення та налагодження наявного обладнання, а саме мікросхема гіроскопа L3GD20, мікросхема акселерометра LSM303DLHC, безпосередньо сам мікроконтролер STM32F303VCT6. Компоненти цієї системи були об'єднані до купи для створення робочого прототипу системи.

За допомогою засобів мови C було написано керуючий алгоритм для мікроконтролера та виконано його прошивку. Зрештою, вся система була протестована в реальному польоті, в якому була продемонстрована працездатність системи. Однак, показники якості перехідного процесу відрізняються від таких, отриманих при моделюванні системи.

Надалі планується додавання контурів стабілізації по тангажу та нишпоренню, а також провести управління центром мас апарату.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Nemati A, Kumar M (2014) Modeling and control of a single axis tilting quadcopter. In: American control conference (ACC), 2014, pp 3077–3082
2. Tanveer MH, Hazry D, Ahmed SF, Joyo MK (2014) NMPC-PID based control structure design for avoiding uncertainties in attitude and altitude tracking control of quadrotor (UAV). In: IEEE 10th international on signal processing & its applications (CSPA), Colloquium, 2014, pp 117–122
3. Sangyam T, Thailand Laohapiengsak P, Chongcharoen W, Nilkhamhang I (2010) Autonomous path tracking and disturbance force rejection of UAV using fuzzy based auto-tuning PID controller. In: International conference in electrical engineering/electronics computer telecommunications and information technology (ECTI-CON), (2010), pp 1–4
4. Salih AL, Moghavvemi M, Mohamed HAF, Gaeid KS (2010) Modelling and PID controller design for a quadrotor unmanned air vehicle. In: IEEE international conference in automation quality and testing robotics (AQTR), (2010), pp 1–5
5. Joyo MK, Hazry D (2013) Altitude and horizontal motion control of quadrotor UAV in the presence of air turbulence. In: IEEE conference in ICSPC, 2013, pp 16–20
6. Goodarzi F, Daewon L, Taeyoung L (2013) Geometric nonlinear PID control of a quadrotor UAV on SE(3). In: European control conference (ECC), 2013, pp 3845–3850
7. Sufendi Trilaksono BR, Nasution SH, Purwanto EB (2013) Design and implementation of hardware-in-the-loop-simulation for UAV using PID control method. In: 3rd international conference in instrumentation, communications, information technology, and biomedical engineering (ICICI-BME), 2013, pp 124–130
8. Castaneda, H., Salas-Pena, O. S., & de Leon Morales, J. (2013, May). Adaptive super twisting flight control-observer for a fixed wing UAV. In Unmanned Aircraft Systems (ICUAS), 2013 International Conference on (pp. 1004-1013). IEEE.
9. Chad C. Haddal “Homeland Security: Unmanned Aerial Vehicles and Border Surveillance “Congressional Research Service 2010

10. David F. Anderson & Scott Eberhardt, "Understanding Flight", McGraw-Hill, 2010, pp.95.
11. Lee, Jangho, and Jindeog Chung. "Point navigation guidance for small unmanned helicopter under predicting future position." Unmanned Aircraft Systems (ICUAS), 2013 International Conference on. IEEE, 2013.
12. M.Hassan Tanveer, S. Faiz Ahmed, D. Hazry, "Stabilized controller design for attitude and altitude controlling of quad-rotor underdisturbance and noisy conditions", American journal of applied sciences ,2013, vol. 10, no. 8, pp 819-831.
13. M. Kamran Joyo, D. Hazry, "Position Controller Design for Quad-rotor under Perturbed Condition", Wulfenia Journal, vol. 20, no. 7, 2013, pp 178-189.
14. S. Jackson, J. Tisdale, "Tracking controllers for small UAVs with wind disturbances: theory and flight results", in Proc. IEEE Conf. Decision Control, 2008, pp. 564-569.
15. Yang Chen; Tianmiao Wang; Jianhong Liang; Chaolei Wang; Chenghao Xue, "A fuzzy robust path following controller for a small unmanned air vehicle," Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on , vol., no., pp.1189,1194, 18-20.
16. HaiYang Chao, YongCan Cao and YangQuan Chen "Autopilots for Small Unmanned Aerial Vehicles: A Survey," International Journal of Control, Automation, and Systems (2010) 8(1):36-44
17. YAO Junkui, QI Zaikang, LI Fugui, GUO Tao, " Study on UAV Flight Based Identification Technology," 29th Chinese Control Conference July 29-31, 2010, Beijing, China
18. Taming Shih; Ho-chung Chang; , "FPGA based hardware in the loop test platform of small size UAV," Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on , vol., no., pp.551-556, 15-18 Dec. 2009

## ДОДАТОК А – КОД БІБЛІОТЕКИ, ЩО РЕАЛІЗУЄ АЛГОРИТМ ПІД-РЕГУЛЯТОРА

```
#include "PID.h"
```

```
float ku1; //Дискретний параметр при виході ПІД-регулятора на до-1 кроці
float ku2; //Дискретний параметр при виході ПІД-регулятора на до-2 кроці
float ku0; //дискретний параметр при сигналі неузгодженості на якому кроку
float ku1; //Дискретний параметр при сигналі неузгодженості на до-1 кроці
float ku2; //дискретний параметр при сигналі неузгодженості на до-2 кроці
```

```
float PIDmax; //максимальне значення регулятора
```

```
float PIDmin; //Мінімальне значення регулятора
```

```
int PID_SetPoint; //Уставка ПІД-регулятора
```

```
float PID_Error[3]; //помилка неузгодженості
```

```
float PID_Out [3]; //Вихід ПІД-регулятора
```

```
void PID_Ini(float Kp, float Ki, float Kd, float N, float PidMin, float PidMax, float dt, int SetPoint){
```

```
    PIDmax = PidMax;
```

```
    PIDmin = PidMin;
```

```
    PID_SetPoint = SetPoint;
```

```
    ku1 = (2+N*dt)/(1+N*dt);
```

```
    ku2 = -1/(1+N*dt);
```

```
    ke0 = (Kp*(1+N*dt)+Ki*dt*(1+N*dt)+Kd*N)/(1+N*dt);
```

```
    ke1 = -(Kp*(2+N*dt)+Ki*dt+2*Kd*N)/(1+N*dt);
```

```
    ke2 = (Kp+Kd*N)/(1+N*dt);
```

```

float PID_control(float angle){ // уставку задаватимемо через налаштування
    for (int i = 2; i > 0; i--){
        PID_Error[i]=PID_Error[i-1];
        PID_Out[i]=PID_Out[i-1];

        PID_Error[0] = PID_SetPoint - angle; //Порахуємо помилку
        PID_Out[0] = ku1*PID_Out[1]+ku2*PID_Out[2]+
        ke0*PID_Error[0] + ke1*PID_Error[1] + ke2*PID_Error[2];
        if (PID_Out[0] > PIDmax) PID_Out[0] = PIDmax;// limit to
        DAC або PWM range
        if (PID_Out[0] < PIDmin) PID_Out[0] = PIDmin; return PID_Out[0];}

    int PWMGen(float Signal){
        float tmp = 1500 + 0.08889 * Signal;
        /*if (Signal >= 4500) return 1900;
        else if (Signal <= -4500) return 1100;
        else*/
        return tmp; //відбувається ковертація оскільки математична модель
        була створена з використанням вхідних даних діапазону (-4500;4500) а
        ШІМ сигнал (1100;1900)}

```

**ДОДАТОК Б – ПРЕЗЕНТАЦІЯ**