Міністерство освіти і науки України
Кам’янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп’ютерних наук

Дипломна робота
магістра

з теми: «МЕТОДИ І ЗАСОБИ СТВОРЕННЯ КОМП’ЮТЕРНИХ СИМУЛЯТОРІВ ДИНАМІЧНИХ ОБ’ЄКТІВ»

Виконав: здобувач вищої освіти
2 курсу, групи KN1-M22
спеціальності 122 Комп’ютерні науки
Задворний Антон Олексійович

Керівник: **Федорчук В.А.,**
професор кафедри комп’ютерних наук,
доктор технічних наук, професор

Рецензент: **Оптасюк С.В.,**
кандидат фізико-математичних наук,
доцент, завідувач кафедри фізики

Кам’янець-Подільський – 2023

ЗМІСТ

ВСТУП

Поштовхом до розвитку безпілотної авіації у всьому світі стала потреба в легких та відносно дешевих літальних апаратах, що мають достатні маневрені характеристики та здатні виконувати широкий спектр завдань.

За допомогою дронів вертолітного типу люди вирішують безліч завдань у різних сферах діяльності. Безпілотні літальні апарати (БПЛА) успішно застосовуються в ході військових операцій в усьому світі, а також не менш успішно виконують завдання цивільного призначення. З роками перелік можливих застосувань буде лише збільшуватися. Високі польотні характеристики за відносно малої складності виготовлення і найчастіше невисокої вартості лише посилили популярність БПЛА.

Протягом останніх десятиліть завдання управління безпілотними літальними апаратами стає актуальнішим. Керування БПЛА – це завдання для добре підготовлених професіоналів. Наприклад, в армії США операторами БПЛА є діючі пілоти ВПС після річної підготовки та спеціальних тренінгів. У багатьох аспектах керування БПЛА навіть складніше, ніж звичайне пілотування літака.

На сьогоднішній день більшість існуючих безпілотних літальних апаратів пілотуються вручну, за допомогою пультів дистанційного керування, що працюють через радіоканали. При ручному управлінні БПЛА виникають труднощі, пов'язані з підготовкою операторів, недостатньою робочою дальністю, а також обмеженнями, що пов'язані з погодними умовами.

Однак конструкція БПЛА, що ґрунтується на кількох тягових двигунах, додатково передбачає низку значних вимог до здійснення роботи системи управління в рамках задоволення необхідності постійної стабілізації апарату в просторі. У зв'язку з цим ведеться розробка як методів загального призначення (переміщення БПЛА у просторі), так і допоміжних алгоритмів (стабілізація, управління аварійними ситуаціями тощо).

Як відомо, більшість аварій БПЛА, стаються через помилки пілотів–операторів та механічні відмови.

У роботі розглядається питання моделювання переміщень квадрокоптера для подальшої побудови такої системи управління, яка дозволяла б мінімізувати ймовірність негативних наслідків у разі серйозних несправностей. Природно вважати, що така (відмовностійка) система управління може працювати тільки на апаратах, які конструктивно допускають роботу в певному (аварійному) режимі, при якому порушено штатну роботу деяких підсистем апарату.

Мета: дослідити методи і засоби моделювання для відображення їх в симуляторах динамічних об’єктів у вигляді комп’ютерної анімації.

Об’єкт дослідження. Процеси створення комп’ютерних симуляторів.

Предмет дослідження. Середовище MATLAB з доповненням Simulink, для моделювання аварійних ситуацій.

Завдання дослідження: полягає у розробці комп’ютерної візуалізації динамічного об’єкта.

Структура роботи: Магістерська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі досліджується БПЛА як об'єкт керування.

- Визначення можливих переміщень об'єкта та моделі,
- Побудова математичної моделі, що описує квадрокоптер як тверде тіло з 6–ма ступенями свободи;
- Вибір параметрів системи на основі реальної конфігурації пристрою.

Другий розділ розкриває такі питання:

- Класифікація аварійних ситуацій;
- Визначення аварійної ситуації з погляду математичної моделі квадрокоптера.
- Опис алгоритмів керування квадрокоптером в аварійних ситуаціях.

- Формалізація методу оцінки відстані посадки квадрокоптера для подальшого використання в алгоритмах стійкості до відмов приземлення.

Третій розділ присвячений чисельним експериментам:

- Моделювання аварійних ситуацій.
- Моделювання аварійного переміщення з алгоритмом «ручного» приземлення.
- Моделювання аварійної ситуації з використанням відмовного алгоритму на основі ПІД–контролера.
- Експеримент щодо оцінки відстані приземлення БПЛА в аварійній ситуації.

Додатковою метою є написання зручного програмного забезпечення щодо чисельних експериментів.

РОЗДІЛ 1. МОДЕЛЮВАННЯ КВАДРОКОПТЕРА ЯК ДИНАМІЧНОГО ОБ'ЄКТА

Динамічною системою зазвичай називають реальний об’єкт, поводження якого з задовільною для досліджування точністю може бути описано системою звичайних диференціальних рівнянь, аргументом яких є час.

Як бачимо, визначення динамічної системи – суто математичне. Воно стосується будь–яких фізичних (механічних, електричних, теплових тощо) і навіть біологічних процесів, поводження яких можна описати за допомогою диференціальних рівнянь.

1.1. Математична модель

Квадрокоптер є різновидом літального апарата з вертикальним вектором тяги [1, 2], що рухається за допомогою чотирьох двигунів (роторів) зі швидкостями обертання Ω_1, Ω_2, Ω_3, Ω_4, закріпленими у центрі мас апарата M на двох металевих балках, що перетинаються хрест–навхрест, двигуни обертаються діагонально у протилежних напрямах (Рис. 1.1).

Багатороторні літальні апарати можуть мати різні конструктивні конфігурації – схеми. Квадрокоптер, як частковий випадок багатороторних апаратів може будуватися за Х– або плюс–подібній схемі.

Відповідно до роботи [2], дана конструкція досить жорстка і єдине, що може змінюватися, це кутові швидкості пропелерів. Осі обертання гвинтів нерухомі і паралельні.

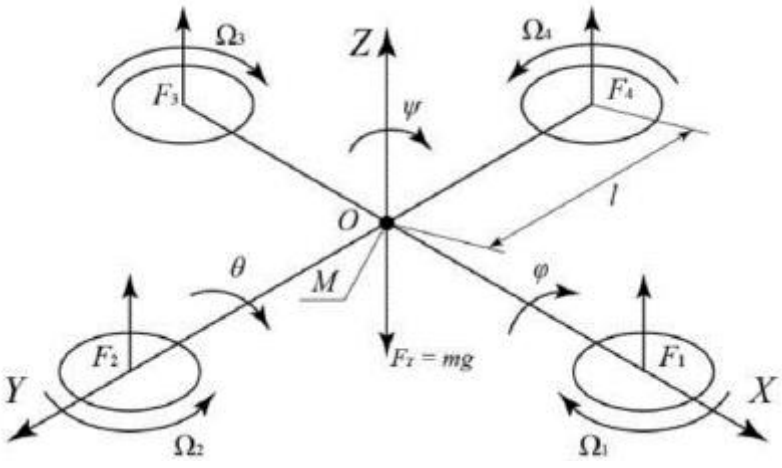


Рис. 1.1 – Основні параметри квадрокоптера

Основні параметри квадрокоптера подані на рис. 1.1.

На апарат діють підйомні сили двигунів F_1, F_2, F_3, F_4, сила тяжіння F_T. Точка M – центр мас апарату (збігається з геометричним центром), l –відстань від центру апарата до точки прикладання підйомних сил.

Нерухома система координат жорстко пов'язана із землею: Вісь X’ спрямована на північ, Y’ на захід, Z’ вгору по відношенню до землі. Рухома система координат жорстко пов'язана з корпусом апарату: вісь X спрямована вздовж напрямку руху квадрокоптера вперед, Y – вздовж напрямку квадрокоптера вліво, Z – вгору. Обидві системи координат є правосторонніми.

Гвинти двигунів, що знаходяться на осі X, обертаються за годинниковою стрілкою, а гвинти на осі – Y проти годинникової стрілки. Така конфігурація протилежних напрямків пар усуває потребу у хвостовому гвинті, що є у стандартній конструкції гелікоптера. Стрілками Ω_1, Ω_2, Ω_3, Ω_4 показані напрямки обертання гвинтів.

Будучи об'єктом з 6–ма ступенями свободи, квадрокоптер оснащений всього чотирма гвинтами, що дозволяє досягти бажаного контролю лише для 4–х величин. Однак, завдяки структурі апарату, досить легко вибрати чотири найкращі контрольовані змінні та розділити їх, щоб спростити керування. Таким чином, при побудові моделі доцільно визначити чотири основні переміщення, які дозволяють квадрокоптеру досягати певної висоти та положення:

1. Прискорення вгору/вниз ($U_1 \rightarrow \ddot{Z}$). Ця команда характеризується зміною кутових швидкостей всіх чотирьох гвинтів на однакову величину.
2. Кутове прискорення крену ($U_2 \rightarrow \ddot{\varphi}$). Така команда характеризується зміною кутових швидкостей гвинтів на осі Y таким чином, що апарат починає обертальний рух навколо осі X.
3. Кутове прискорення тангажу ($U_3 \rightarrow \ddot{\theta}$). Ця команда характеризується зміною кутових швидкостей гвинтів на осі X таким чином, що апарат починає обертальний рух навколо осі Y.
4. Кутове прискорення рискання ($U_4 \rightarrow \ddot{\psi}$). Ця команда характеризується зміною кутових швидкостей всіх чотирьох гвинтів,

причому гвинти, що знаходяться на різних осях, одержують протилежний вплив таким чином, що апарат починає обертальний рух навколо осі Z.

Таким чином переміщення квадрокоптера можна вважати сумою поступального руху центру мас та сферичного руху тіла відносно центру мас. Такий рух може бути описано наступною системою [3]:

$$\begin{gathered}
\frac{dx}{dt} = V_x, \frac{dy}{dt} = V_y, \frac{dz}{dt} = V_z, \\
m\frac{dV_x}{dt} = (\sin\psi \sin\varphi + \cos\psi \sin\theta \cos\varphi)U_1, \\
m\frac{dV_y}{dt} = (-\cos\psi \sin\varphi + \sin\psi \sin\theta \cos\varphi)U_1, \\
m\frac{dV_z}{dt} = U_1 \cos\theta \cos\varphi - mg, \\
\frac{d\varphi}{dt} = \omega_\varphi, \frac{d\theta}{dt} = \omega_\theta, \frac{d\psi}{dt} = \omega_\psi, \\
I_{xx}\frac{d\omega_\varphi}{dt} = \left(I_{yy} - I_{zz}\right)\omega_\theta\omega_\psi - J_{TP}\omega_\theta\Omega + U_2, \\
I_{yy}\frac{d\omega_\theta}{dt} = (I_{zz} - I_{xx})\omega_\psi\omega_\varphi - J_{TP}\omega_\varphi\Omega + U_3, \\
I_{zz}\frac{d\omega_\psi}{dt} = \left(I_{xx} - I_{yy}\right)\omega_\psi\omega_\theta + U_4,
\end{gathered} \tag{1.1}$$

де *x*, *y*, *z* – координати центру мас, V_x, V_y, V_z – проекції вектора лінійної швидкості, φ, θ, ψ – кут крену, тангажу та рискання відповідно, φ – кутова швидкість крену, θ – кутова швидкість тангажу, ψ – кутова швидкість рискання, m – маса квадрокоптера, I_{xx}, I_{yy}, I_{zz} – моменти інерції навколо осей X, Y і Z відповідно, U_1, U_2, U_3, U_4 – канали управління, **Ω** – загальна швидкість чотирьох гвинтів, J_{TP} – загальний обертальний момент інерції навколо осі гвинта:

$$J_{TP} = J_P + \mu N^2 J_M, \tag{1.2}$$

де J_P момент інерції двигуна, J_M передатне число, μ – момент інерції пропелера, N – ефективність передачі.

Рівняння зв'язку каналів управління U_1, U_2, U_3, U_4 зі швидкостями обертання гвинтів Ω_1, Ω_2, Ω_3, Ω_4 мають вигляд:

$$\begin{gathered} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2), \\ U_2 = lb(-\Omega_1^2 + \Omega_3^2), \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \end{gathered} \tag{1.3}$$

де l – відстань між центрами квадрокоптера та гвинта, b та d — аеродинамічні складові тяги та коефіцієнту опору відповідно. Квадрокоптер приводиться обертанням гвинтів, швидкості яких можна виразити з системи рівнянь (1.3):

$$\begin{cases} \Omega_1 = \sqrt{\frac{1}{4b}U_1 - \frac{1}{2bl}U_3 - \frac{1}{4d}U_4}, \\ \Omega_2 = \sqrt{\frac{1}{4b}U_1 - \frac{1}{2bl}U_2 + \frac{1}{4d}U_4}, \\ \Omega_3 = \sqrt{\frac{1}{4b}U_1 + \frac{1}{2bl}U_3 - \frac{1}{4d}U_4}, \\ \Omega_4 = \sqrt{\frac{1}{4b}U_1 + \frac{1}{2bl}U_3 + \frac{1}{4d}U_4} \end{cases} \tag{1.4}$$

Таким чином, можна визначити завдання управління квадрокоптером як завдання побудови стратегії управління швидкостями обертання чотирьох моторів $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{\Omega}_3$, $\boldsymbol{\Omega}_4$ так, щоб забезпечити асимптотично стійке положення квадрокоптера x_0, y_0, z_0 з утриманням одного з кутів (наприклад, кута рискання). При цьому сама точка x_0, y_0, z_0 вибирається відповідно до польотного завдання.

Важливо відзначити, що описана система є математичною моделлю деякого «ідеального» квадрокоптера, що переміщується у середовищі без збурень. Така модель дозволяє досліджувати як характер зміни величин, що спостерігаються в часі, так і їх залежності. Емпіричний підбір констант у моделі дає можливість досить точно наблизити розрахункові результати до спостерігається на реальному апараті, проте повністю замінити систему управління квадрокоптером, побудовану на датчиках і регуляторах, такою системою неможливо.

1.2. Вибір параметрів системи на основі реальної конфігурації пристрою

Ринок БПЛА рясніє великою різноманітністю різних конфігурацій: від маленьких пристроїв до великих вантажних. Конструктивний підхід до класифікації БПЛА наведений у [4].

В даній роботі буде розглянута задача керування саме легким квадрокоптером. Слово «легкий» обрано не випадково, оскільки для достатньо важких квадрокоптерів втрата навіть одного гвинта призводить до швидкого падіння та на роботу аварійного закону керування просто може не вистачити часу. Слово «легкий» можна замінити на «мікро», скориставшись відомою класифікацією літальних апаратів [5], більш повна класифікація наведена в документі [6].

У подальшому розгляді модель матиме параметри, що відповідають квадрокоптеру з діаметром рами 350 мм [7, 8]:

Таблиця 1 — Приклад конфігурації відмовостійкого квадрокоптера з рамою діаметром 350 мм

Характеристика	Значення
Ємність батареї	3 А·ч
Запаси енергії	44 Вт·ч
Навнтаження на акумулятор	19 Кл
Номінальна напруга АКБ	14,8 В
Маса апарата (споряджена маса)	1 кг
Сукупна маса двигунів (роторів)	0,726 кг
Модель двигуна	Turnigy Multistar Elite 2810–750
Тяговооруженность тягоозброєність	3,6 споряджених мас
Удельная тяга питома тяга	8,43 г/Вт
Мінімальний польотний час	2,8 хв.
Час зависання	19,9 хв.

Максимальна горизонтальна швидкість	54 км/год
Газ зависання (лінійний)	42%
Струм електродвигуна (зависання)	1,92 А
Струм електродвигуна (максимальний режим)	14,27 А
Максимальний струм ЕРХ електродвигуна	20 А
ККД (зависання)	83,9 %
ККД (максимальний режим)	85,1 %

Параметри математичної моделі (1.1), що відповідають апарату вказаної конфігурації мають вигляд:

$$m = 1 \text{ кг}, l = 0{,}175 \text{ м}, b = 26{,}5 \cdot 10^{-6} \text{Н} \cdot \text{м} \cdot \text{с}^2,$$
$$d = 0{,}6 \cdot 10^{-6} \text{Н} \cdot \text{м} \cdot \text{с}^2, I_{xx} = I_{yy} = I_{zz} = 0{,}1 \text{ Н} \cdot \text{м} \cdot \text{с}^2,$$
$$J_{TP} = 0{,}005 \text{ Н} \cdot \text{м} \cdot \text{с}^2 \qquad (1.5)$$

РОЗДІЛ 2. МОЖЛИВІ СПОСОБИ КЕРУВАННЯ КВАДРОКОПТЕРОМ В АВАРІЙНИХ СИТУАЦІЯХ

У цьому розділі описані особливості побудови математичної моделі квадрокоптера, а саме:

1. Визначення та класифікація аварійних ситуацій;
2. Дії рятувального алгоритму;
3. Визначення можливих управлінь квадрокоптером;
4. Алгоритм дії ПІД–регулятора;
5. Оцінка дальності приземлення апарату у аварійної ситуації.

2.1. Визначення та класифікація аварійних ситуацій.

Популярність БПЛА, особливо квадрокоптерів, спричинила появу великої кількості робіт присвячених керуванню цим класом літальних апаратів. Наприклад у [9] проведена класифікація алгоритмів керування для різних моделей квадрокоптерів з якісним аналізом цих алгоритмів. Однак серед цього розмаїття можна виділити цілі напрями, які погано пророблені або присутні у роботах різних авторів на рівні формулювання проблеми. Один з таких напрямів — керування квадрокоптером у позаштатних ситуаціях. Одній з перших робіт на цю тему є [10]. Заслугою авторів є те, що вони класифікували позаштатні ситуації та запропонували методи вирішення, але повна відмова двигуна як позаштатна ситуація не розглядалася. Скористаємось результатами [10] та введемо нову, розширену класифікацію несправностей:

1. Падіння тяги гвинта не нижче критичного значення. З практичних міркувань критичне значення вибирається рівним 30% максимальної тяги. Відбувається зазвичай за нерівномірного розряду батареї.

2. Тимчасова відмова гвинта або тимчасове падіння тяги нижче 30% від максимальної тяги. Інтервал часу, у якому відбуваються дані події досить короткий, тобто час, протягом якого проявляється несправність, менший за час, який потрібен для обчислення управління.

3. Повна відмова гвинта або падіння тяги нижче 30% від максимальної тяги.

4. Повна відмова двох симетричних гвинтів.

5. Повна відмова двох суміжних гвинтів.

6. Повна відмова трьох гвинтів.

Використовуючи наведену класифікацію можна навести кілька робіт у яких з різним ступенем успішності розв’язується задача керування у випадку виникнення однієї або кількох несправностей.

Для спрощення тип несправності буде відповідати номеру у запропонованій класифікації. Отже, у [10] основна увага приділена розв’язанню задачі з несправностями №1 та №2, але питання про їх виявлення під час польоту не розглядалося. У [11] наведено розв’язання задачі керування квадрокоптером при виникненні несправності №3. В [12] запропоновано розв’язання задачі управління гексакоптером при виникненні несправності №3. Принципова відмінність від попередньої роботи полягає у тому, що наявність двох додаткових гвинтів дозволяє перевести політ з позаштатної ситуації у штатний режим, використовуючи лише перерозподіл тяги п’яти гвинтів що лишилися. Аналогічне рішення запропоновано у [13], але вже для октокоптерів, відповідно з надлишковістю у чотири гвинти.

В [14–16] запропонований метод виявлення несправностей №№ 1 и 2 у квадрокоптері, що дозволяє посадити апарат. Найбільш цікавим є підхід врахування несправностей при прокладанні шляху. Автори пропонують прокладати таку траєкторію руху квадрокоптера, щоб він у випадку несправностей типу №№ 1, 2 все ж мав можливість за нею пролетіти.

Зауважимо, що квадрокоптер у випадку виникнення несправності типу №3 можна вважати вже трикоптером, тому наведемо кілька робіт, я яких розглядається керування трикоптером. У [17, 18] ставиться та розв’язується задача керування трикоптером особливої конструкції, в якому мотори можуть відхилятися від вертикального положення, змінюючи вектор тяги. На жаль, таке конструктивне рішення для квадрокоптера важко втілити, тому широкого розповсюдження цей підхід не отримав. У [19] ставиться та розв’язується задача керування трикоптером, в якому мотори зафіксовані та повернуті на певний кут відносно горизонтальної та вертикальної осей.

Екзотичний об’єкт, який формально можна віднести до квадрокоптеру з несправністю типу №6 докладно описаний у [20]. На жаль, конструктивна витонченість літального апарату безпосередньо пов'язана з його керованістю та специфічним рухом у повітрі, що не дуже схоже на квадрокоптер у якого працює один гвинт.

2.2. Рятувальний алгоритм

Стосовно моделі (1.1) розглянемо ситуації, пов'язані з виходом з ладу одного або кількох гвинтів. Причиною відмови пропелера можуть бути різні причини. Тим не менш, з точки зору моделі порушення працездатності може бути представлене як миттєва зміна відповідних величин:

$$\Omega_i^{em} = \Omega_i - \varepsilon_i, \varepsilon_i \in [0, \Omega_i], i = \overline{1,4} \quad (2.1)$$

Величина ε_i, має досить простий практичний сенс – кількісне погіршення (в обертах на секунду) працездатності гвинта внаслідок зовнішніх (торкання пропелера сторонніх поверхонь) або внутрішніх (збій електричної частини двигуна) несправностей. Випадок, коли $\varepsilon_i = \Omega_i$, і, отже, $\Omega_i^{em} = 0$, описуватиме ситуацію з повною непрацездатністю відповідного двигуна. Підстановка аварійних значень Ω_i^{em} у вирази для елементів управління (1.3) і далі у систему (1.1) визначить математичну модель аварійної ситуації.

Маючи постановку завдання моделювання аварійної ситуації, доцільно також запровадити план як зовнішньої (людина), так і внутрішньої (програма) реакції на виникнення позаштатного становища у вигляді відповідної обставин методології (посадка квадрокоптера «вручну» або автоматизоване приземлення апарату). Таким чином, далі під рятувальним алгоритмом мається на увазі управління двигунами апарату таким чином, щоб стабілізувати апарат за висотою, мінімізуючи ймовірність неконтрольованого падіння:

$$\Omega_i = \Omega_i^{fs}, \qquad i = \overline{1,4} \quad (2.2)$$

де Ω_i^{fs} кутові швидкості обертання гвинтів, що визначаються аварійною програмою посадки. Підстановка значень Ω_i^{fs} у вирази для елементів управління (1.3) і далі в систему (1.1) описує вплив рятувального алгоритму у рамках математичної моделі аварійної ситуації.

2.3. Визначення можливих управлінь квадрокоптером

Враховуючи описані раніше у п. 1.1 1 розділу можливі переміщення апарату, для реалізації стандартних типів рухів квадрокоптера (зліт, посадка, рух у горизонтальній площині) можуть бути використані керуючі сигнали виду:

$$\Omega_i = \begin{cases} C_i, & \text{якщо } t \in T_1, \\ C_i + \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cdot \cos(kt) + b_k \cdot \sin(kt)), & \text{якщо } t \in T_2. \end{cases} \quad (2.3)$$

Тут C_i – кутова швидкість, необхідна для компенсації сили тяжіння, а також параметри тригонометричного багаточлена, що задають відхилення, необхідні для гладкого виходу на стаціонарний режим, T_1 – множина відрізків часу, при яких апарат підтримує висіння, T_2 – множина відрізків часу, при яких апарат маневрує.

Таке подання керуючих сигналів близько (є наближенням за допомогою тригонометричного багаточлена) до оптимального за витратами енергії і при цьому є гладкою функцією, що застосовується в практичному сенсі. Іншими словами, така форма дозволяє апроксимувати сигнали Ω_i^{fs} з виразу (2.2) при симуляції реалізації рятувального алгоритму, та Ω_i^{em} з (2.1) при моделюванні наслідків аварій.

За допомогою безперервного аналізу даних з гіроскопів, акселерометрів та даних про швидкість обертання гвинтів можна визначити відмову одного із двигунів. Ознаки порушення працездатності, такі як руйнування гвинта без виходу з ладу електричної частини (відмова електричної частини ротора), спричиняють різку зміну швидкості обертання гвинта, що не є нормою у стандартному режимі польоту. Більш того, подібна несправність провокує обертання апарату з наступним неконтрольованим падінням. Тим не менш,

наявність цих ознак може бути виявлена польотним контролером з високою точністю не більш ніж за 0,5 секунд.

Таким чином, рятувальний алгоритм передбачає здійснення заходів, спрямованих на керування висотою апарату щодо його посадки. Як було сказано раніше, реалізація даної методології може ґрунтуватися як на зовнішньому впливі (людина), так і на внутрішній конфігурації (автоматичне перемикання з нормального режиму управління в аварійне та подальше управління відповідно до відмово стійкого алгоритму). Чисельне моделювання аварійної ситуації з «ручним» керуванням описано в п. 3.2. Симуляція позаштатного положення з автоматизованою реалізацією рятувального алгоритму показано у п. 3.3.

2.3 Алгоритм впливу ПІД–контролера на квадрокоптер

Сформулюємо методологію відмовостійкого приземлення для випадку програмної реакції на виникнення аварійної ситуації. Для стабілізації положення апарату щодо висоти було прийнято рішення використовувати алгоритм на основі ПІД–контролерів. Тому спочатку розглянемо рівняння вихідного сигналу ПІД–регулятора:

$$u(t) = P + I + D = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt}, \quad (2.4)$$

де K_p, K_i, K_d – коефіцієнти посилення пропорційної, інтегруючої та диференціюючої складових регулятора.

Кожен з параметрів ПІД–контролера має ряд особливостей, які важливо враховувати при налаштуванні регулятора для квадрокоптера з конкретною конфігурацією:

1. Пропорційна складова K_p:

- Посилення впливу пропорційної складової шляхом збільшення коефіцієнта K_p забезпечує збільшення стійкості. Однак важливо зауважити, що надто велике значення K_p може призвести як до різких коливань, так і до втрати керованості.
- Послаблення впливу пропорційної складової шляхом зменшення коефіцієнта K_p, зменшує силу реакції апарату на керуючий вплив.

2. Інтегральна складова K_i:

- Підвищення впливу інтегральної складової шляхом збільшення коефіцієнта K_i, забезпечує посилення курсової стійкості (апарат набуває покращеної інертності) та послаблює дрейф. Проте разом з цим зростає тривалість повернення у початкове становище. Також важливо зауважити, що при постійному K_p збільшення K_i призводить до зменшення впливу K_p, на керуючий вплив.
- Ослаблення впливу інтегральної складової шляхом зменшення коефіцієнта K_i збільшує швидкість реакції квадрокоптера на керуючий вплив, але разом з цим посилює дрейф та погіршує здатність утримувати стабільне положення.

3. Диференційна складова K_d

- Посилення впливу диференціальної складової шляхом збільшення коефіцієнта K_d підвищує швидкість стабілізації при зміні положення апарату в просторі після зовнішнього впливу або керування. Також зростання K_d значно збільшує вплив пропорційної складової, що сприяє підвищенню ймовірності появи надмірного регулювання та осциляцій.
- Ослаблення впливу диференціальної складової внаслідок зменшення коефіцієнта K_d зменшує швидкість та розмір коливань у процесі повернення апарату в стабілізоване становище.

Вказані вище особливості були враховані при доборі параметрів ПІД–контролера в рамках чисельного моделювання п. 3.3. Загальний план використання ПІД–регуляторів при реалізації автоматичного алгоритму відмовостійкого приземлення з метою безпечної посадки апарата має такий вигляд:

1. На ділянці зниження ($t \in [t_1, t_2]$) ПІД–регулятор має набір параметрів для стабілізації швидкості зниження до заданої величини;
2. На ділянці гальмування ($t \in [t_3, t_4]$) ПІД–регулятор має набір параметрів для гасіння вертикальної швидкості на момент торкання землі ($z' = 0$ при $z = 0$).

2.4. Метод оцінки відстані посадки квадрокоптера

БПЛА мають широке поширення як серед цивільного населення, так і в промисловому середовищі. Тому кількість різних ситуацій, у які може потрапити апарат, дуже різноманітна. У разі виникнення аварії важливо розуміти, в якому середовищі знаходиться апарат, що його оточує. Квадрокоптер може виявитися несправним, перебуваючи над поверхнею води або навіть вогнем. Будь–яка подібна ситуація тягне за собою обмеження на посадку апарата, які, якщо їх не враховувати при реалізації алгоритму відмово–стійкого приземлення, можуть значно вплинути на кінцевий результат. З огляду на це стає очевидним, що мінімізація ймовірності безконтрольного падіння не є достатньою умовою для успішного проведення операції порятунку як квадрокоптера, так і корисного навантаження (яка може не мати захисту від води/вогню). У зв'язку з цим пропонується підхід до оцінки відстані посадки БПЛА, що ґрунтується на проведенні низки чисельних експериментів.

У межах математичної моделі (п. 1.1) розуміння динаміки переміщення апарату доцільно провести чисельне моделювання зі зміною деяких параметрів. При аналізі аварійних ситуацій такими параметрами можуть бути:

1. Величина Ω_i^{em} двигуна, що виявився несправним, внаслідок однієї з можливих причин несправності;
2. Час початку реакції виникнення аварійної ситуації (алгоритм «ручної» посадки або реалізація стійкого до відмови алгоритма з використанням ПІД–регуляторів). Позначимо цей параметр як $t_{reaction}$.

Змінюючи певним чином дані параметри, з теореми про неперервність рішень систем диференціальних рівнянь від правих частин, ми зможемо отримати сукупність точок приземлення апарату, яка і визначатиме очікувану зону падіння квадрокоптера в аварійної ситуації. Реалізація цього методу представлена у п. 3.4.

РОЗДІЛ 3. ЧИСЕЛЬНІ ЕКСПЕРИМЕНТИ

Цей розділ складається з опису чисельних експериментів, що демонструють можливості як математичної моделі, так і підходів до взаємодії з нею. У середовищі MATLAB з доповненням Simulink [21] розроблено програмний пакет (див. додатки А та Б), який дозволяє моделювати аварійні ситуації та вирішувати проблеми управління з метою мінімізації наслідків аварійних ситуацій.

У першому пункті пропонується дослідити наслідки повної втрати тяги на одному з двигунів квадрокоптера за відсутності будь–яких алгоритмів порятунку. Результатом моделювання буде картина можливих наслідків такої відмови.

Для моделювання вибрано такі аварійні ситуації:

1. З погіршенням працездатності одного двигуна при горизонтальному польоті,
2. З повною втратою функціонування одного гвинта при зниженні апарату.

Моделювання обох ситуацій відбувається при $0 < t < 15$ з відмовою у момент часу t^*.

Другий пункт містить демонстрацію рятувальної методології [7; 8]:

1. Алгоритм «ручної» безпечної посадки квадрокоптера за участю зовнішнього управління (оператор БПЛА) з урахуванням міркувань щодо виявлення несправностей, описаних у п. 2.2.
2. Автоматичний відмовостійкий алгоритм на основі ПІД–регуляторів, описаних у п. 2.3.

Метою третього пункту є формалізація методу оцінки очікуваної області падіння (або безпечного приземлення) квадрокоптера (див. п. 2.4) при виникненні аварійної ситуації (або реалізації рятувального алгоритму при виявленні несправності апарату).

При реалізації чисельних експериментів використовують параметри квадрокоптера (1.5), що відповідають конфігурації з п. 1.2.

3.1. Моделювання аварійних ситуацій

3.1.1. Аварія під час горизонтального руху

Проведемо моделювання часткової втрати працездатності одного з двигунів (зменшення тяги) у процесі горизонтального переміщення квадрокоптера. Аварії подібного характеру можуть відбуватися в різних ситуаціях, наприклад у процесі перебування у безпосередньої близькості з об'єктами, що досліджуються. Навіть у випадку торкання одним з гвинтів дрона нерухомого об'єкта висока імовірність порушення цілісності його лопатей, що в кращому випадку тягне за собою зменшення тяги пошкодженого двигуна, а в найгіршому – повну втрату працездатності.

У межах математичної моделі квадрокоптера кутові швидкості обертання Ω_i визначаються на основі раніше описаного у п. 2.2 способу завдання управлінь апаратом, тобто. проводиться наближення сигналу Ω_i до оптимального. Для цього (2.3) будуються у формі:

$$\Omega_i = \begin{cases} \mathrm{C}, & t \in T_1, \\ \mathrm{C} + \sin\left(\frac{\pi}{4}t\right), & t \in T_2. \end{cases}$$

1. У момент часу t_0 = 0 с усі двигуни обертаються з необхідною для компенсації сили тяжіння швидкістю C = 304,1691 об/с;
2. Кутові швидкості $\Omega_{1,2}$ до моменту t^* = 4 с задаються поліномом, далі визначені як C;
3. У момент часу t = t^* відбувається часткова відмова третього двигуна (Ω_3 = 100 об/с при t ≥ 4с).

На рис. 3.1 представлений графік кутових швидкостей гвинтів апарату. Рух апарата при даному способі завдання Ω_i, починаючи з моменту t*, є падінням по спіралі (рис. 3.4).

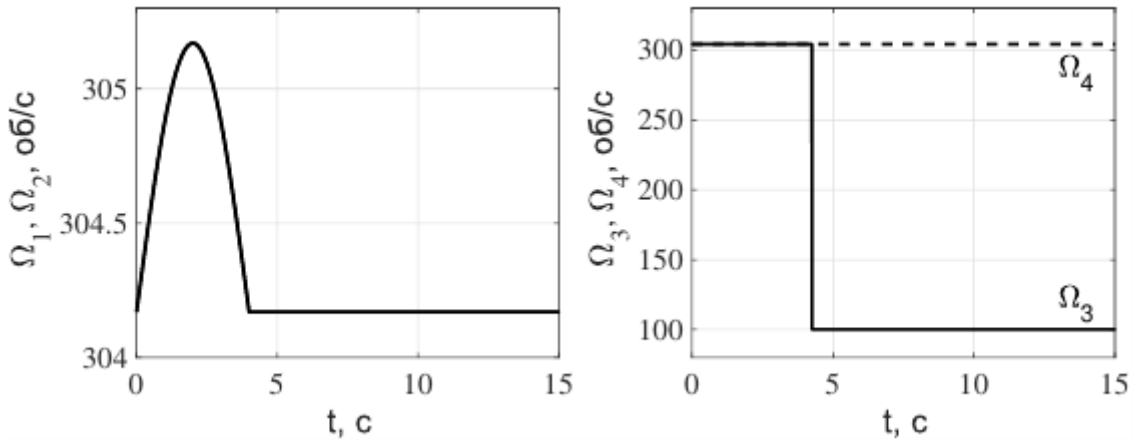


Рис. 3.1 – Аварія при горизонтальному польоті: кутові швидкості Ω_i

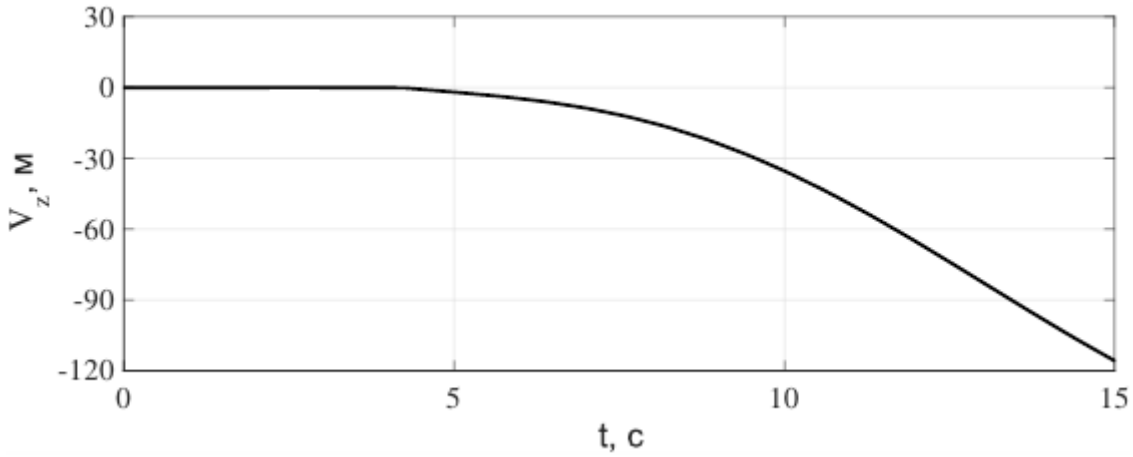


Рис. 3.2 – Аварія при горизонтальному польоті: швидкість V_z

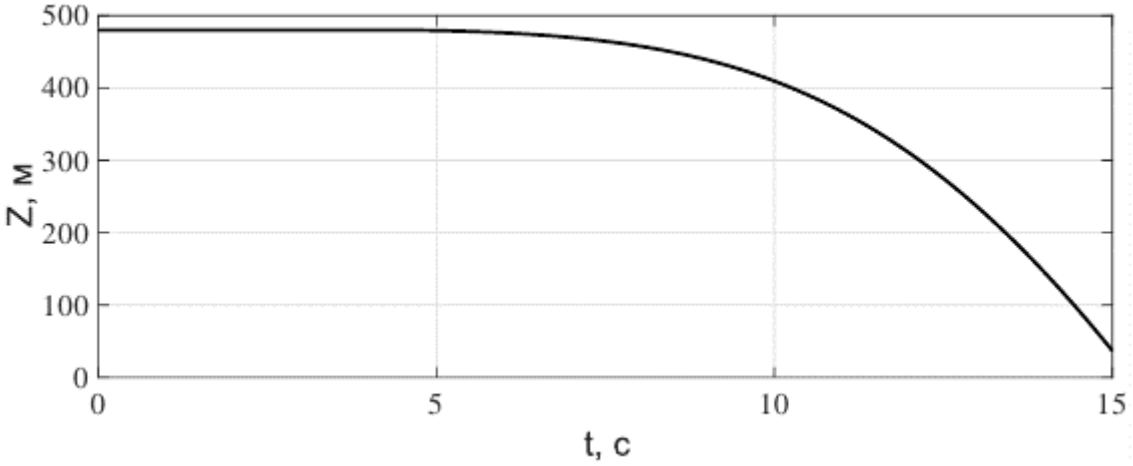


Рис. 3.3 – Аварія при горизонтальному польоті: висота Z

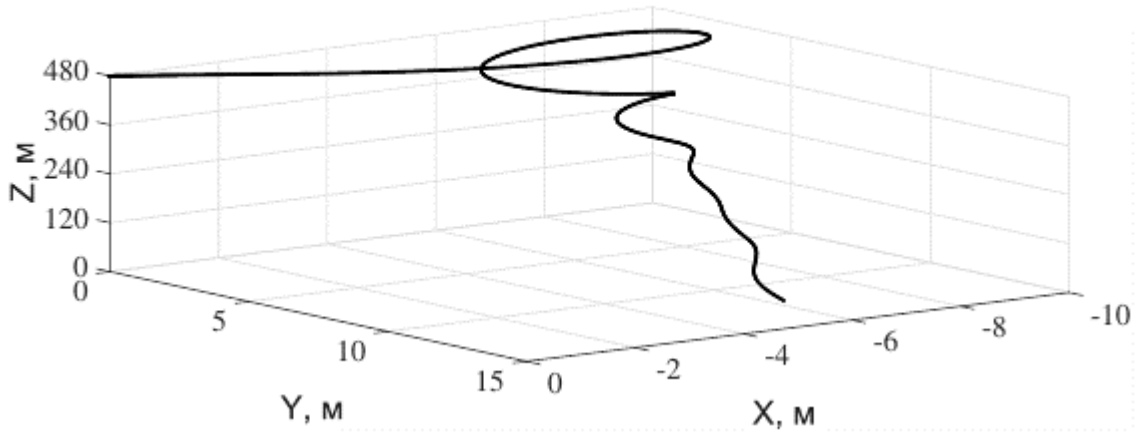


Рис. 3.4 – Аварія при горизонтальному польоті: траєкторія переміщення апарата

3.1.2. Аварія під час вертикального переміщення

Нехай аварійна ситуація настала у процесі зниження апарату після нерухомого висіння в точці (x_0, y_0, z_0) = (0, 0, 250). З точки зору математичної моделі отримаємо такі зміни кутових швидкостей Ω_i:

$$\Omega_i = \begin{cases} 304{,}1691, & \text{при } t \in T_1, \\ 304{,}1691 + 20\sin\left(\frac{\pi}{4}t\right), & \text{при } t \in T_2. \end{cases}$$

1. При t ∈ [0; 4] с всі двигуни обертаються з необхідною для компенсації сили тяжіння швидкістю С = 304,1691 об/с;
2. При t ∈ [4; 5] с апарат отримує негативне вертикальне прискорення внаслідок зниження кутових швидкостей;
3. При t ∈ [5; 6] с проводиться повернення кутових швидкостей до необхідних для висіння значень (відбувається компенсація від'ємного прискорення);
4. При t = t^* = 6 с відбувається аварійна ситуація – обнулення кутовий швидкості Ω_1.

На рис. 3.5 представлений графік кутових швидкостей гвинтів апарату. Рух апарата при даному способі завдання Ω_i, починаючи з моменту часу t^*, являє собою безконтрольне падіння (рис. 3.8).

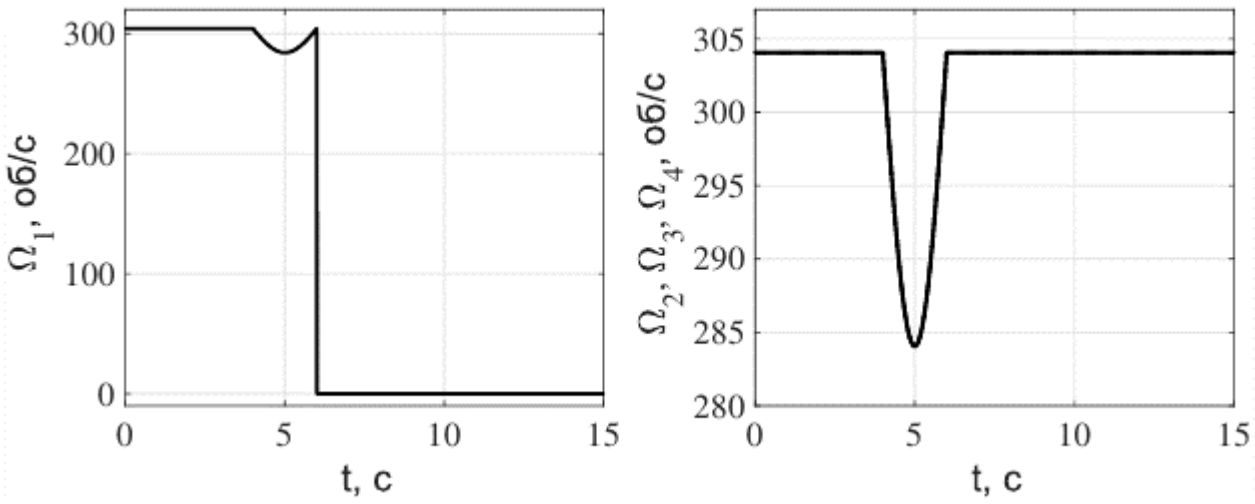


Рис. 3.5 – Аварія при зниженні: кутові швидкості Ω_i

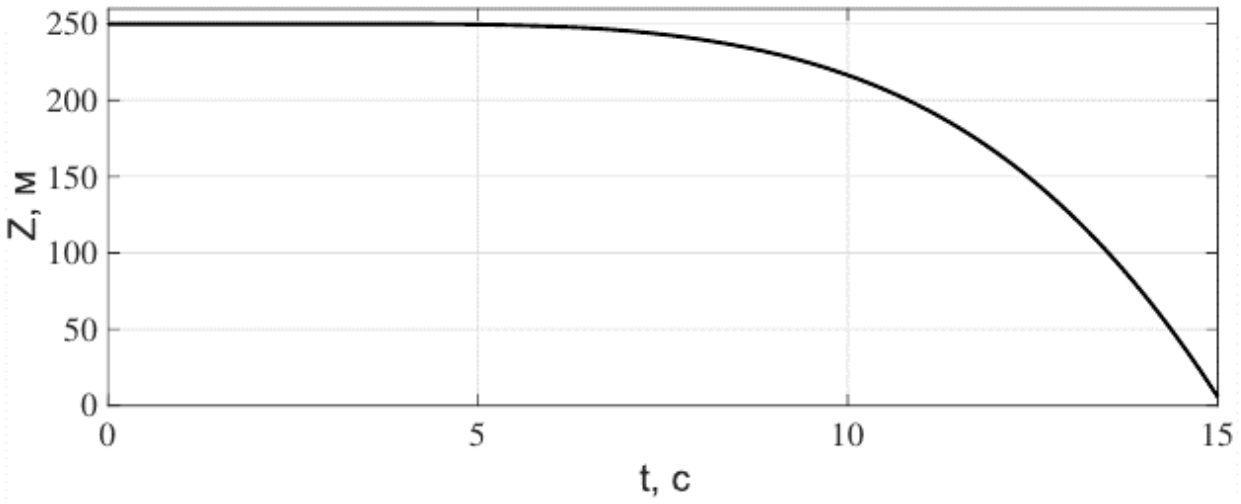


Рис. 3.6 – Аварія при зниженні: висота Z

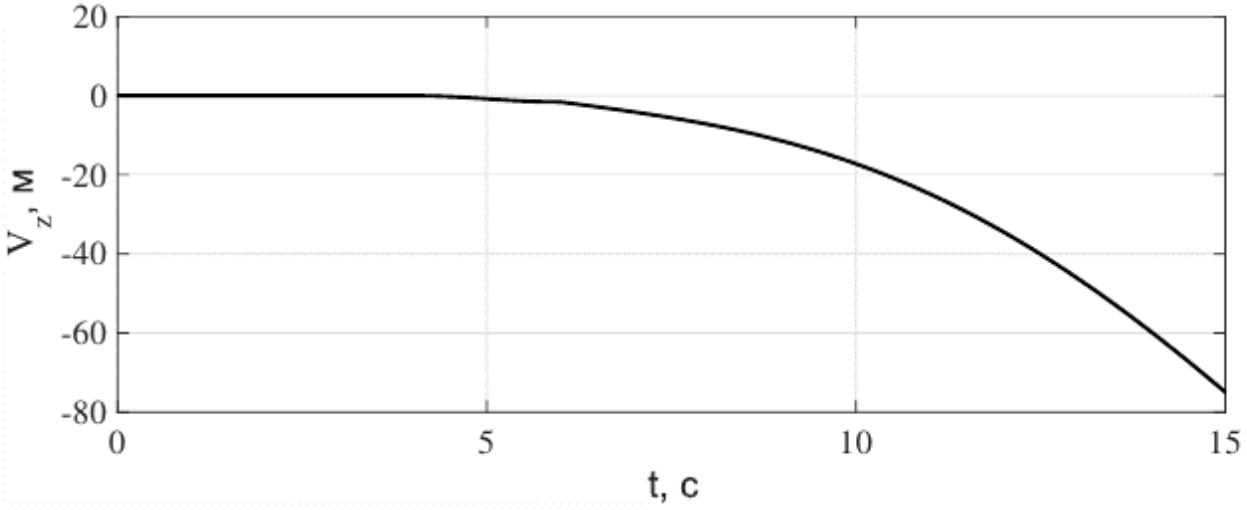


Рис. 3.7 – Аварія при зниженні: швидкість V_z

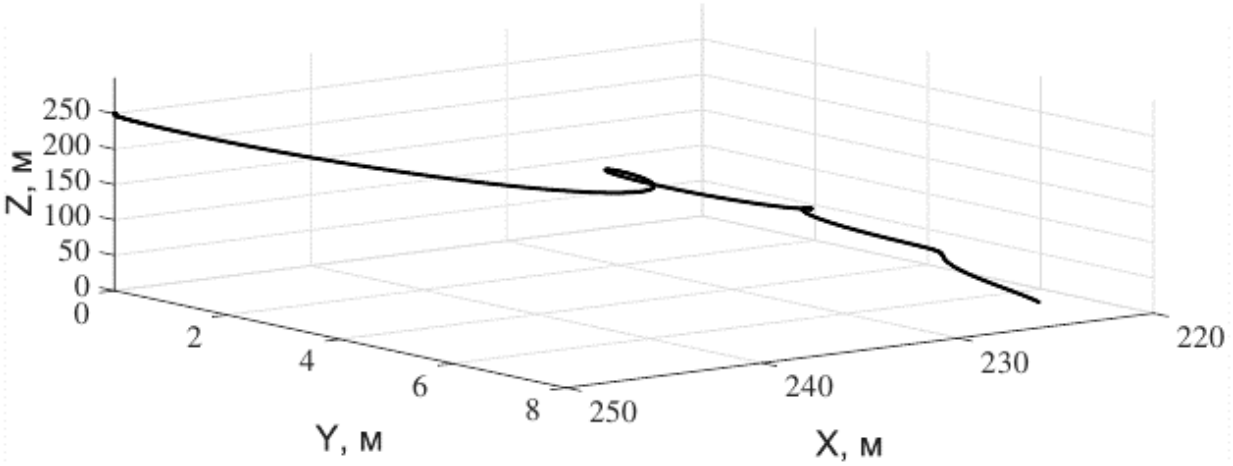


Рис. 3.8 – Аварія при зниженні: траєкторія переміщення апарата

За отриманими результатами, можна зробити такі висновки:

- Втрата тяги хоча б на одному гвинті квадрокоптера призводить до значної та слабко прогнозованої зміни його положення;
- При аварійній ситуації перед зіткненням із поверхнею землі квадрокоптер набирає значної швидкості. Доцільно вважати, що ця обставина спричиняє величезні збитки як для апарату, так і для корисного навантаження. Найімовірніше, що після такого падіння пристрій перестане бути придатним для подальшого використання (без ґрунтовної реконструкції).

3.2. Моделювання аварійних ситуацій із використанням рятувального алгоритму

3.2.1. Моделювання аварійних рухів квадрокоптера з алгоритмом «ручної» посадки

Розглянемо моделювання алгоритму, що передбачає автоматичне перемикання з нормального режиму керування в аварійний та подальше «ручне» керування висотою апарату для його посадки. Для цього наведемо докладну стратегію керування двигунами апарату відповідно до п. 2.2.

Як було зазначено раніше, наявність ознак несправності апарату може бути визначено польотним контролером з високою точністю лише за 0,5 секунди.

Нехай на 5 секунді польоту виникає аварійна ситуація повна відмова другого двигуна (Ω_2 = 0 об / с при t > 5 с). Алгоритм «ручної» посадки у цій ситуації передбачає таку послідовність дій:

1–й крок: Згідно з п. 2.2, в момент ідентифікації несправності управління апаратом перемикається на «ручний» аварійний режим, і навіть відбувається інформування оператора БПЛА через інтерфейс пульта управління. Відразу після виявлення дефектного двигуна (в даному прикладі при t = 5,5 с) примусово відключається двигун, що розташовується по діагоналі з гвинтом, що вийшов з ладу (Ω_4 = 0 об/с, див. рис. 3.11);

2–й крок: Спільно з діями 1 кроку (у момент відключення двигунів пошкодженої діагональної пари) збільшується тяга працюючих двигунів на другій діагоналі. Результуюче значення тяги пари гвинтів, що працює, має наближатися до сукупної тяги апарата для підтримки висіння (значення $\Omega_1 = \Omega_3$ = 400 об/с, див. рис. 3.11);

3–й крок: Управління тягою на двох гвинтах із зміненим нульовим положенням ручки керування (щодо необхідної тяги для висіння з урахуванням кроків 1 і 2) переходить до оператора;

4–й крок: На основі доступних засобів контролю оператор БПЛА забезпечує керування висотою апарату для досягнення безпечної посадки. У межах цього прикладу на інтервалі є [5,5; 11,7] проводиться зменшення вертикальної швидкості. У міру наближення апарата до поверхні землі знижується вертикальна швидкість до нуля.

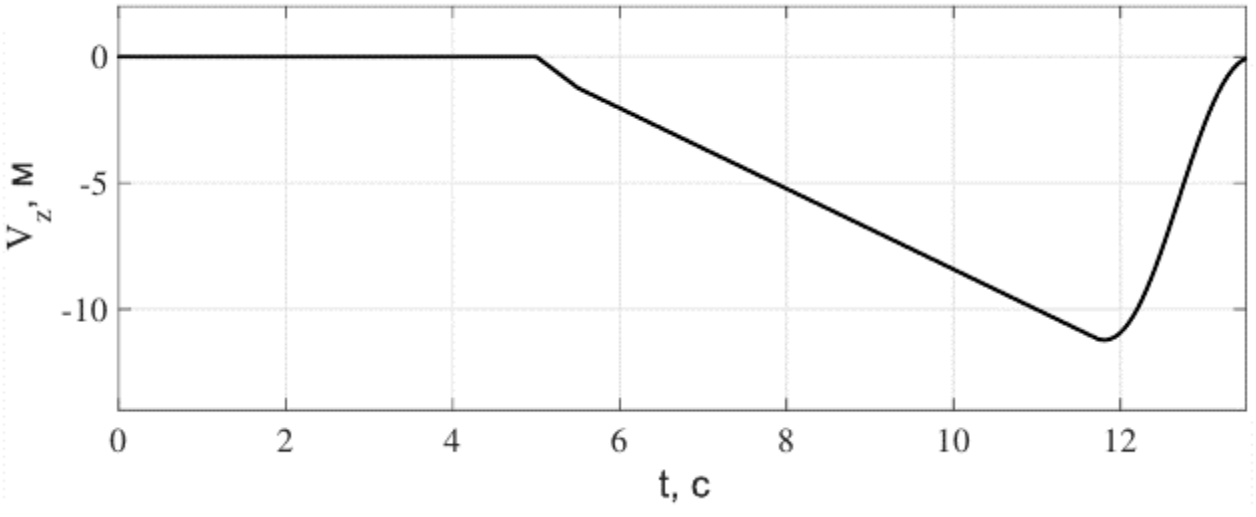


Рис. 3.9 – «Ручний» рятувальний алгоритм: швидкість V_z

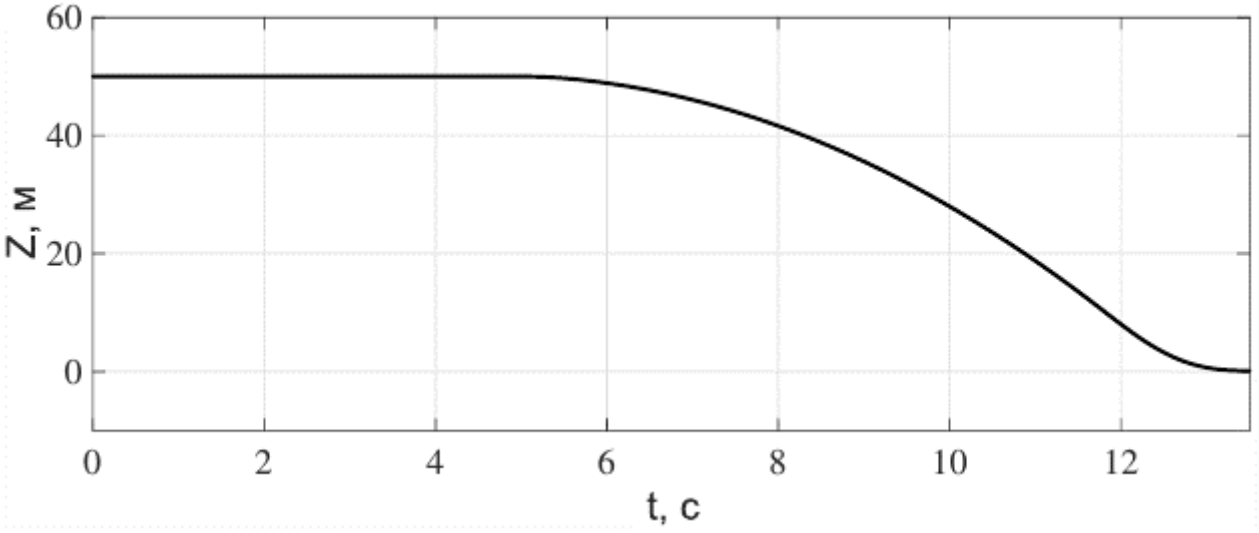
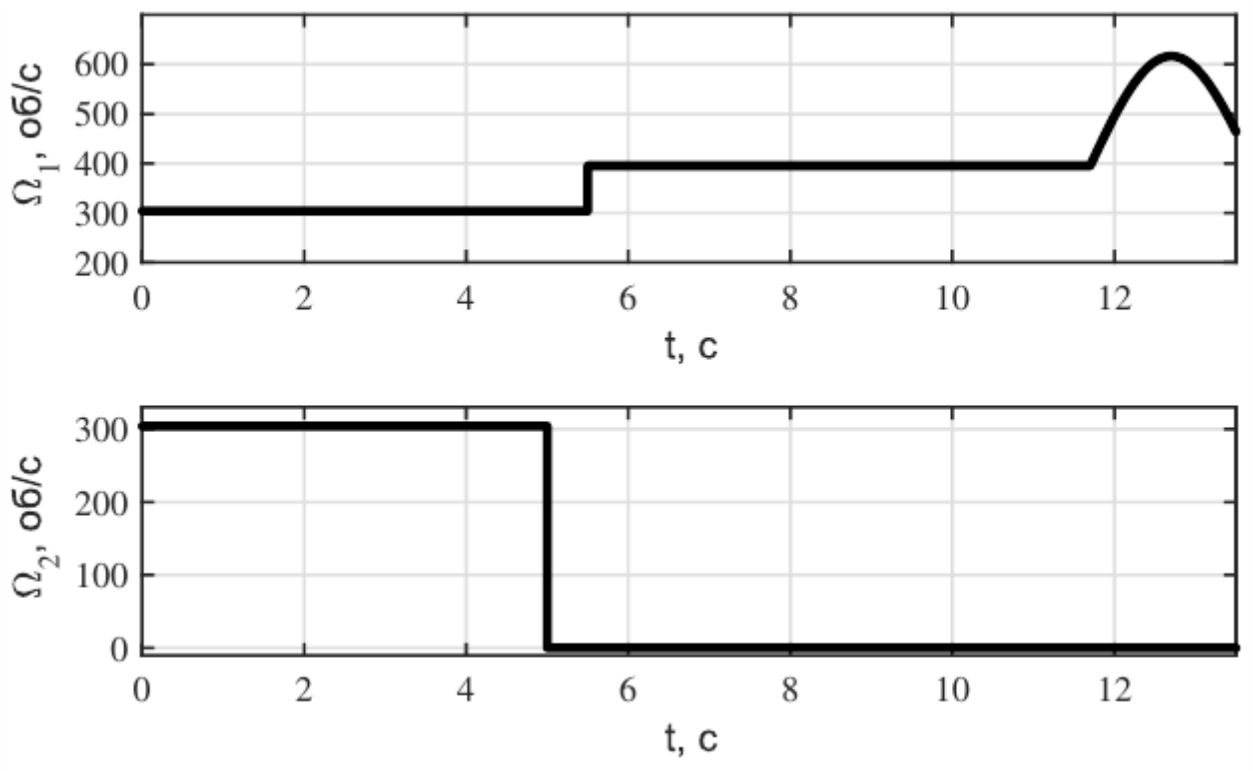


Рис. 3.10 – «Ручний» рятувальний алгоритм: висота Z

В результаті отримаємо наступну траєкторію переміщення квадрокоптера (див. рис. 3.12). Значення кутових швидкостей, швидкості V_z і висоти Z показані на графіках 3.11, 3.9 і 3.10 відповідно.



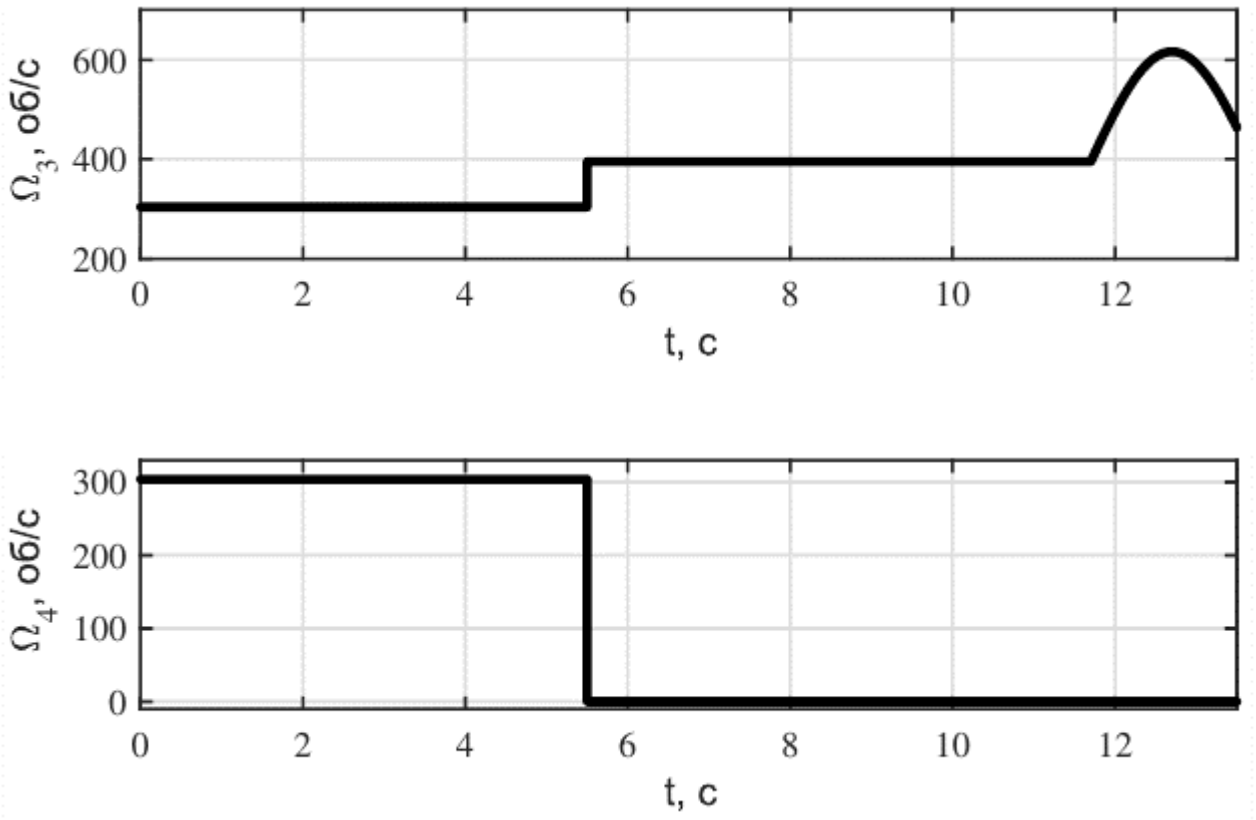


Рис. 3.11 – «Ручний» рятувальний алгоритм: кутові швидкості

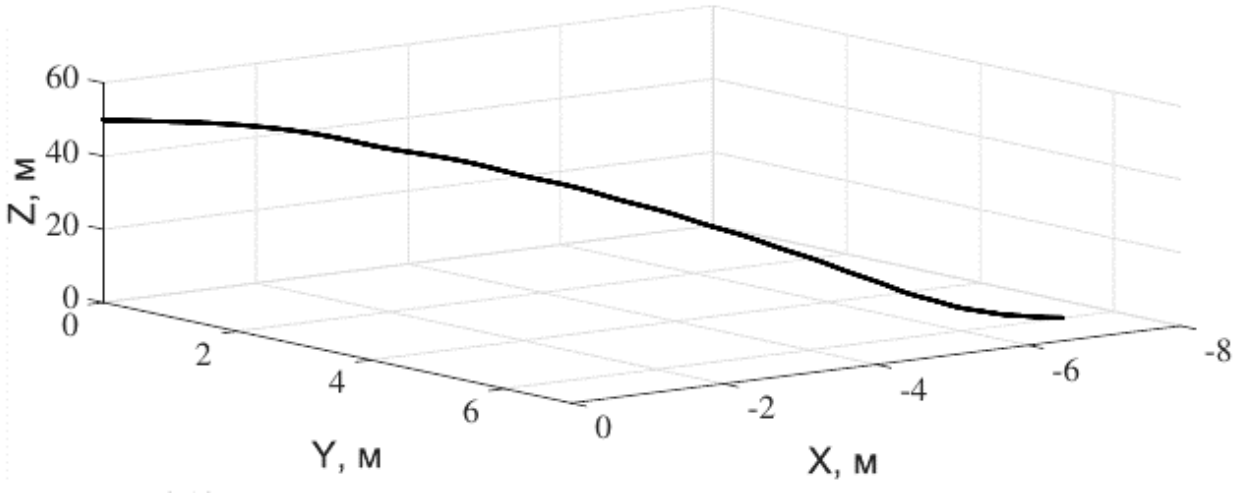


Рис. 3.12 – «Ручний» рятувальний алгоритм: траєкторія переміщення апарата

Ключовою умовою для безпечної посадки є гасіння швидкості приземлення апарату при наближенні до землі ($V_Z \to 0$ при $Z \to 0$). Діючи за вказаним алгоритмом, можна звести ризик пошкодження корисного навантаження апарата до мінімуму.

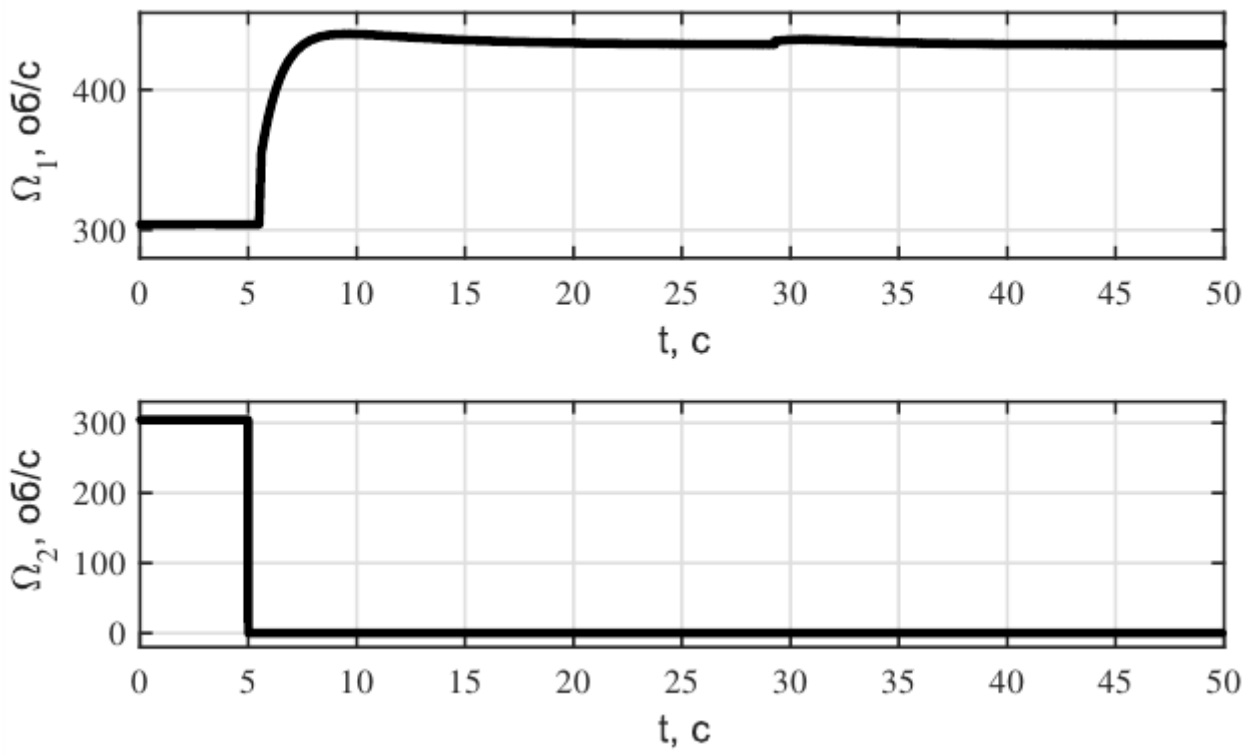
3.2.2. Моделювання аварійних рухів квадрокоптера з використанням ПІД контролера

Розглянемо автоматизований підхід до безпечної посадки квадрокоптера, що базується на міркуваннях з п. 2.3. Припустимо, що при переміщенні апарату в горизонтальній площині в момент часу $t^* = 5$ с виникає аварійна ситуація Ω_2 падає до нуля внаслідок пошкодження гвинта (див. рис. 3.13).

Відповідно до п. 2.2, за 0,5 с відбувається ідентифікація аварійної ситуації з наступним автоматичним перемиканням із нормального режиму управління в аварійний з використанням ПІД регулятора:

$[K_p, K_i, K_d] = [50, 8, 20]$; висота переходу на ділянку гальмування $z_0 = 5$ м.

Швидкість обертання робочих гвинтів контролюється двома ПІД–регуляторами відповідно до п. 2.3. Перший забезпечує задану вертикальну швидкість (–1 м/с) на інтервалі $t \in [5.5, 29.3]$, поки апарат знаходиться на висоті Z>5 м (див. рис. 3.14). Далі відбувається перемикання на ПІД–регулятор, що зводить до 0 швидкість зниження апарату при досягненні поверхні землі.



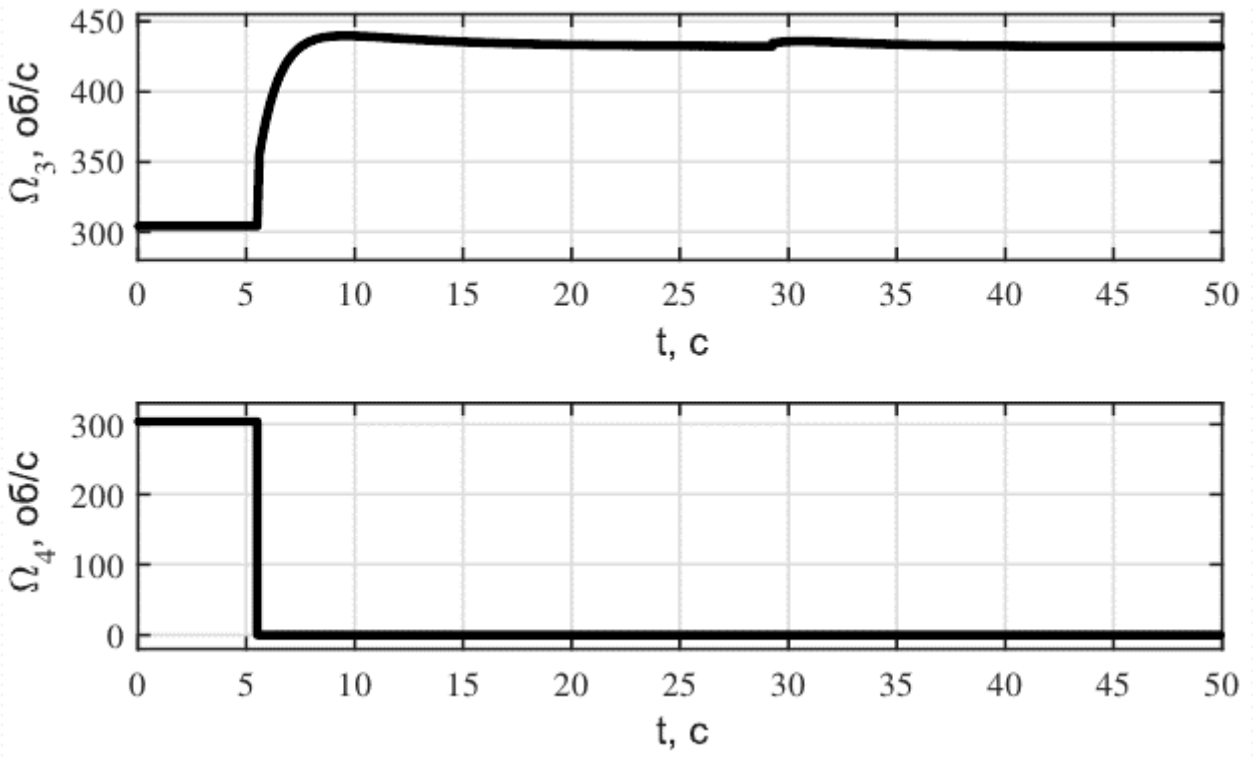


Рис. 3.13 – Автоматизований рятувальний алгоритм: кутові швидкості Ω_i

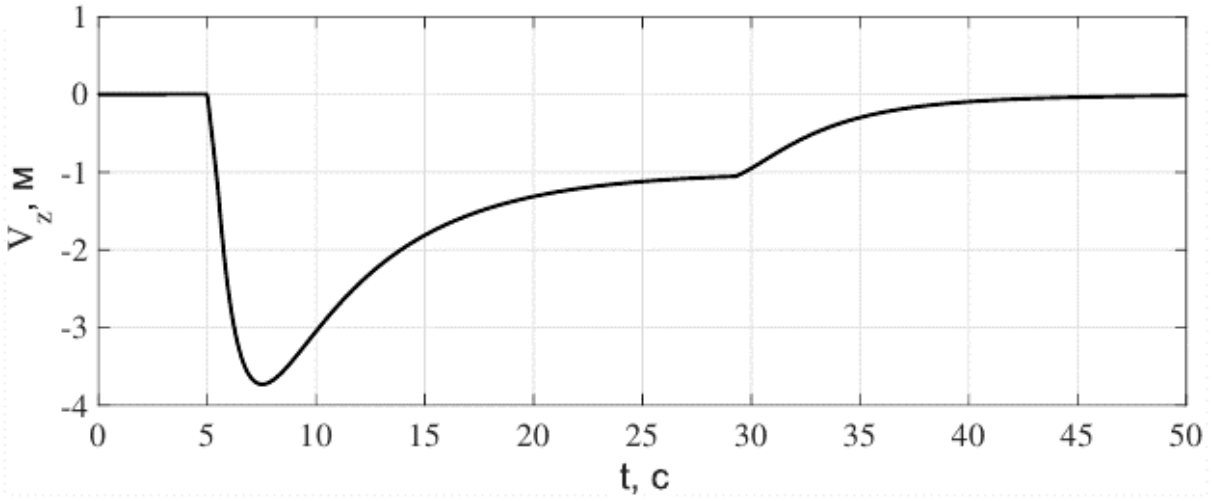


Рис. 3.14 – Автоматизований рятувальний алгоритм: швидкість V_z

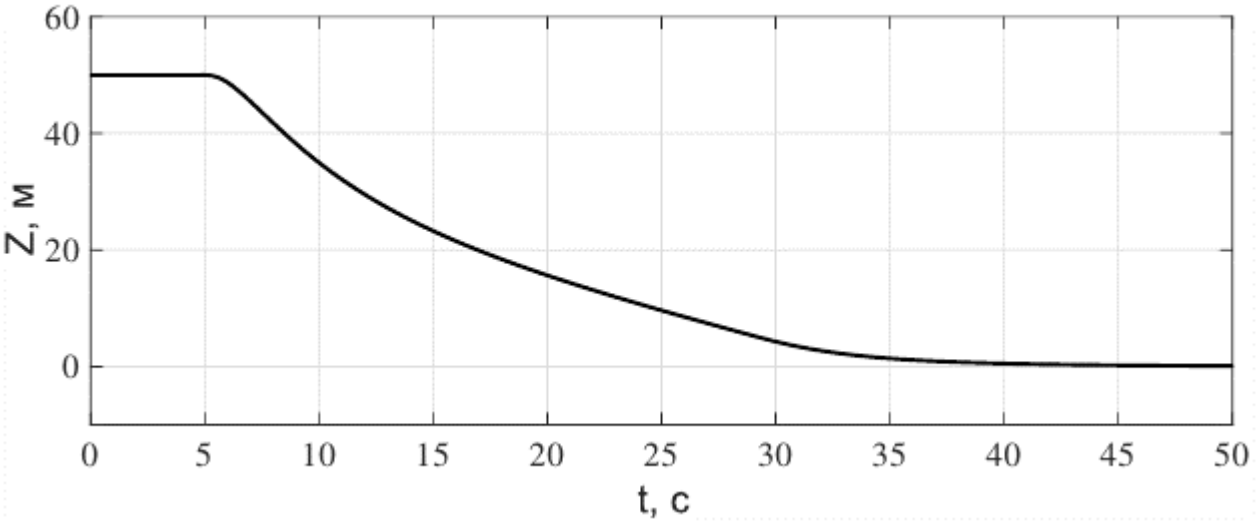


Рис. 3.15 – Автоматизований рятувальний алгоритм: висота Z

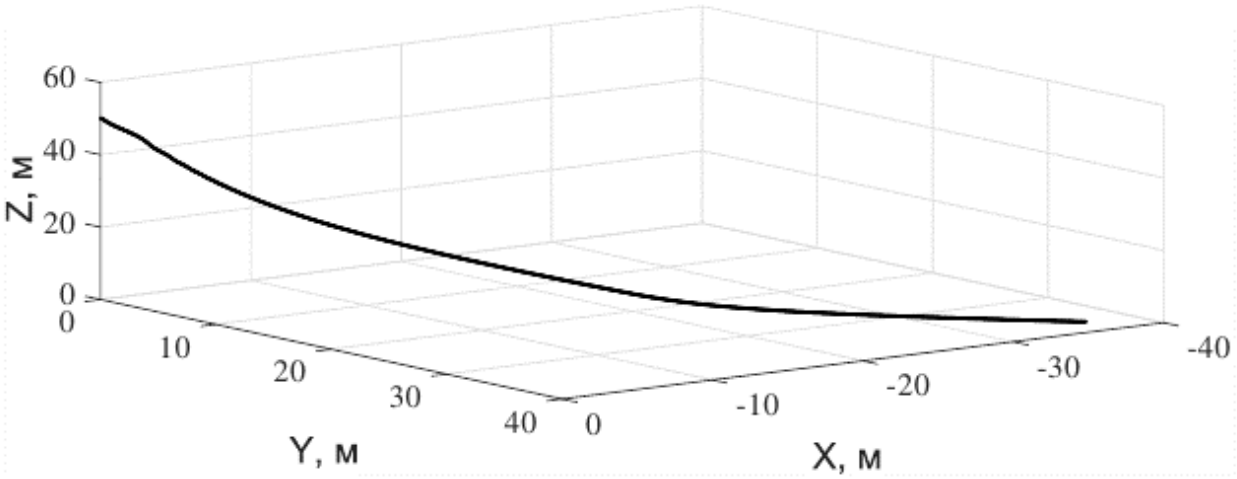


Рис. 3.16 – Автоматизований рятувальний алгоритм: траєкторія переміщення апарата

В результаті ми маємо безпечну посадку (див. рис. 3.15) з нульовою вертикальною швидкістю при Z = 0. При t = 50 с та Z = 0.09 м, маємо V_z = –0.01 м (див. рис. 3.14). Графік траєкторії переміщення апарату подано на рис. 3.16.

Слід зазначити, що робота алгоритму була б неможлива, якби не було враховано рекомендації щодо конфігурації квадрокоптера (див. табл. 1).

3.3. Експерименти щодо оцінки відстані приземлення БПЛА в аварійній ситуації

Цей пункт демонструє залежність траєкторії переміщення апарата від величини несправності одного з двигунів (роторів) апарату. Важливо розуміти,

що автоматичне управління, крім переваг від бездоганної точності виконання рятувальної методології, має також велику слабкість у вигляді неприпустимості відхилень від раніше заданої послідовності процесів. Тому подібний «аналіз у деталях» як аварійної ситуації, так і впровадження автоматичного методу порятунку, дозволяє з'ясувати залежність динаміки переміщення квадрокоптера від ступеня пошкодження гвинта (або його електричної частини) Ω_i^{em} та часу початку дії алгоритму безпечної посадки.

Для початку проведемо ряд моделювань аварійних ситуацій, взявши як початкове положення апарату (x_0, y_0, z_0) = (0, 0, 50). Розглянемо наступні значення кутової швидкості пошкодженого гвинта Ω_2:

$$\Omega_2^{em_i} = 0 + 3i, i = \overline{0, 100} \tag{3.1}$$

Значення $\Omega_2^{em_i}$ = 300 об/с відповідає слабкому пошкодженню пропелера, оскільки необхідне значення кутової швидкості кожного гвинта для підтримки стану спокою (у рамках даної моделі квадрокоптера з параметрами (1.5)) Ω_{hovering} = 304.1691 об/с.

Отже, отримаємо такі результати (рис. 3.17). Кожна крива відповідає траєкторії падіння апарата в 1–й аварійній ситуації. Для зручності аналізу отриманих даних залишимо точку приземлення апарату у кожному окремому випадку. Отримаємо наступний графік (рис. 3.18).

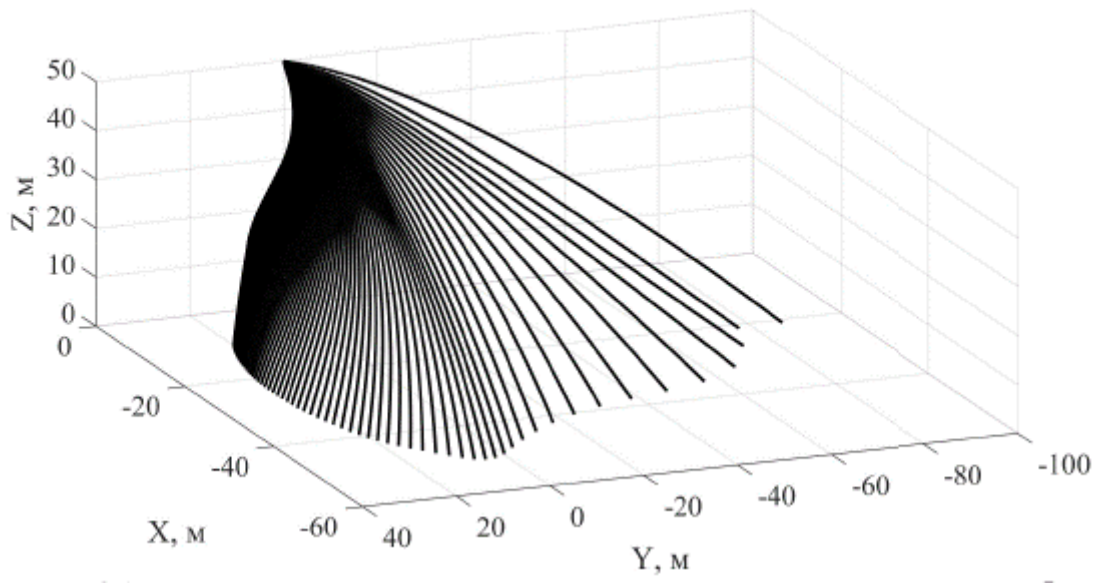


Рис. 3.17 – Множина траєкторій посадки апарата при різних значеннях Ω_2^{em}

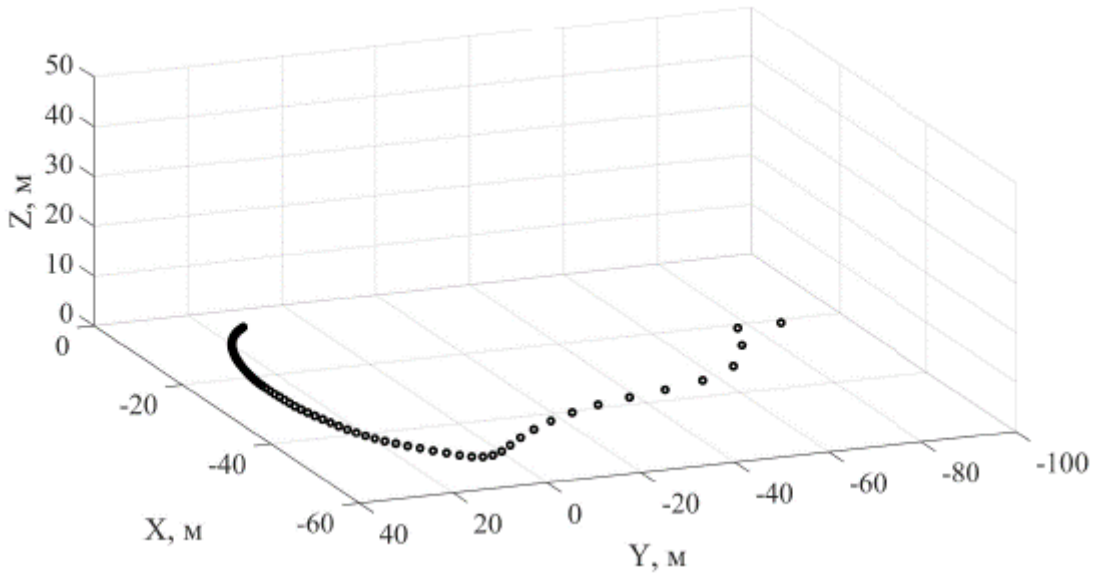


Рис. 3.18 – Множина точок приземлення апарата при різних значеннях Ω_2^{em}

Виходячи з отриманих результатів можна зробити висновок, що малі значення $\Omega_2^{em_i}$ не сильно впливають на відстань від точки приземлення апарата до початкового положення, в той час як між точками зі значеннями кутової швидкості, що відповідають слабкому пошкодженню апарата, зміна відстані помітно більше. Це можна пояснити тим, що за слабкого порушення працездатності гвинта квадрокоптер тривалий час не досягає критичного нахилу, що веде до безконтрольного падіння. Іншими словами, чим менше пошкоджений гвинт, тим далі апарат пролетить перш ніж впаде на землю.

Розглянемо вплив алгоритму відмовостійкого приземлення квадрокоптера (п. 3.2.2) у тих самих умовах моделювання. Для цього введемо додатковий параметр, що вимірюється в секундах – параметр $\mathrm{t}_{\mathrm{reaction}} = 0{,}5 + 0{,}1;\ \mathrm{i} = \overline{0{,}5}$.

Ця змінна несе такий практичний зміст: час виявлення несправності системою з наступним перемиканням в аварійний режим. Випадки з відкладеною реакцією ($t_{reaction} > 5$ с) становлять більший інтерес, ніж ситуації з ($t_{reaction} < 5$ с), оскільки раніше вже була продемонстрована ефективність стійкості до відхилення алгоритму з затримкою в 0,5 с [8] і, крім для більш швидкої реакції БПЛА доцільно очікувати підвищення вимог до конфігурації самого апарату, що не

доцільно в межах даної математичної моделі. Здійснення пізнішого запуску рятувального алгоритму, всупереч сказаному вище, не залежить від пристрою безпосередньо – можна програмно визначити необхідний момент часу.

Важливо помітити, що у цій роботі метод аварійної посадки не повністю обнулює швидкості пошкодженого та симетричного йому гвинтів, а зберігає кутові швидкості обох пропелерів однаковими. В даному випадку на час дії відмовостійкого алгоритму $\Omega_4 = \Omega_2^{em_i}$, що усуває нахил апарату і дозволяє зберегти деяку тягу, полегшуючи роботу по посадці на повністю справної парі гвинтів.

Таким чином, можна дослідити залежність переміщення квадрокоптера від часу початку дії рятувального алгоритму після виникнення аварії (рис. 3.19).

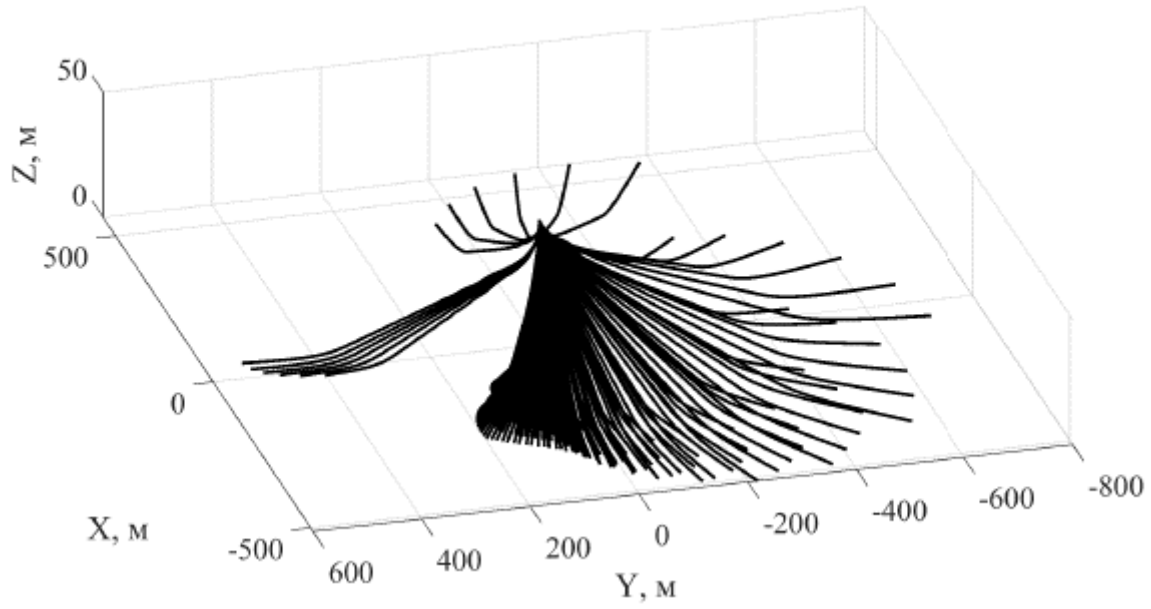


Рис. 3.19 – Множина траєкторій посадки апарата при різних $\Omega_2^{em}, t_{reaction}$

Як і в попередньому експерименті, для зручності аналізу отриманих даних залишимо точку приземлення апарату у кожному окремому випадку. Отримаємо наступний графік (рис. 3.20).

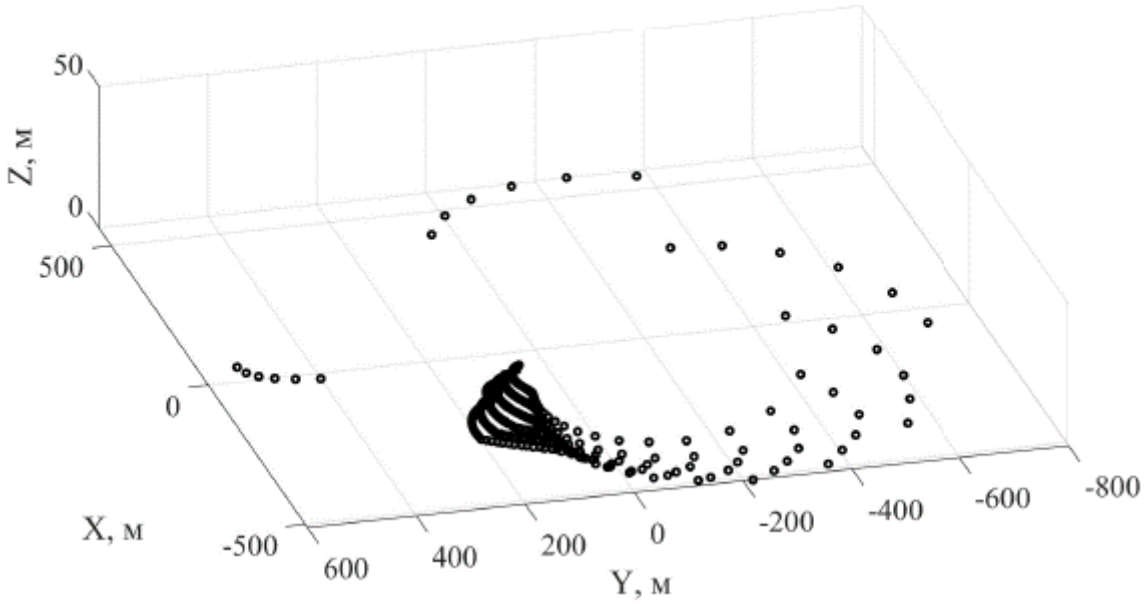


Рис. 3.20 – Множина точок приземлення апарата при різних $\Omega_2^{em}, t_{reaction}$

На графіку можна побачити, що під час роботи відмовостійкого методу апарат летить на більшу відстань при більшому значенні $\Omega_2^{em_i}$. Проте, порівняно з попереднім експериментом, запровадження параметра $t_{reaction}$ ускладнило візуальний аналіз. У зв'язку з цим згрупуємо отриману множину точок за часом початку роботи алгоритму посадки після виникнення аварії $t_{reaction}$ (див. рис. 3.21).

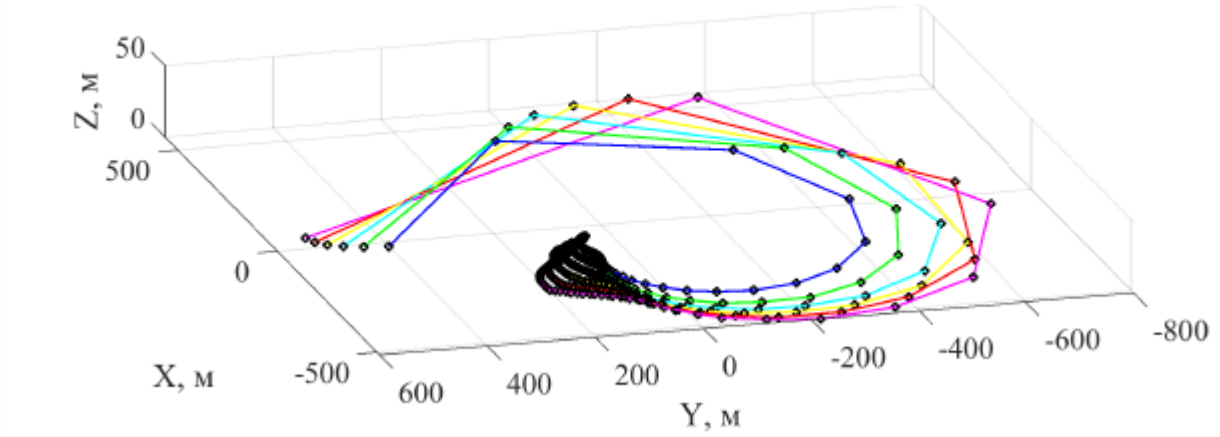


Рис. 3.21 – Множина точок приземлення апарата при різних $\Omega_2^{em}, t_{reaction}$ (групування за $t_{reaction}$)

Виходячи з отриманих результатів можна дійти невтішного висновку, що час початку дії рятувального алгоритму певною мірою впливає на напрям посадки квадрокоптера безліч точок приземлення утворює «витки», у яких кожна точка відповідає певному часу перемикання на рятувальний алгоритм.

Таким чином, цей підхід можна використовувати в наступних напрямках:

1. При виникненні аварійної ситуації можна визначити область падіння апарату або, у разі підключення відмовостійкого алгоритму, визначити приблизну дальність і напрямок посадки, що дозволяє уникнути переміщення в зону, що не рекомендується для посадки, варіюючи час запуску програми після втрати тяги на одному з гвинтів.

2. Більш детальне вивчення динаміки квадрокоптера може дозволити на практиці здійснити контроль в автоматичному режимі над дальністю та напрямом польоту несправного апарату.

Іншими словами, отримані дані потенційно можуть бути використані для вдосконалення роботи алгоритмів посадки [7; 8], що описані у п. 3.2.

3.4. Висновок з результатів моделювання

Цей розділ демонструє моделювання лише окремих випадків аварійних ситуацій. Однак важливо зауважити, що позаштатні ситуації з несправностями будь-якого роду фактично тягнуть за собою однакові наслідки: втрата контролю над квадрокоптером, неминучий переворот апарату з подальшим падінням.

Справжнім «частковим» випадком є «ручний» порятунок апарату, оскільки результуючі дії здійснюються в рамках певних обставин і можуть бути певною мірою узагальнені.

Автоматизований алгоритм посадки позбавлений цього недоліку завдяки повному програмному контролю за будь–якої аварії, наслідки котрої дозволяють керувати двома діагональними гвинтами. Проте, це також є і проблемою даного підходу: фіксовані заздалегідь параметри (бажана швидкість зниження, висота переходу на «знижуючий» швидкість приземлення режим, час початку роботи) найчастіше визначають як час посадки, так і відстань, яку апарат пролетить перед досягнення поверхні землі. Дані обставини не дозволяють отримати відповідну реакцію алгоритму на вимоги або зміни зовнішнього середовища (обмежена область посадки, остаточна величина заряду акумуляторів).

Вирішувати зазначені проблеми автоматизованого ПЗ пропонується ускладненням рятувального алгоритму, проте ця рекомендація не у змозі вирішити весь спектр можливих ситуацій без використання прорахунку подальшої траєкторії апарату у кожний момент часу.

ВИСНОВКИ

Основні результати роботи:

1. Описана математична модель квадрокоптера та визначені параметри, що відповідають реальній конфігурації апарата (п. 1.1 та п. 1.2);
2. Наведена класифікація аварійних ситуацій;
3. З точки зору математичної моделі визначена аварійна ситуація, способи порятунку та оцінка прогнозуємої області падіння (приземлення) квадрокоптера з врахуванням визначення можливих переміщень апарата (п. 2.1 – 2.4);
4. Проведено чисельне моделювання різноманітних аварійних ситуацій (п. 3.1);
5. Здійснено впровадження методології рятувального приземлення у математичну модель квадрокоптера (п. 3.2);
6. Здійснені чисельні експерименти, що демонструють можливості оцінки очікуваної області падіння (п. 3.3);
7. Здійснено аналіз отриманих результатів (п.3.4);
8. Для реалізації поставлених задач було розроблено ПЗ на основі програмного пакета MATLAB R2019b з доповненням Simulink v.10.0 (див. додатки А та Б).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Beji L., Abichou A. Trajectory and Tracking of a Mini–Rotorcraft // Proceedings of the IEEE Int. Conf. on Robotics and Automation. 2005. – P. 2618–2623.

2. Bresciani T. Modeling, identification and control of a quadrotor helicopter : Master's thesis / T. Bresciani ; Lund University. — Sweden, Lund, 2008. — 184 pp.

3. Sklyarov A. A., Sklyarov S. A. Sinergeticheskij podkhod k upravleniyu bespilotnym letatel'nym apparatom v srede s vneshnimi vozmushheniyami [Synergistic approach to the quadrocopter control in environment with external perturbations]. News of SFedU. Technical science, 2012, no. 8, pp. 159–170.

4. Корченко А. Г. Обобщённая классификация беспилотных летательных аппаратов / А. Г. Корченко, О. С. Ильяш // Збірник наукових праць Харківського університету Повітряних сил. – 2012. – Вип. 4. – С. 27–36. – Режим доступу: http://nbuv.gov.ua/UJRN/ZKhUPS_2012_4_9

5. Weibel R., Hansman J. Safety Considerations for Operation of Unmanned Aerial Vehicles in the National Airspace System // MIT ICAT–2005–1, 2005. [online] http://hdl.handle.net/1721.1/30364

6. Maddalon J.M., Hayhurst K.J., Koppen D.M., et al. Perspectives on Unmanned Aircraft Classification for Civil Airworthiness Standards // National Aeronaut. Space Administrat., Langley Res. Center, Hampton, Virginia, 2013. [online] https://shemesh.larc.nasa.gov/people/jmm/NASA–TM–2013–217969.pdf

7. Baranov O. V., Smirnov N. V., Smirnova T. E., Zholobov Ye. V. Design of Fail–Safe Quadrocopter Configuration // Intelligent Distributed Computing XIII (IDC 2019) / Ed. by Igor Kotenko, Vasily Desnitsky, Costin Badica, Didier El Baz, Mirjana Ivanovic. — Studies in Computational Intelligence. — Germany : Springer Nature, 2020. — jan. — P. 13–22. DOI: 10.1007/978–3–030–32258–8_2.

8. Baranov O. V., Smirnov N. V., Smirnova T. E., Zholobov Ye. V. Design of a quadrocopter with PID–controlled fail–safe algorithm // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. — 2020. — jul. — Vol. 11, no. 2. — P. 23–33. DOI: 10.22667/JOWUA.2020.06.30.023.

9. Zulu A., John S. A Review of Control Algorithms for Autonomous Quadrotors //Syst. Control. [online] http://arxiv.org/abs/1602.02622v1

10. Ranjbaran M., Khorasani K. Fault recovery of an underactuated quadrotor aerial vehicle // 49th IEEE Conf. Decision Control (CDC–2010). Atlanta, 2010.P. 4385–4392.

11. Freddi A., Lanzon A., Longhi S. A feedback linearization approach to fault tolerance in quadrotor vehicles // IFAC World Congr. 2011. V. 44. No. 1. P. 5413–5418.

12. Scaramuzza D., Achtelik M., Doitsidis L., et al. Vision–controlled micro flying robots: from system design to autonomous navigation and mapping in gps–denied environments. [online] http://robotics.ethz.ch/scaramuzza/DavideScaramuzzafiles/publications/pdf/IEEERAMsubmitted.pdf

13. Marks A., Whidborne J.F., Yamamoto I. Control allocation for fault tolerant control of a VTOL octorotor // Proc. UKACC Int. Conf. Control. Cardiff(UK), 2012. P. 357–362.

14. Izadi H.A., Zhang Y., Gordon B.W. Fault tolerant model predictive control of quad–rotor helicopters with actuator fault estimation // IFAC World Congr. 2011. V. 18. No. 1. P. 6343–6348.

15. Chamseddine A., Zhang Y., Rabbath C.A., et al. Flatness–based trajectory planning/replanning for a quadrotor unmanned aerial vehicle // IEEE Trans. Aerospace Electron. Syst. 2012. V. 48. No. 4. P. 2832–2848.

16. Zhang Y., Chamseddine A., Rabbath C., et al. Development of advanced FDD and FTC, techniques with application to an unmanned quadrotor helicopter testbed //J. Franklin Institut. 2013. V. 350. No. 9. P. 2396–2422.

17. Salazar–Cruz S., Kendoul F., Lozano R., et al. Real–Time Control of a Small–Scale Helicopter Having Three Rotors // Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst. Beijing (China), 2006. P. 2924–2929.

18. Escareo J., Sanchez A., Garcia O., et al. Triple tilting rotor mini–UAV: Modeling and embedded control of the attitude // Proc. Amer. Control Conf. 2008. P. 3476–3481.

19. Kataoka Y., Sekiguchi K., Sampei M. Nonlinear Control and Model Analysis of Trirotor UAV Model // Proc. 18 Int. Federat. Autom. Control World Congr. 2011. V. 44. No. 1. P. 10391–10396.

20. Ulrich E.R., Humbert J.S., Pines D.J. Pitch and Heave Control of Robotic Samara Micro Air Vehicles // J. Aircraft. 2010. V. 47. No. 4. P. 1290–1299.

21. Лазарєв Ю. Ф. Моделювання динамічних систем у Matlab. Електронний навчальний посібник. – Київ: НТУУ "КПІ", 2011. – 421 с.

ДОДАТКИ

Додаток А. Структура математичної моделі в Simulink (MATLAB).

Математична модель квадрокоптера складається з двох головних блоків:

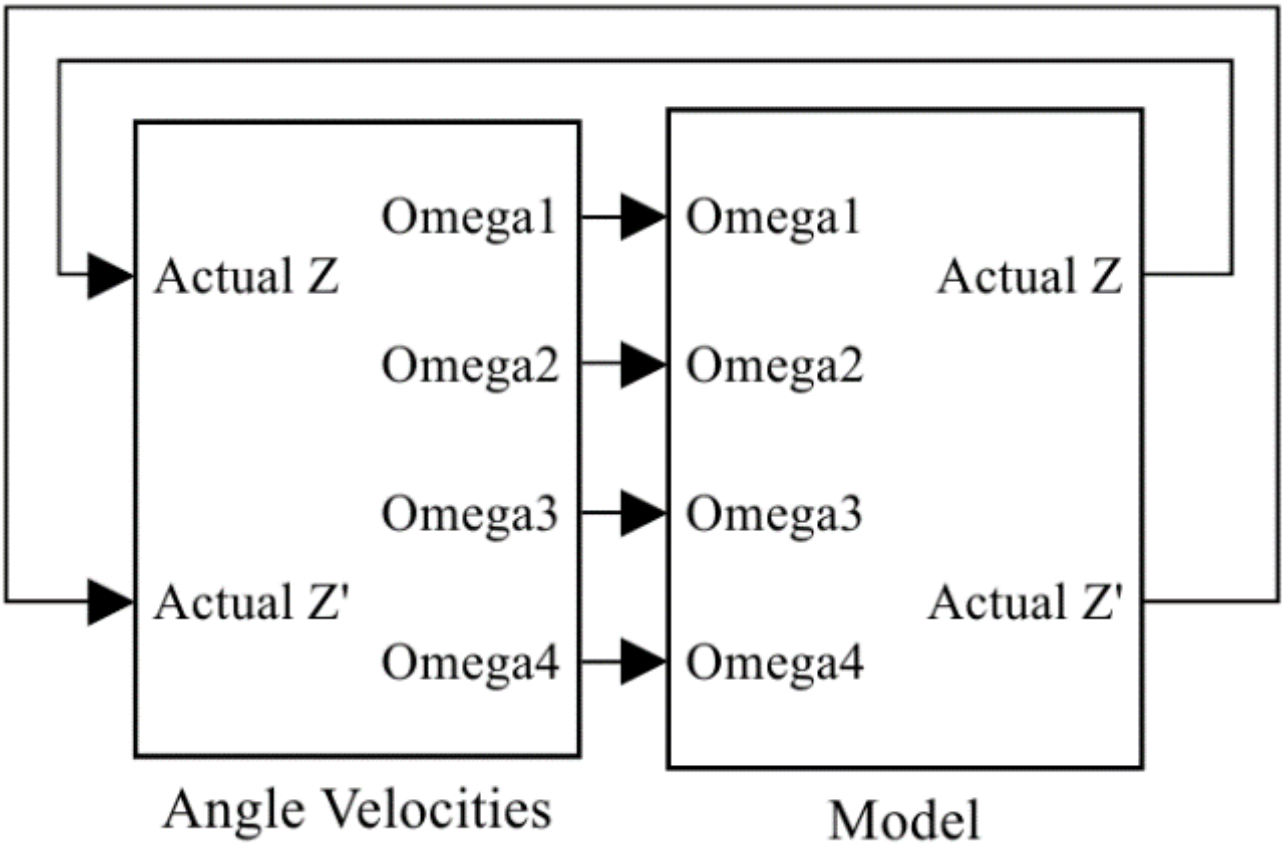


Рис. А.1 – Структура моделі в Simulink

А.1 Структура блока Angle Velocities

Внутрішня структура блока Angle Velocities розбита на кілька частин:

1. Симуляція аварії без використання рятувального алгоритма (червона область);

2. Симуляція аварії із застосуванням відмовостійкого алгоритма (зелена область);

3. Компоненти для передавання сигналів (бірюзова область).

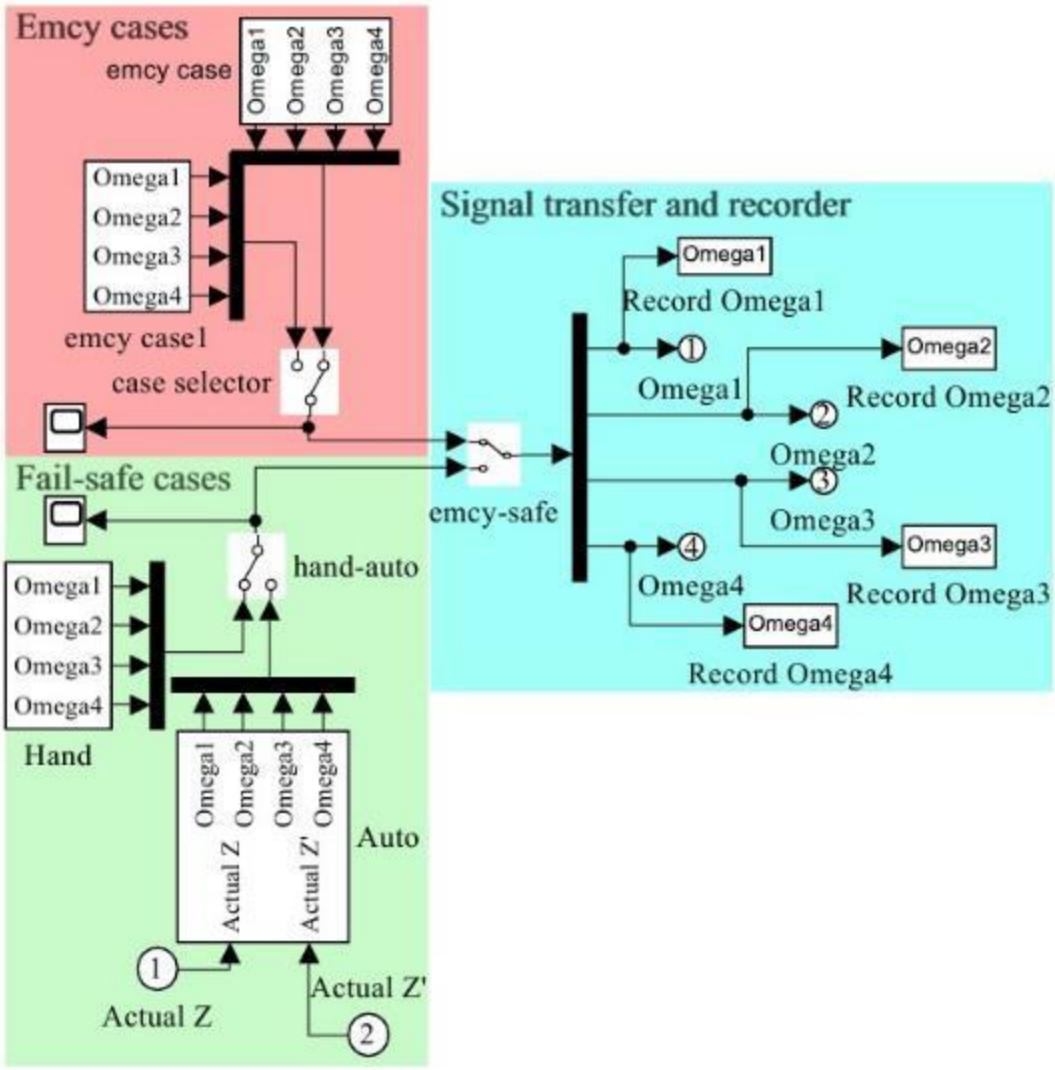


Рис. А.2 – Структура блоку Angle Velocities

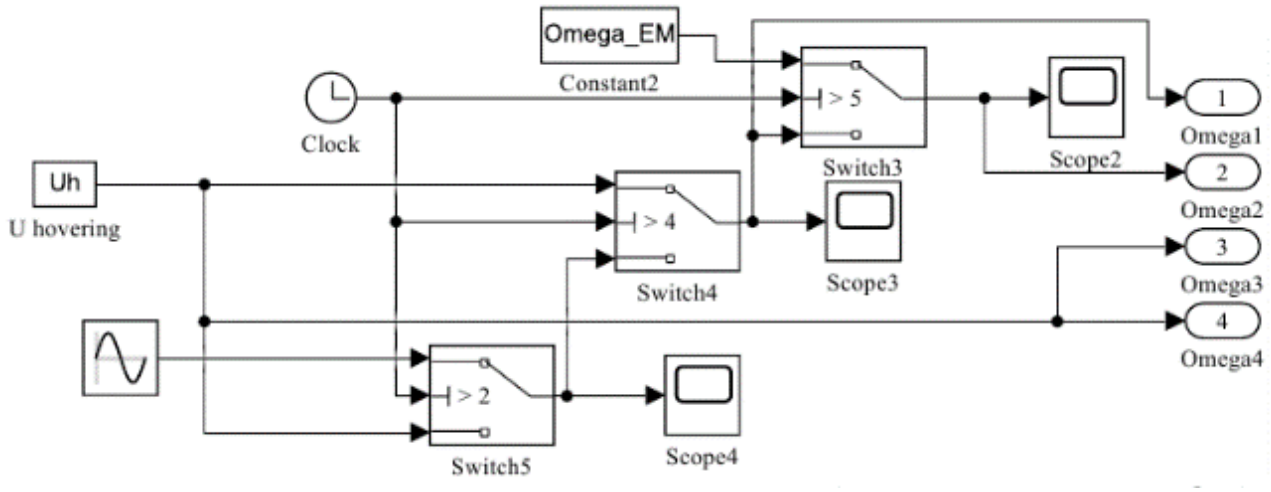


Рис. А.3 – Структура внутрішньої системи emcy case
(випадок різних значень Ω^{em})

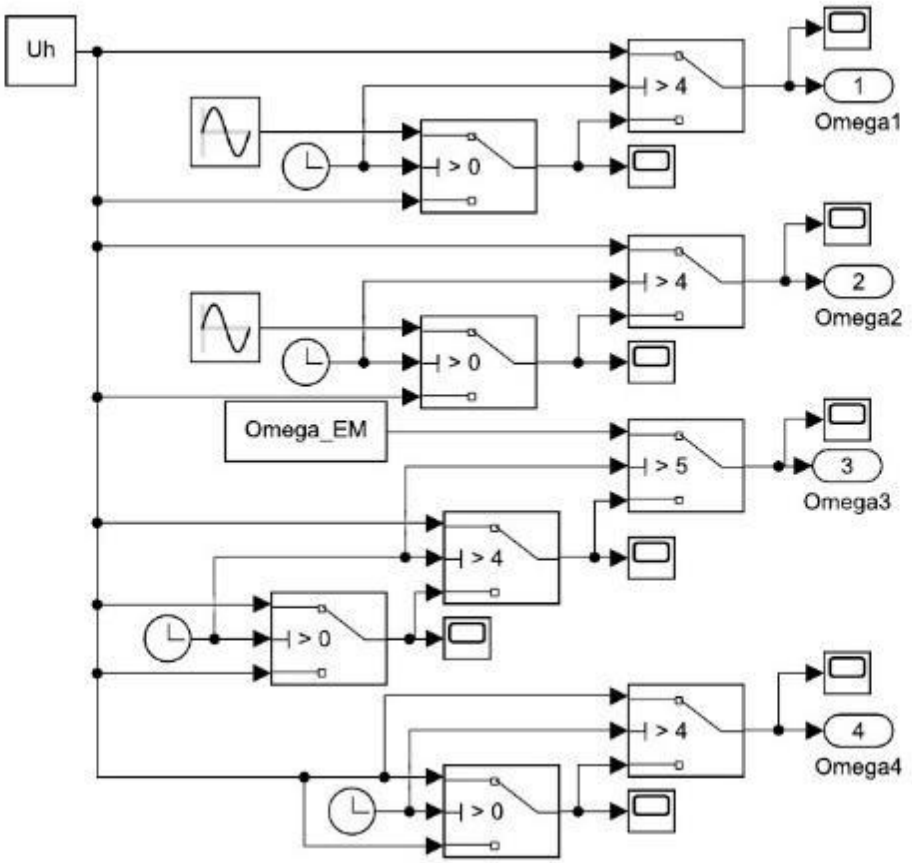


Рис. А.4 – Структура внутрішньої системи emcy case 1(випадок відмови гвинта при горизонтальному русі)

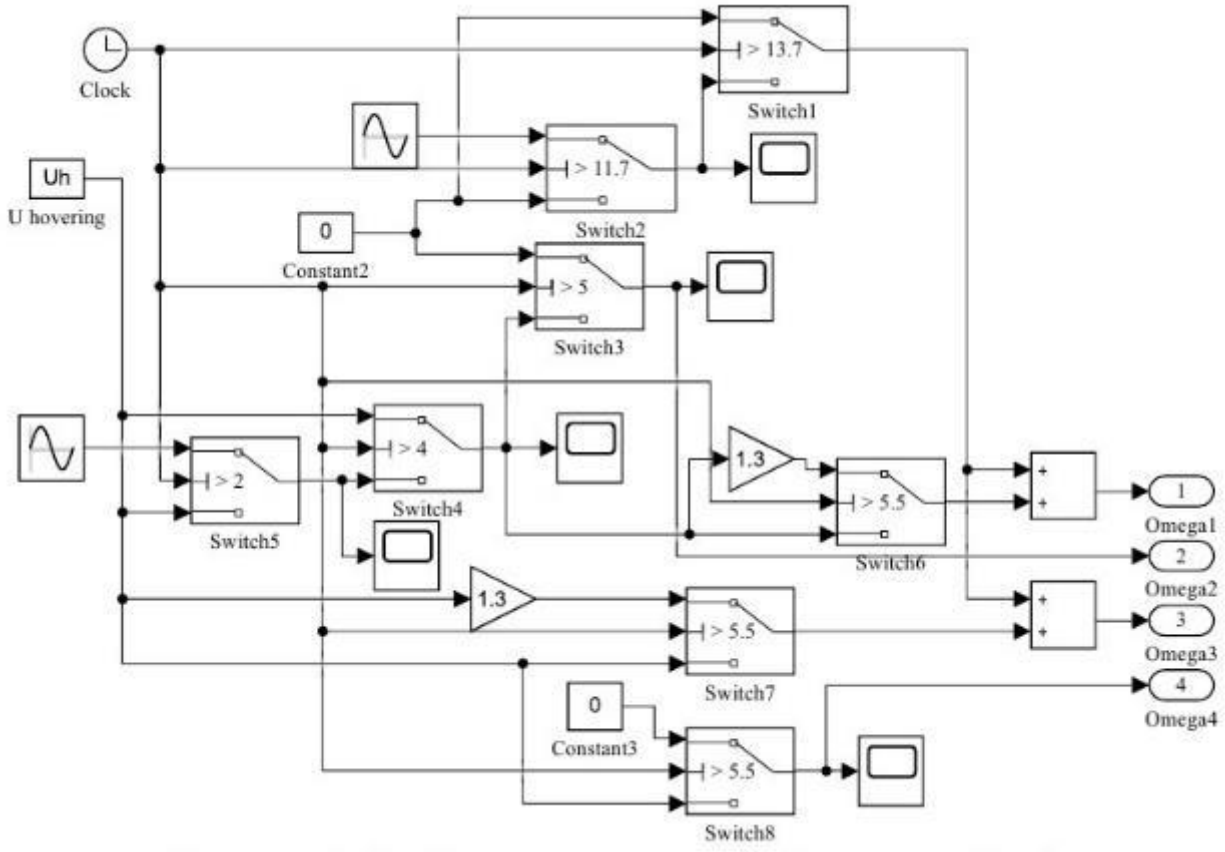


Рис. А.5 – Структура внутрішньої системи Hand

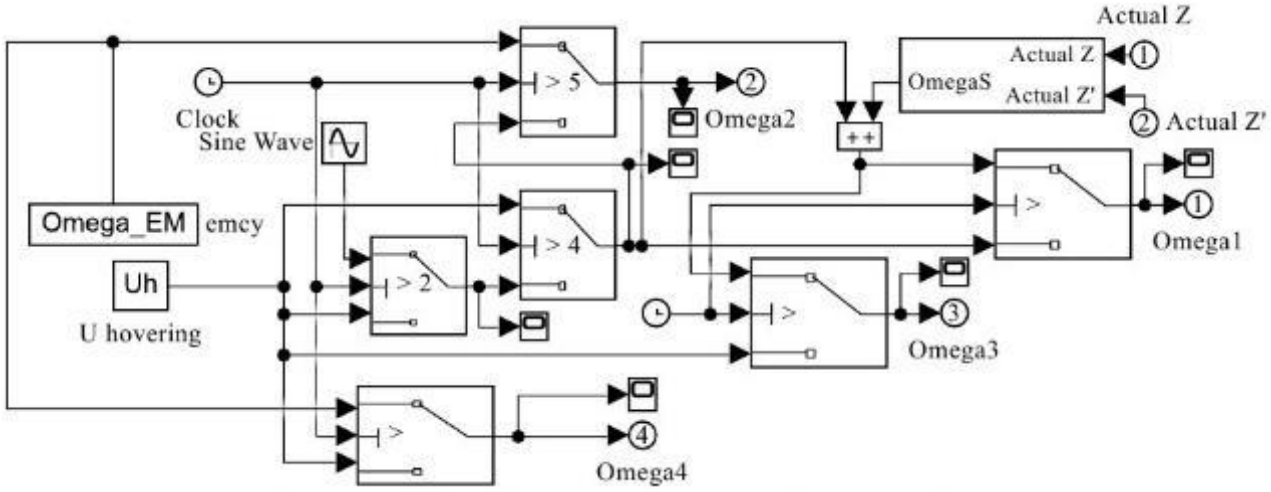


Рис. А.6 – Структура внутрішньої системи Auto

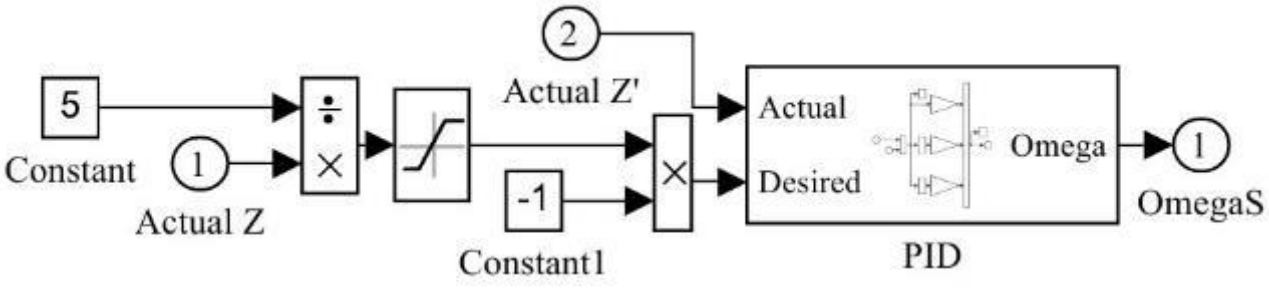


Рис. А.7 – Структура системи ПІД контролера в Auto

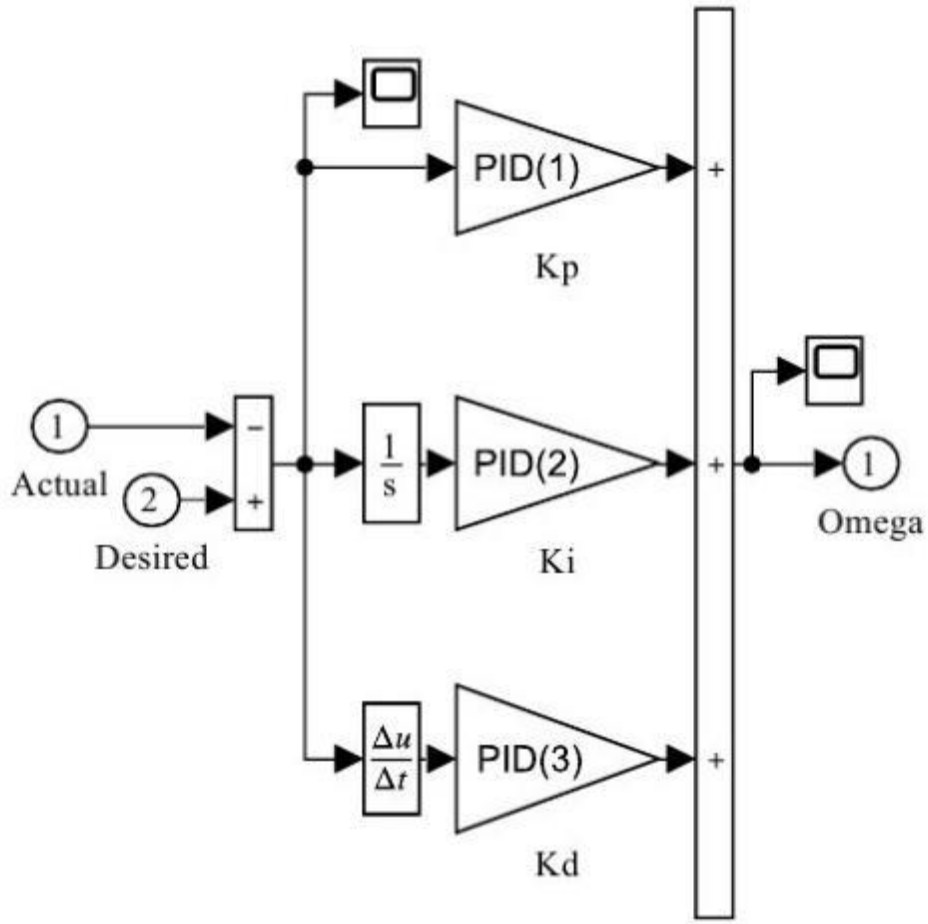


Рис. А.8 – Структура системи ПІД контролера в Auto

А.2. Структура блока Model

Блок Model містить математичну модель квадрокоптера та блоки введення/виведення даних.

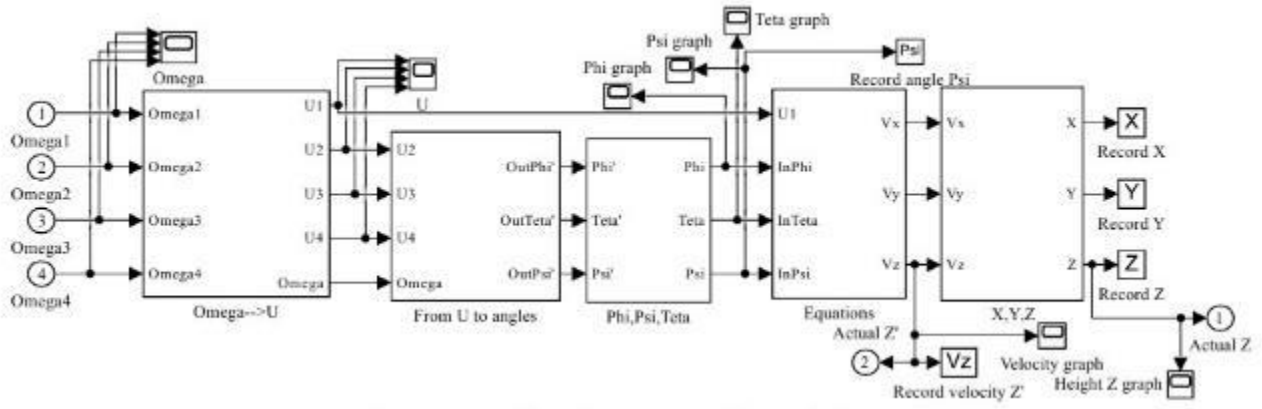


Рис. А.9 – Структура блоку Model

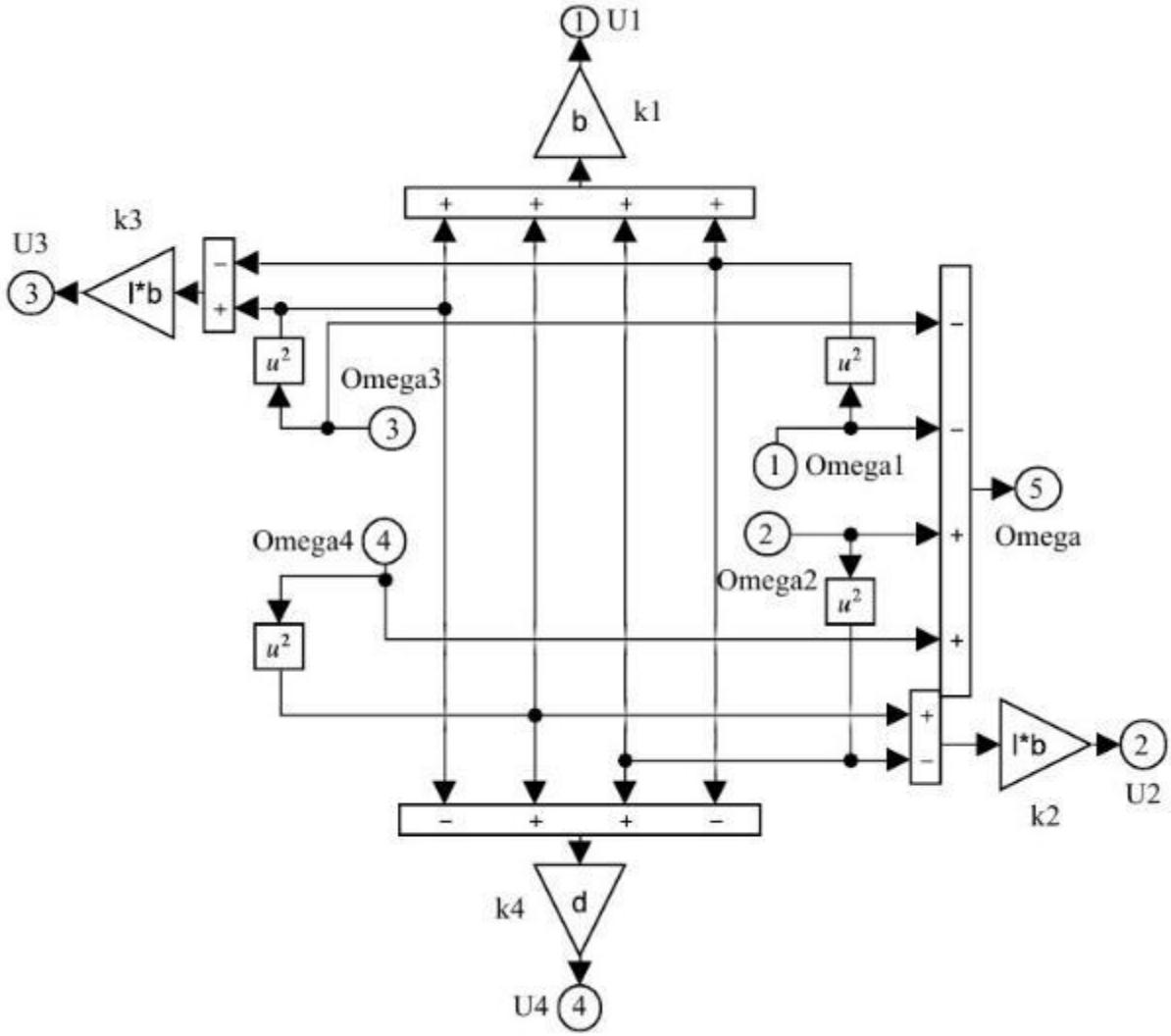


Рис. А.10 – Структура блоку Omega → U. Реалізація формули 1.3

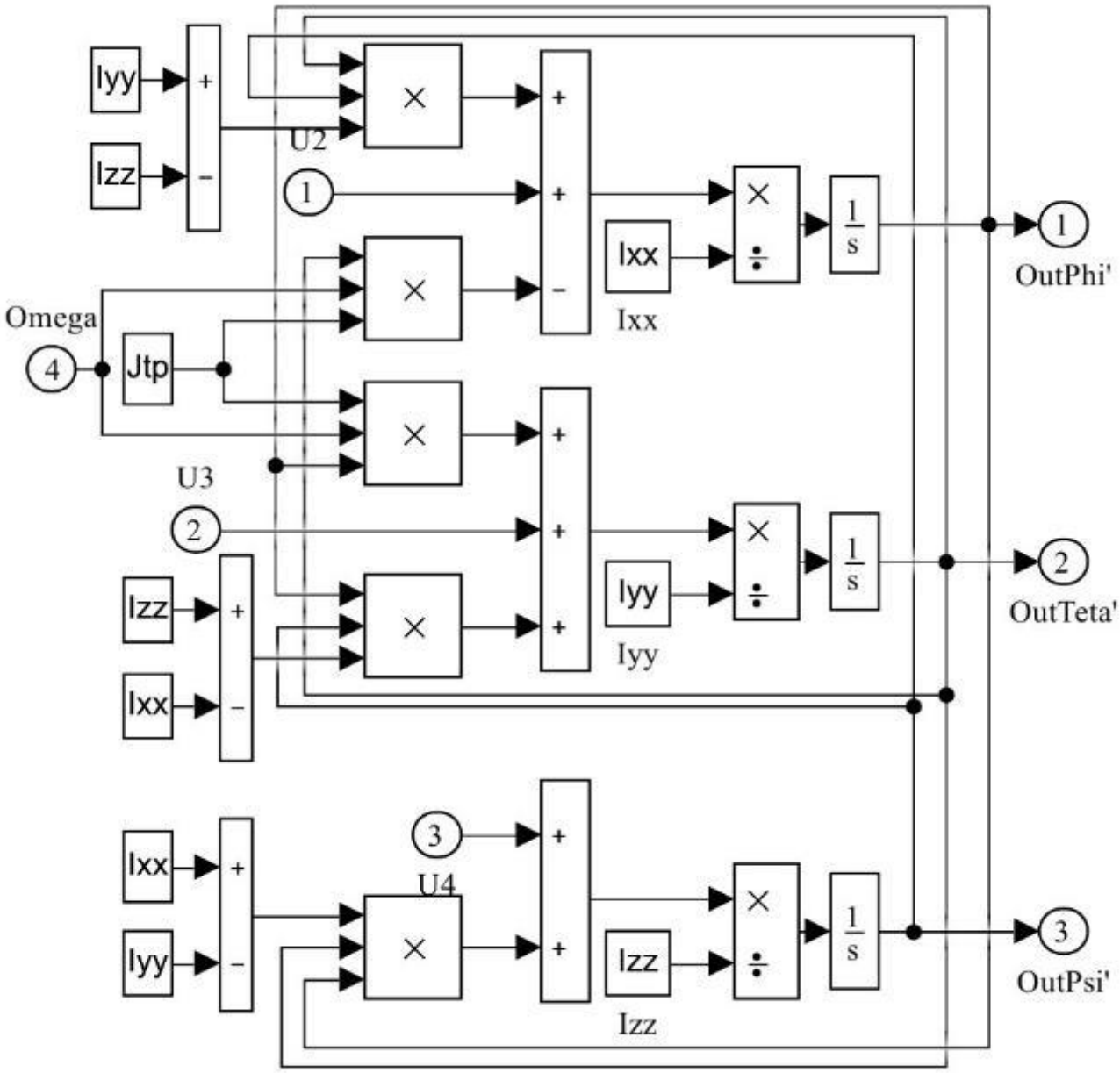


Рис. А.11 – Структура блоку From U to angles

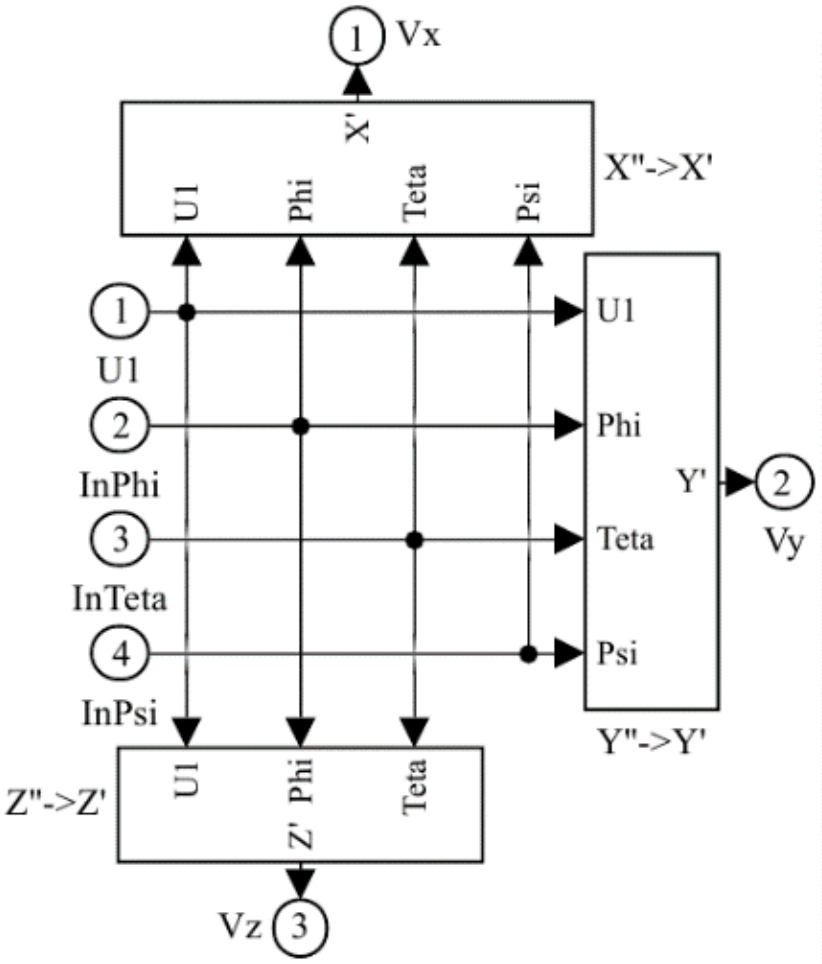


Рис. А.12 – Структура блока Equations

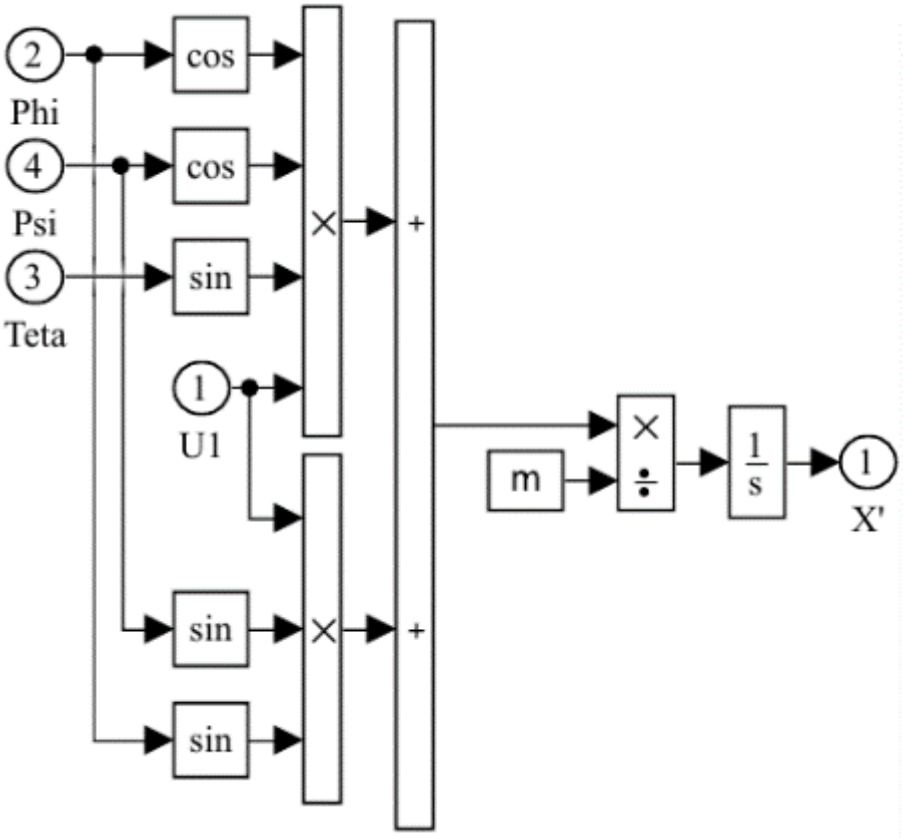


Рис. А.13 – Структура блоку Equations (X)

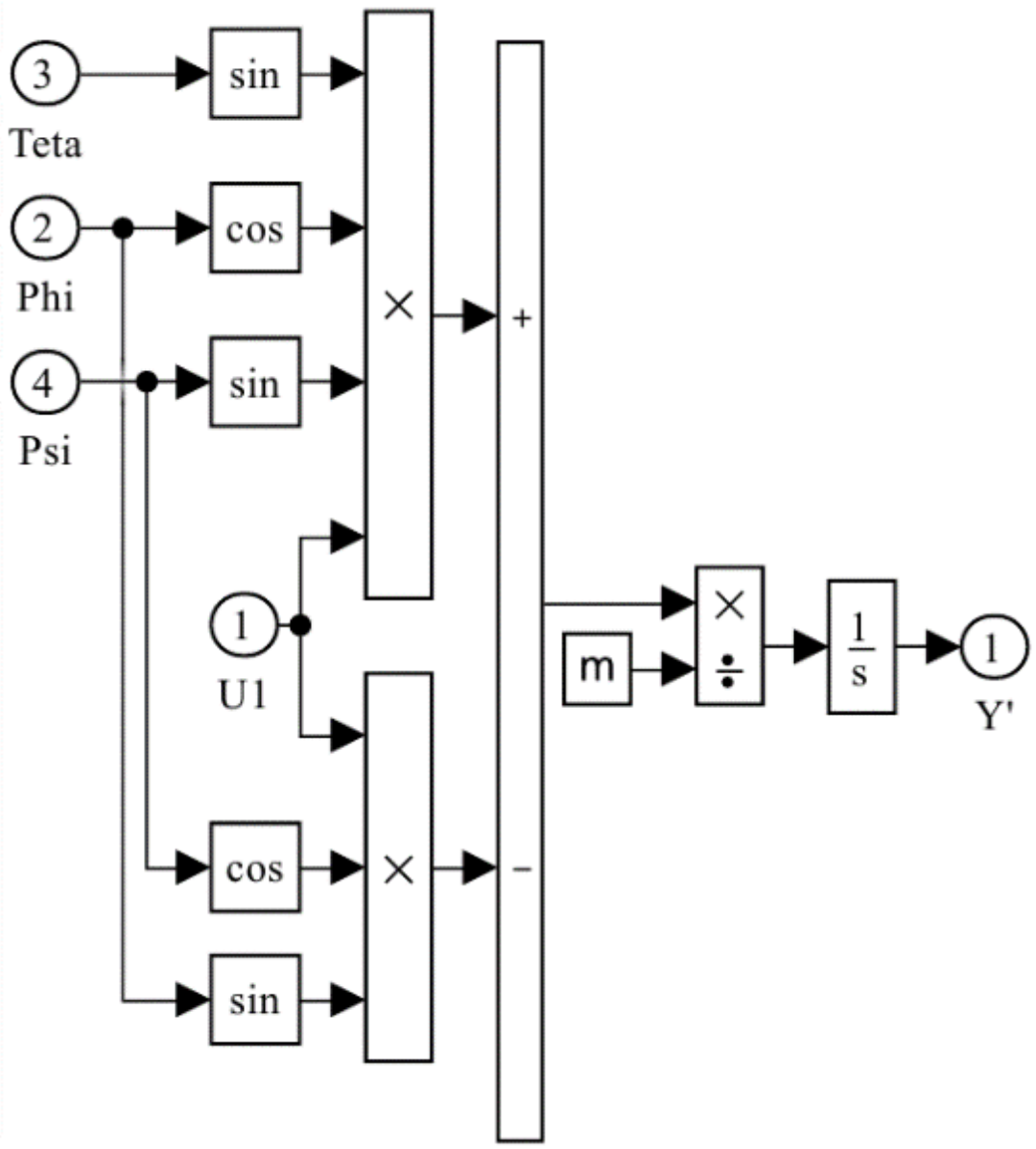


Рис. А.14 – Структура блоку Equations (Y)

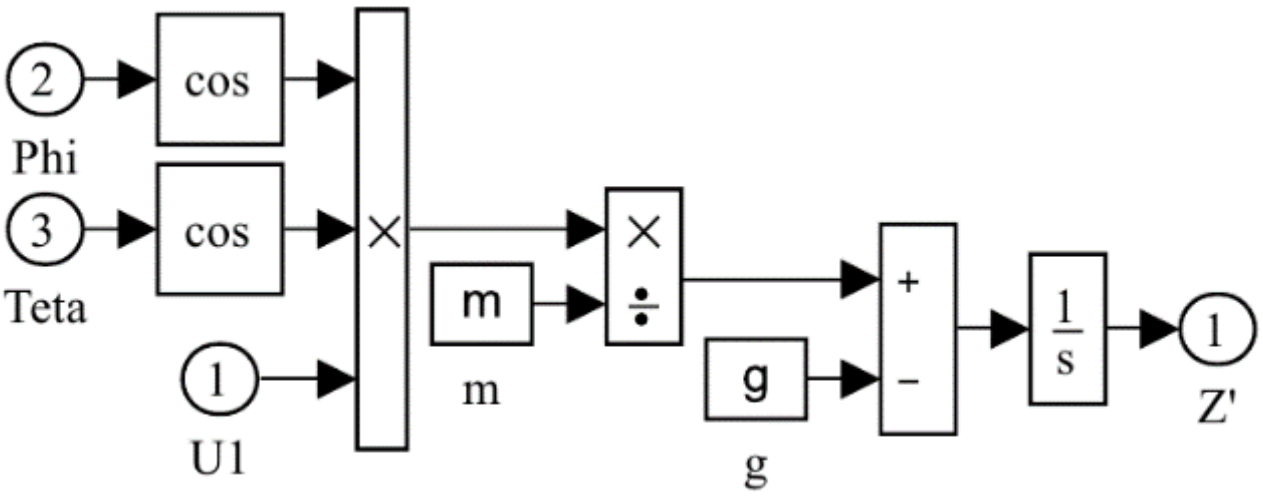


Рис. А.15 – Структура блоку Equations (Z)

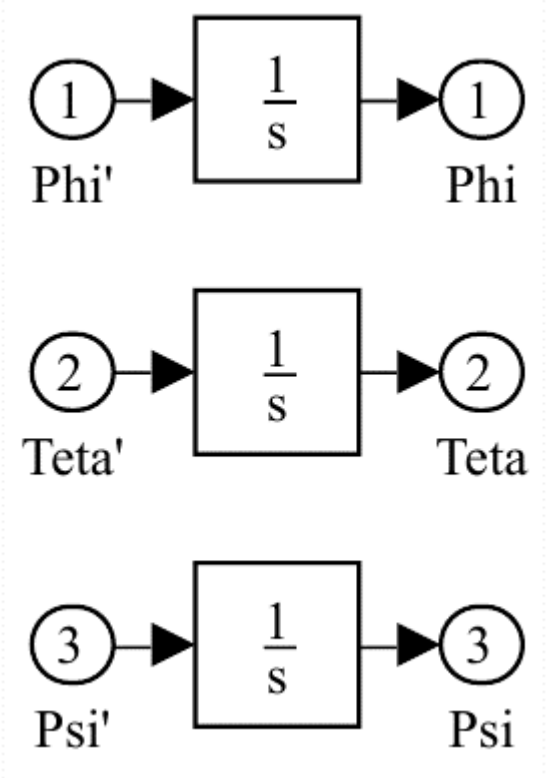


Рис. А.16 – Структура блоку Phi, Psi, Theta

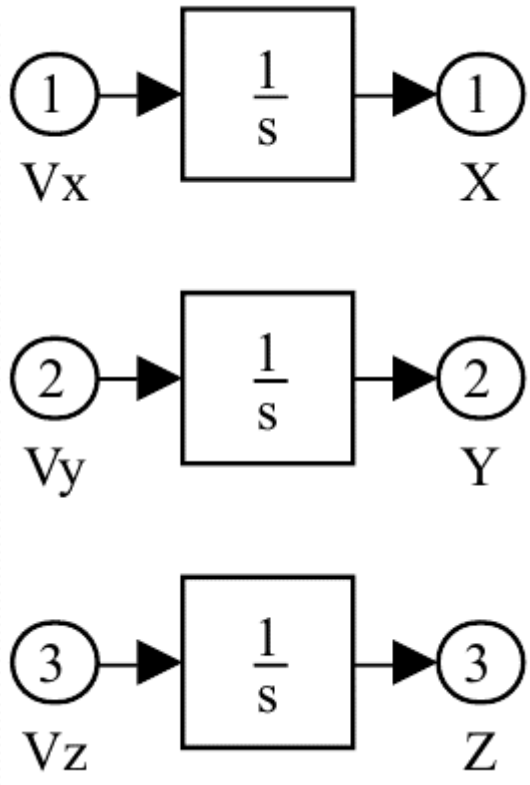


Рис. А.17 – Структура блоку X, Y, Z

Додаток Б. Моделювання аварійних ситуацій із застосуванням методів безпечного приземлення

Д.1 (d1.m) Скрипт запуску симуляції з побудовою графіків аварії

```
%% d1.m
%% Скрипт запуску симуляції з побудовою графіків аварії

clear
clc

%% Задаємо параметри моделі
l=0.175;
b=26.5*10^(-6);
d=0.6*10^(-6);
Ixx=0.1;
Iyy=Ixx;
Izz=Iyy;
Jtp=0.005;
g=9.807;
m=1;
Mg=m*g;
X0=0;
Y0=0;
Z0=50;
Omega_EM = 100;
Uh=sqrt(g/(4*b));

%% Запускаем симуляцію (для аварійної ситуації)
set_param('Model/Angle Velocities/emcy-safe', 'sw', '1');
set_param('Model/Angle Velocities/hand-auto', 'sw', '1');
set_param('Model/Angle Velocities/case selector', 'sw', '0');
set_param('Model/Angle Velocities/emcy case','commented','off');
set_param('Model/Angle Velocities/Auto','commented','on');
set_param('Model/Angle Velocities/Hand','commented','on');
set_param('Model/Angle Velocities/emcy case1','commented','on');
res = sim('Model', 'StartTime','0','StopTime','13.5');

%% Будуємо графіки величин, що відслідковуются
print_Figmod(res,'Po')
```

Д.2 (d2.m) Скрипт запуску симуляції аварії з ручним приземленням та побудовою графіків

```
%% d2.m
%% Скрипт запуску симуляції аварії з ручним приземленням
```

```
%% та побудовою графіків

clear
clc

%% Задаємо параметри моделі
l=0.175;
b=26.5*10^(-6);
d=0.6*10^(-6);
Ixx=0.1;
Iyy=Ixx;
Izz=Iyy;
Jtp=0.005;
g=9.8;
m=1;
Mg=m*g;
X0=0;
Y0=0;
Z0=50;
Uh=sqrt(g/(4*b));
Omega_EM = 100;

%% Запускаємо симуляцію (для аварійної ситуації з ручним рятуванням)
set_param('Model/Angle Velocities/emcy-safe', 'sw', '0');
set_param('Model/Angle Velocities/hand-auto', 'sw', '1');
set_param('Model/Angle Velocities/case selector', 'sw', '1');
set_param('Model/Angle Velocities/emcy case','commented','on');
set_param('Model/Angle Velocities/Auto','commented','on');
set_param('Model/Angle Velocities/Hand','commented','off');
set_param('Model/Angle Velocities/emcy case1','commented','on');
res = sim('Model', 'StartTime','0','StopTime','13.5');

%% Будуємо графіки величин, що відслідковуються
print_Figmod(res,'Povzp')
```

Д.3 (d3.m) Скрипт запуску симуляції з відмовостійким алгоритмом із застосуванням ПІД регуляторів та побудовою графіків

```
%% d3.m
%% Скрипт запуску симуляції з відмовостійким алгоритмом
%% із застосуванням ПІД регуляторів та побудовою графіків

clear
clc

%% Задаємо параметри моделі
l=0.350;
b=26.5*10^(-6);
d=0.6*10^(-6);
Ixx=0.1;
Iyy=Ixx;
Izz=Iyy;
Jtp=0.005;
g=9.807;
```

```
m=1;
Mg=m*g;
X0=0;
Y0=0;
Z0=50;
Uh=sqrt(g/(4*b));
Omega_EM = 0;
time_reaction = 0;

%% Задаємо коефіцієнти ПІД регулятора
PID=[50 8 20];

%% Запускаємо симуляцію (для аварійної ситуації з автоматичним
рятуванням)
set_param('Model/Angle Velocities/emcy-safe', 'sw', '0');
set_param('Model/Angle Velocities/hand-auto', 'sw', '0');
set_param('Model/Angle Velocities/case selector', 'sw', '0');
set_param('Model/Angle Velocities/emcy case','commented','on');
set_param('Model/Angle Velocities/Auto','commented','off');
set_param('Model/Angle Velocities/Hand','commented','on');
set_param('Model/Angle Velocities/emcy case1','commented','on');
res = sim('Model', 'StartTime','0','StopTime','50');

%% Будуємо графіки величин, що відслідковуються
print_Figmod(res,'Povzp')
```

Д.4 (d4.m) Допоміжна функція виведення графіків print_Figmod

```
%% d4.m
%% Допоміжна функція виведення графіків print_Figmod

function print_Figmod(res,names)
% на вході структура с даними та
% змінна-рядок (для вибору потрібних графіків)
X_bounds = [1*min(res.X)-0.1*abs(max(res.X)-min(res.X))
1*max(res.X)+0.1*abs(max(res.X)-min(res.X))];
Y_bounds = [1*min(res.Y)-0.1*abs(max(res.Y)-min(res.Y))
1*max(res.Y)+0.1*abs(max(res.Y)-min(res.Y))];
Z_bounds = [0 1*max(res.Z)+0.1*abs(max(res.Z)-min(res.Z))];
flagpos = false;
flagomega = false;
flagvz = false;
flagz = false;
flagpsi = false;
for i = 1:size(names,2)
  switch names(i)
    case 'P'
    flagpos = true;
    case 'o'
    flagomega = true;
    Om1_bounds = [1*min(res.Omega1)-0.1*abs(max(res.Omega1)-
min(res.Omega1))-3 1*max(res.Omega1)+0.1*abs(max(res.Omega1)-
min(res.Omega1))+3];
```

```
    Om2_bounds = [1*min(res.Omega2)-0.1*abs(max(res.Omega2)-
min(res.Omega2))-3 1*max(res.Omega2)+0.1*abs(max(res.Omega2)-
min(res.Omega2))+3];
    Om3_bounds = [1*min(res.Omega3)-0.1*abs(max(res.Omega3)-
min(res.Omega3))-3 1*max(res.Omega3)+0.1*abs(max(res.Omega3)-
min(res.Omega3))+3];
    Om4_bounds = [1*min(res.Omega4)-0.1*abs(max(res.Omega4)-
min(res.Omega4))-3 1*max(res.Omega4)+0.1*abs(max(res.Omega4)-
min(res.Omega4))+3];
    case 'v'
    flagvz = true;
    Vz_bounds = [1*min(res.Vz)-0.1*abs(max(res.Vz)-min(res.Vz))
1*max(res.Vz)+0.1*abs(max(res.Vz)-min(res.Vz))];
    case 'z'
    flagz = true;
    case 'p'
    flagpsi = true;
    Psi_bounds = [1*min(res.Psi)-0.1*abs(max(res.Psi)-
    min(res.Psi)) 1*max(res.Psi)+0.1*abs(max(res.Psi)-min(res.
    Psi))];
    otherwise
    disp('other value');
  end
end

%% Графік аварійного руху
if flagpos == true
  width = 16.5*3;
  height = 5*3;
  figure('Units','centimeters','Position',[1 1 width
height],'Name','Fail-safe
landing','NumberTitle','off','Color','white')
  plot3(res.X,res.Y,res.Z,'black','linewidth',3);
  xlabel('X, м','Interpreter','tex');
  ylabel('Y, м','Interpreter','tex');
  zlabel('Z, м','Interpreter','tex');

zticks(round(linspace(0,round(1*max(res.Z)),round(0.01*abs(max(res.Z
)-min(res.Z)))+3)));
  title('Траєктоpія переміщення апарату')
  grid on
  ax = gca;
  ax.ZLim = Z_bounds;
  set(ax, 'LineWidth', 1);
  set(ax,'FontName','Times New Roman','FontSize',28,'
  FontWeight','normal');
  ax.XDir = 'reverse';
  ax.YDir = 'reverse';
end

%% Графік швидкостей обертання гвинтів
if flagomega == true
  width = 16.5;
  height = 30;
```

```
  figure('Units','centimeters','Position',[1 1 width
height],'Name','Fail-safe landing:
Omega','NumberTitle','off','Color','white')
  title('Кутові швидкості \Omega_{i}')
  subplot(4,1,1);
  plot(res.tout,res.Omega1,'black','linewidth',3);
  axis([0 max(res.tout) Om1_bounds(1) Om1_bounds(2)]);
  grid on;
  xlabel('t, c','Interpreter','tex');
  ylabel('\Omega_{1}, об/с','Interpreter','tex');

yticks(round(linspace(0.95*min(res.Omega1),round(1*max(res.Omega1)),
round(0.01*abs(max(res.Omega1)-min(res.Omega1)))+3)))
  set(gca, 'LineWidth', 1);
  set(gca,'FontName','Times New
Roman','FontSize',12,'FontWeight','normal');
  subplot(4,1,2);
  plot(res.tout,res.Omega2,'black','linewidth',3);
  axis([0 max(res.tout) Om2_bounds(1) Om2_bounds(2)]);
  grid on;
  xlabel('t, c','Interpreter','tex');
  ylabel('\Omega_{2}, об/с','Interpreter','tex');

yticks(round(linspace(0.95*min(res.Omega2),round(1*max(res.Omega2)),
round(0.01*abs(max(res.Omega2)-min(res.Omega2)))+3)))
  set(gca, 'LineWidth', 1);
  set(gca,'FontName','Times New
Roman','FontSize',12,'FontWeight','normal');
  subplot(4,1,3);
  plot(res.tout,res.Omega3,'black','linewidth',3);
  axis([0 max(res.tout) Om3_bounds(1) Om3_bounds(2)]);
  grid on;
  xlabel('t, c','Interpreter','tex');
  ylabel('\Omega_{3}, об/с','Interpreter','tex');

yticks(round(linspace(0.95*min(res.Omega3),round(1*max(res.Omega3)),
round(0.01*abs(max(res.Omega3)-min(res.Omega3)))+3)))
  set(gca, 'LineWidth', 1);
  set(gca,'FontName','Times New Roman','FontSize',12,'
  FontWeight','normal');
  subplot(4,1,4);
  plot(res.tout,res.Omega4,'black','linewidth',3);
  axis([0 max(res.tout) Om4_bounds(1) Om4_bounds(2)]);
  grid on;
  xlabel('t, c','Interpreter','tex');
  ylabel('\Omega_{4}, об/с','Interpreter','tex');

yticks(round(linspace(0.95*min(res.Omega4),round(1*max(res.Omega4)),
round(0.01*abs(max(res.Omega4)-min(res.Omega4)))+3)))
  set(gca, 'LineWidth', 1);
  set(gca,'FontName','Times New
Roman','FontSize',12,'FontWeight','normal');
end

%% Будуємо графіки Vz, Z
if flagvz == true
```

```
  width = 16.5*3;
  height = 5*3;
  figure('Units','centimeters','Position',[1 1 width
height],'Name','Emergency (gorizontal
move)','NumberTitle','off','Color','white')
  plot(res.Vz,'black','linewidth',3);
  xlabel('t, c','Interpreter','tex');
  ylab = 'V_{z}, м';
  ylabel(ylab,'Interpreter','tex');
  title('Графік швидкості V_z');
  grid on
  ax = gca;
  ax.TickLabelInterpreter = 'tex';
  ax.XLim = [0 max(res.tout)];
  ax.YLim = Vz_bounds;
  set(ax, 'LineWidth', 1);
  set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');
end
if flagz == true
  width = 16.5*3;
  height = 5*3;
  figure('Units','centimeters','Position',[1 1 width
height],'Name','Emergency (gorizontal
move)','NumberTitle','off','Color','white')
  plot(res.tout,res.Z,'black','linewidth',3);
  xlabel('t, c');
  ylabel('Z, м');
  title('Графік висоти Z');
  grid on
  ax = gca;
  ax.TickLabelInterpreter = 'tex';
  ax.XLim = [0 max(res.tout)];
  ax.YLim = Z_bounds;
  set(ax, 'LineWidth', 1);
  set(ax,'FontName','Times New Roman','FontSize',28,'
  FontWeight','normal');
end

%% Графік кута Psi
if flagpsi == true
  width = 16.5*3;
  height = 5*3;
  figure('Units','centimeters','Position',[1 1 width
height],'Name','Emergency (gorizontal move)','NumberTitle','off',
  'Color','white')
  plot(res.Psi,'black','linewidth',3);
  xlabel('t, c','Interpreter','tex');
  ylab = '\psi, рад';
  ylabel(ylab,'Interpreter','tex');
  title('Графік кута \psi');
  grid on
  ax = gca;
  ax.TickLabelInterpreter = 'tex';
  ax.XLim = [0 max(res.tout)];
  ax.YLim = Psi_bounds;
```

```
  set(ax, 'LineWidth', 1);
  set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');
end
```

Д.5 (d5.m) Скрипт для побудови множини траєкторій переміщення апарата при різних $\boldsymbol{\Omega_i^{em}}$

```
%% d5.m
%% Скрипт для побудови множини траєкторій переміщення апарата при
різних Omega

Jtp=0.005;
g=9.807;
m=1;
Mg=m*g;
x0=0;
y0=0;
z0=50;
Uh=sqrt(g/(4*b));

%% Задаємо коефіцієнти ПІД регулятора
PID=[50 8 20];

%% Задаємо матрицю початкових даних (генерація набору параметрів)
X0 = x0;
Y0 = y0;
Z0 = z0;
omega_em = linspace(0,300,101);
om_l = size(omega_em,2);

%% Запускаємо симуляцію для каждого випадку
cases = struct;
open('Model.slx');
set_param('Model/Angle Velocities/emcy-safe', 'sw', '1');
set_param('Model/Angle Velocities/hand-auto', 'sw', '1');
set_param('Model/Angle Velocities/case selector', 'sw', '0');
set_param('Model/Angle Velocities/emcy case','commented','off');
set_param('Model/Angle Velocities/Auto','commented','on');
set_param('Model/Angle Velocities/Hand','commented','on');
set_param('Model/Angle Velocities/emcy case1','commented','on');

for i2 = 1:om_l
  disp(i2)
  Omega_EM = omega_em(i2);
  res = sim('Model', 'StartTime','0','StopTime','20');
  X = res.X;
  Y = res.Y;
  Z = res.Z;
  Z = Z(Z>0);
  Cx = X(1:length(Z));
  Cy = Y(1:length(Z));
  Cz = Z;
  [cases(i2).X] = Cx;
```

```
  [cases(i2).Y] = Cy;
  [cases(i2).Z] = Cz;
  [cases(i2).last_point] = [Cx(end),Cy(end),Cz(end)];
end
Xp = zeros(1,om_l);
Yp = zeros(1,om_l);
Zp = zeros(1,om_l);
for i = 1:(om_l)
  Ppos = cases(i).last_point;
  Xp(i) = Ppos(1);
  Yp(i) = Ppos(2);
  Zp(i) = Ppos(3);
end
printing = struct;
printing.X = Xp;
printing.Y = Yp;
printing.Z = Zp;

%% Будуємо графіки величин, що відслідковуються
width = 16.5*3;
height = 11*3;
Z_bounds = [0 50];
figure('Units','centimeters','Position',[1 1 width
height],'Name','Landing dots','NumberTitle','off','Color','white')
title('Множина траєкторій переміщення з різним значенням
\Omega_i^{em}')
plot3(cases(1).X,cases(1).Y,cases(1).Z,'black','linewidth',3);
hold on
grid on

for i=2:(om_l)
  plot3(cases(i).X,cases(i).Y,cases(i).Z,'black','linewidth',3);
end
xlabel('X, м','Interpreter','tex');
ylabel('Y, м','Interpreter','tex');
zlabel('Z, м','Interpreter','tex');
ax = gca;
ax.ZLim = Z_bounds;
set(ax, 'LineWidth', 1);
set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');
ax.XDir = 'reverse';
ax.YDir = 'reverse';

%% Графік кінцевих точок
width = 16.5*3;
height = 11*3;
Z_bounds = [0 50];
figure('Units','centimeters','Position',[1 1 width
height],'Name','Landing curves
(PID)','NumberTitle','off','Color','white')
scatter3(printing.X,printing.Y,printing.Z,'black','linewidth',3);
grid on
xlabel('X, м','Interpreter','tex');
ylabel('Y, м','Interpreter','tex');
zlabel('Z, м','Interpreter','tex');
```

```
title('Множина точок приземлення з різним значенням \Omega_i^{em}')
ax = gca;
ax.ZLim = Z_bounds;
set(ax, 'LineWidth', 1);
set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');
ax.XDir = 'reverse';
ax.YDir = 'reverse';
```

Д.6 (d6.m) Скрипт для побудови множини траєкторій переміщення апарата при різних $\boldsymbol{\Omega_i^{em}}$, $t_{reaction}$

```
%% d6.m
%% Скрипт для побудови множини траєкторій
%% переміщення апарата при різних Omega^em_i, t_reaction

clear
clc

%% Задаємо параметри моделі
l=0.175;
b=26.5*10^(-6);
d=0.6*10^(-6);
Ixx=0.1;
Iyy=Ixx;
Izz=Iyy;
Jtp=0.005;
g=9.807;
m=1;
Mg=m*g;
x0=0;
y0=0;
z0=50;
Uh=sqrt(g/(4*b));

%% Задаємо коэффициенты пиД регулятора
PID=[50 8 20];

%% Задаємо матрицу начальных данных (генерация набора параметро B)
X0 = x0;
Y0 = y0;
Z0 = z0;
omega_em = linspace(0,300,101);
time_r = linspace (0.5,1,6);

om_l = size(omega_em, 2);
tr_1 = size(time_r,2);

%% Запускаємо симуляцію для кожного випадку
cases = struct;
set_param('Model/Angle Velocities/emcy-safe', 'sw', '0');
set_param('Model/Angle Velocities/hand-auto', 'sw', '0');
set_param('Model/Angle Velocities/case selector', 'sw', '0');
set_param('Model/Angle Velocities/emcy case','commented','on');
```

```
set_param('Model/Angle Velocities/Auto','commented','off');
set_param('Model/Angle Velocities/Hand','commented','on');
set_param('Model/Angle Velocities/emcy case1','commented','on');
for i2 = 1:om_l
  for i3 = 1:tr_l
    i = i3 + tr_l*(i2-1)
    Omega_EM = omega_em(i2);
    time_reaction = time_r(i3);
    res = sim('Model', 'StartTime','0','StopTime','60');
    X = res.X;
    Y = res.Y;
    Z = res.Z;
    Z = Z(Z>0);
    Cx = X(1:length(Z));
    Cy = Y(1:length(Z));
    Cz = Z;
    [cases(i).X] = Cx;
    [cases(i).Y] = Cy;
    [cases(i).Z] = Cz;
    [cases(i).last_point] = [Cx(end),Cy(end),Cz(end)];
  end
end
Xp = zeros(1,tr_l*om_l);
Yp = zeros(1,tr_l*om_l);
Zp = zeros(1,tr_l*om_l);
for i = 1:(tr_l*om_l)
  Ppos = cases(i).last_point;
  Xp(i) = Ppos(1);
  Yp(i) = Ppos(2);
  Zp(i) = Ppos(3);
end
printing = struct;
printing.X = Xp;
printing.Y = Yp;
printing.Z = Zp;

%% Будуємо графіки величин, що відслідковуються
width = 16.5*3;
height = 11*3;
Z_bounds = [0 50];
figure('Units','centimeters','Position',[1 1 width
height],'Name','Landing curves
(PID)','NumberTitle','off','Color','white')
plot3(cases(1).X,cases(1).Y,cases(1).Z,'black','linewidth',3);
hold on
grid on
for i=2:(tr_l*om_l)
  plot3(cases(i).X,cases(i).Y,cases(i).Z,'black','linewidth',3);
end
xlabel('X, м','Interpreter','tex');
ylabel('Y, м','Interpreter','tex');
zlabel('Z, м','Interpreter','tex');
title('Множина траєкторій переміщення з різним значенням
\Omega_i^{em}, t_{reaction}')
ax = gca;
ax.ZLim = Z_bounds;
```

```
set(ax, 'LineWidth', 1);
set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');

%% Обираємо множини точок, що відповідають
%% конкретному часу переключення t_reaction
t_set1 = struct; j1 = 0;
t_set2 = struct; j2 = 0;
t_set3 = struct; j3 = 0;
t_set4 = struct; j4 = 0;
t_set5 = struct; j5 = 0;
t_set6 = struct; j6 = 0;
for i=1:length(printing.X)
  disp(i)
  switch mod(i,6)
    case 0
      j6 = j6+1;
      t_set6.X(j6) = printing.X(i);
      t_set6.Y(j6) = printing.Y(i);
      t_set6.Z(j6) = printing.Z(i);
    case 1
      j1 = j1+1;
      t_set1.X(j1) = printing.X(i);
      t_set1.Y(j1) = printing.Y(i);
      t_set1.Z(j1) = printing.Z(i);
    case 2
      j2 = j2+1;
      t_set2.X(j2) = printing.X(i);
      t_set2.Y(j2) = printing.Y(i);
      t_set2.Z(j2) = printing.Z(i);
    case 3
      j3 = j3+1;
      t_set3.X(j3) = printing.X(i);
      t_set3.Y(j3) = printing.Y(i);
      t_set3.Z(j3) = printing.Z(i);
    case 4
      j4 = j4+1;
      t_set4.X(j4) = printing.X(i);
      t_set4.Y(j4) = printing.Y(i);
      t_set4.Z(j4) = printing.Z(i);
    case 5
      j5 = j5+1;
      t_set5.X(j5) = printing.X(i);
      t_set5.Y(j5) = printing.Y(i);
      t_set5.Z(j5) = printing.Z(i);
  end
end

%% Графік кінцевих точок
width = 16.5*3;
height = 5*3;
Z_bounds = [0 50];
figure('Units','centimeters','Position',[1 1 width
height],'Name','Landing curves
(PID)','NumberTitle','off','Color','white')
scatter3(printing.X,printing.Y,printing.Z,'black','linewidth',3);
```

```
hold on
grid on
plot3(t_set1.X,t_set1.Y,t_set1.Z,'blue','linewidth',2);
plot3(t_set2.X,t_set2.Y,t_set2.Z,'green','linewidth',2);
plot3(t_set3.X,t_set3.Y,t_set3.Z,'cyan','linewidth',2);
plot3(t_set4.X,t_set4.Y,t_set4.Z,'yellow','linewidth',2);
plot3(t_set5.X,t_set5.Y,t_set5.Z,'red','linewidth',2);
plot3(t_set6.X,t_set6.Y,t_set6.Z,'magenta','linewidth',2);
xlabel('X, м','Interpreter','tex');
ylabel('Y, м','Interpreter','tex');
zlabel('Z, м','Interpreter','tex');
title('Множина точок приземлення з різним значенням \Omega_i^{em},
t_{reaction}')
ax = gca;
ax.ZLim = Z_bounds;
set(ax, 'LineWidth', 1);
set(ax,'FontName','Times New
Roman','FontSize',28,'FontWeight','normal');
```