Санкт–Петербургский государственный университет
Факультет прикладной математики, процессов управления
Кафедра моделирования экономических систем

Жолобов Ефим Витальевич

Выпускная квалификационная работа

Управление квадрокоптером в аварийных ситуациях

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»

Научный руководитель:
профессор, доктор физ.-мат. наук
Смирнов Николай Васильевич

Рецензент:
доцент, кандидат физ.-мат. наук
Степенко Николай Анатольевич

Санкт–Петербург
2021

Оглавление

Стр.

Стр.

Введение

Прогресс не стоит на месте. На протяжении десятилетий задача управления беспилотными летательными аппаратами (БПЛА) становится более актуальной. С помощью дронов вертолетного типа люди решают множество задач в различных сферах деятельности. С годами список возможных применений увеличивается. Высокие полетные характеристики при относительно малой сложности изготовления и зачастую невысокой стоимости лишь усилили популярность БПЛА. В этой связи последние 10 лет данные устройства стали широко использоваться как в промышленных целях, так и простыми обывателями.

Однако конструкция БПЛА, основывающаяся на нескольких тяговых двигателях, предполагает ряд значительных требований к осуществлению работы системы управления в рамках удовлетворения необходимости постоянной стабилизации аппарата в пространстве. В связи с этим ведется разработка как методов общего назначения (перемещения БПЛА в пространстве), так и вспомогательных алгоритмов (стабилизация, управление в аварийных ситуациях).

В настоящей работе рассматривается вопрос моделирования перемещений квадрокоптера для последующего построения такой системы управления, которая позволяла бы минимизировать вероятность негативных исходов в случае серьезных неисправностей. Естественно полагать, что такая (отказоустойчивая) система управления может работать только на аппаратах конструктивно допускающих работу в определённом (аварийном) режиме, при котором нарушена штатная работа некоторых подсистем аппарата.

Основные результаты докладывались и обсуждались на конференциях «Процессы управления и устойчивость» (2019 и 2021 гг.) [1], «13 th International Symposium on Intelligent Distributed Computing (IDC)» (2019 год) [2]; опубликованы в журнале «Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)» (Том 11, №2) [3].

Постановка задачи

Общую постановку задачи для исследования в целом можно разделить на три подзадачи:

На первом этапе происходит рассмотрение квадрокоптера как динамического объекта, которое включает в себя:

- Определение возможных перемещений объекта и модели, описывающей квадрокоптер как твёрдое тело с 6-ю степенями свободы.
- Выбор параметров системы на основе реальной конфигурации устройства.

Второй этап содержит:

- Определение аварийной ситуации с точки зрения математической модели квадрокоптера.
- Описание алгоритмов управления квадрокоптером в аварийных ситуациях.
- Формализация метода оценки расстояния посадки квадрокоптера для последующего использования в алгоритмах отказоустойчивого приземления.

Третий этап основывается на численных экспериментах:

- Моделирование аварийных ситуаций.
- Моделирование аварийного перемещения с алгоритмом «ручной» посадки.
- Моделирование аварийной ситуации с использованием отказоустойчивого алгоритма на основе ПИД-контроллера.
- Эксперимент по оценке расстояния приземления БПЛА в аварийной ситуации.

Дополнительной целью является написание удобного программного обеспечения для проведения численных экспериментов.

Обзор литературы

БПЛА значительно упрощают людям работу во многих сферах деятельности. Квадрокоптеры вертолетного типа активно используются в широком спектре задач [4]: от гражданских и промышленных (любительская съемка [5], частные исследования [6], доставка предметов [7], экстренные ситуации [8]) до государственных и военных (стратегические миссии [9], боевые операции).

Различные применения квадрокоптеров часто подразумевают продолжительное маневрирование аппарата в непосредственной близости с неподвижными объектами. В некоторых задачах необходимо нести полезные нагрузки (например, огнемет для удаления мусора [10]), порождающие дополнительные силы, дестабилизирующие аппарат. При таком движении высока вероятность аварийных ситуаций, приводящих к нарушению управляемости и даже к неконтролируемому падению. Это делает актуальной задачу моделирования и анализа аварийных движений квадрокоптера.

Пример построения модели квадрокоптера в качестве твёрдого тела с 6-ю степенями свободы наиболее полно представлены в [11]. Первыми работами, описывающими конструирование системы управления аппаратом, основанной на обработке сигналов, поступающих от различных бортовых датчиков, являются [11; 12]. Актуальность разработки методологии, улучшающей «живучесть» БПЛА и не ухудшающей работоспособность самого аппарата, отображена в работах [13; 14].

Обычно под управлением беспилотным летательным аппаратом подразумевается контроль транспортного средства (ТС) в соответствии с полетным заданием. Однако важно понимать, что некоторые подходы к данной задаче могут быть дополнены алгоритмами, направленными на анализ поведения квадрокоптера [15]. Например, нахождение БПЛА над поверхностью с водой (охваченной пламенем площадью, густым лесом) [16] вносит некоторые ограничения на управление аппаратом в виде отсутствия пространства для посадки. В случае возникновения аварии наличие прогноза поведения аппарата может существенно повлиять на успешность аварий-

ной программы посадки. В связи с этим имеет смысл формализация метода оценки расстояния посадки квадрокоптера для последующего использования результатов в алгоритмах управления аварийной посадкой.

Глава 1. Моделирование квадрокоптера как динамического объекта

1.1 Математическая модель

Квадрокоптер среди БПЛА характеризуется моделью с четырьмя роторами в поперечной конфигурации. Согласно работе [11], данная конструкция достаточно жёсткая и единственное, что может меняться, — это угловые скорости пропеллеров. Оси вращения винтов неподвижны и параллельны.

Основные параметры квадрокоптера представлены на рис. 1.1.

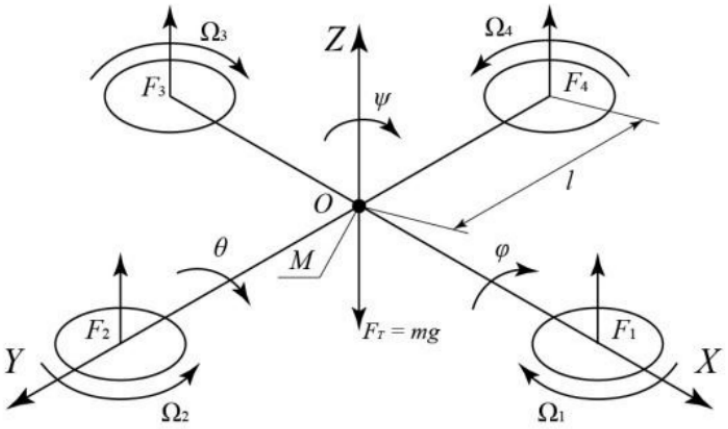


Рисунок 1.1 — Основные параметры квадрокоптера

На аппарат действуют подъёмные силы двигателей F_1, F_2, F_3, F_4, сила тяжести F_T. Точка M — центр масс аппарата (совпадает с геометрическим центром), l — расстояние от центра аппарата до точки приложения подъёмных сил.

Неподвижная система координат жёстко связана с землёй: Ось X' показывает на север, Y' — на запад, Z' — вверх по отношению к земле. Подвижная система координат жёстко связана с корпусом аппарата: ось X направлена вдоль направления движения квадрокоптера вперед, Y — по

направлению квадрокоптера влево, Z — вверх. Обе системы координат правосторонние [17].

Пропеллеры, находящиеся на оси X, вращаются по часовой стрелке, а винты на оси Y — против часовой стрелки. Такая конфигурация противоположных направлений пар устраняет необходимость в хвостовом винте, присутствующем в стандартной конструкции вертолета. Стрелками Ω_1, Ω_2, Ω_3, Ω_4 показаны направления вращения винтов.

Будучи объектом с 6-ю степенями свободы, квадрокоптер оснащен всего четырьмя пропеллерами, что позволяет достичь желаемого контроля лишь для 4-х. Однако, благодаря структуре аппарата, довольно легко выбрать четыре лучшие контролируемые переменные и разделить их, чтобы упростить управление. Таким образом, при построении модели целесообразно определить четыре основных перемещения, которые позволяют квадрокоптеру достигать определенной высоты и положения:

1. Ускорение вверх/вниз ($U_1 \rightarrow \ddot{Z}$). Данная команда характеризуется изменением угловых скоростей всех четырёх винтов на одинаковую величину.
2. Угловое ускорение крена ($U_2 \rightarrow \ddot{\varphi}$). Данная команда характеризуется изменением угловых скоростей винтов на оси Y таким образом, что аппарат начинает вращательное движение вокруг оси X.
3. Угловое ускорение тангажа ($U_3 \rightarrow \ddot{\theta}$). Данная команда характеризуется изменением угловых скоростей винтов на оси X таким образом, что аппарат начинает вращательное движение вокруг оси Y.
4. Угловое ускорение рысканья ($U_4 \rightarrow \ddot{\psi}$). Данная команда характеризуется изменением угловых скоростей всех четырёх винтов, причём пропеллеры, находящиеся на разных осях, получают противоположное влияние таким образом, что аппарат начинает вращательное движение вокруг оси Z.

В итоге перемещение квадрокоптера можно считать суммой поступательного движения центра масс и сферического движения тела относительно центра масс. Такое движение может быть описано следую-

щей системой [18]:

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

где x, y, z — координаты центра масс, V_x, V_y, V_z — проекции вектора линейной скорости, φ, θ, ψ — угол крена, тангажа и рыскания соответственно, ω_φ — угловая скорость крена, ω_θ — угловая скорость тангажа, ω_ψ — угловая скорость рыскания, m — масса квадрокоптера, I_{xx}, I_{yy}, I_{zz} — моменты инерции вокруг осей x, y и z соответственно, U_1, U_2, U_3, U_4 — каналы управления, Ω — общая скорость четырёх винтов, J_{TP} — общий вращательный момент инерции вокруг оси винта:

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

где J_P — момент инерции двигателя, J_M — момент инерции пропеллера, N — передаточное число, η — эффективность передачи.

Уравнения связи каналов управления U_1, U_2, U_3, U_4 со скоростями вращения винтов Ω_1, Ω_2, Ω_3, Ω_4 имеют вид

$$\begin{gathered}
U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\
U_2 = lb(-\Omega_2^2 + \Omega_4^2), \\
U_3 = lb(-\Omega_1^2 + \Omega_3^2), \\
U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \\
\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4,
\end{gathered} \tag{1.3}$$

где l – расстояние между центрами квадрокоптера и пропеллера, b и d — аэродинамические составляющие тяги и коэффициента сопротивления соответственно. Квадрокоптер приводится за счёт вращения пропеллеров, скорости вращения которых можно выразить из системы уравнений (1.3):

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

Таким образом, можно определить задачу управления квадрокоптером как задачу построения стратегии управления скоростями вращения четырёх моторов Ω_1, Ω_2, Ω_3, Ω_4 так, чтобы обеспечить асимптотически устойчивое положение квадрокоптера x_0, y_0, z_0 с удержанием одного из углов (например, угла рыскания φ_0). При этом сама точка x_0, y_0, z_0 выбирается в соответствии с полетным заданием.

Важно отметить, что описанная система является математической моделью некоторого «идеального» квадрокоптера, перемещающегося в среде без возмущений. Такая модель позволяет исследовать как характер изменения наблюдаемых величин во времени, так и их зависимости. Эмпирический подбор констант в модели дает возможность достаточно точно приблизить расчетные результаты к наблюдаемым на реальном аппарате [19–21], однако полностью заменить систему управления квадрокоптером, построенную на датчиках и регуляторах, такой системой невозможно [22].

1.2 Выбор параметров системы на основе реальной конфигурации устройства

Рынок БПЛА изобилует большим разнообразием различных конфигураций: от маленьких устройств [23] до больших грузовых [24]. В то же время существуют конструктивные подходы к классификации аппаратов [25]. В дальнейшем рассмотрении модель будет иметь параметры, соответствующие квадрокоптеру с диаметром рамы, равным 350 мм. [3]:

Таблица 1 — Пример конфигурации отказоустойчивого квадрокоптера с рамой диаметром 350 мм

Характеристика	**Значение**
Емкость батареи	$3\,\text{А} \cdot \text{ч}$
Запасённая энергия	$44\,\text{Вт} \cdot \text{ч}$
Нагрузка на аккумулятор	$19\,\text{Кл}$
Номинальное напряжение АКБ	$14{,}8\,\text{Вт}$
Вес аппарата (снаряженная масса)	$1\,\text{кг}$
Совокупный вес роторов	$0{,}726\,\text{кг}$
Модель электродвигателя	Turnigy Multistar Elite 2810-750
Тяговооруженность	3,6 снаряженных масс
Удельная тяга	$8{,}43\,\text{г}/\text{Вт}$
Минимальное полетное время	$2{,}8\,\text{мин}$
Время висения	$19{,}9\,\text{мин}$
Макс. горизонтальная скорость	$54\,\text{км}/\text{ч}$
Газ висения (линейный)	$42\ \%$
Ток электродвигателя (висение)	$1{,}92\,\text{А}$
Ток электродвигателя (максимальный режим)	$14{,}27\,\text{А}$
Максимальный ток ЭРХ электродвигателя	$20\,\text{А}$
КПД (висение)	$83{,}9\ \%$
КПД (максимальный режим)	$85{,}1\ \%$

Параметры математической модели (1.1), соответствующие аппарату указанной конфигурации, имеют следующий вид:

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

Глава 2. Возможные управления квадрокоптером в аварийных ситуациях

В данной главе описаны особенности построения математической модели квадрокоптера, такие как:

1. Определение аварийной ситуации/действия спасательного алгоритма;
2. Определение возможных управлений квадрокоптером;
3. Алгоритм действия ПИД-регулятора;
4. Оценка дальности приземления аппарата в аварийной ситуации.

2.1 Определение аварийной ситуации. Спасательный алгоритм

Применительно к модели (1.1) рассмотрим ситуации, связанные с выходом из строя одного или нескольких винтов. Причиной отказа пропеллера могут послужить разные причины [22]. Тем не менее, с точки зрения модели нарушение работоспособности может быть представлено как мгновенное изменение соответствующих величин:

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

Величина ε_i несет достаточно простой практический смысл — количественное ухудшение (в оборотах в секунду) работоспособности винта вследствие внешних (касание пропеллера инородных поверхностей) или внутренних (сбой электрической части двигателя) неисправностей. Случай, когда $\varepsilon_i = \Omega_i$ и, следовательно, $\Omega_i^{em} = 0$, будет описывать ситуацию с полной неработоспособностью соответствующего ротора. Подстановка аварийных значений Ω_i^{em} в выражения для элементов управления (1.3) и далее в систему (1.1) определит математическую модель аварийной ситуации.

Имея постановку задачи моделирования аварийной ситуации, целесообразно также ввести план как внешней (человек), так и внутренней (программа) реакции на возникновение внештатного положения в виде соответствующей обстоятельствам методологии (посадка квадрокоптера «вручную» или автоматизированное приземление аппарата). Таким образом, далее под спасательным алгоритмом будет подразумеваться управление двигателями аппарата таким образом, чтобы стабилизировать аппарат относительно высоты, минимизируя вероятность неконтролируемого падения:

$$\Omega_i = \Omega_i^{fs}, \qquad i = \overline{1,4} \tag{2.2}$$

где Ω_i^{fs} — угловые скорости вращения винтов, определяемые аварийной программой посадки. Подстановка значений Ω_i^{fs} в выражения для элементов управления (1.3) и далее в систему (1.1) описывает влияние спасательного алгоритма в рамках математической модели аварийной ситуации.

2.2 Определение возможных управлений квадрокоптером

Учитывая описанные ранее в п. 1.1 1 главы возможные перемещения аппарата, для реализации стандартных типов движений квадрокоптера (взлёт, посадка, движение в горизонтальной плоскости) могут быть использованы управляющие сигналы вида [26]

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

Здесь C_i — угловая скорость, необходимая для компенсации силы тяжести, a_k и b_k — параметры тригонометрического многочлена, задающие отклонения, необходимые для гладкого выхода на стационарный режим, T_1 — множество отрезков времени, при которых аппарат поддерживает висение, T_2 — множество отрезков времени, при которых аппарат маневрирует.

Такое представление управляющих сигналов близко (является приближением с помощью тригонометрического многочлена) к оптимальному по

расходу энергии [26] и при этом является гладкой функцией, что применимо в практическом смысле. Иными словами, такая форма позволяет аппроксимировать сигналы Ω_i^{fs} из выражения (2.2) при симуляции реализации спасательного алгоритма , и Ω_i^{em} из (2.1) при моделировании последствий аварий.

Посредством непрерывного анализа данных с гироскопов, акселерометров и данных о скорости вращения винтов легко определить отказ одного из двигателей. Признаки нарушения работоспособности, такие как разрушение винта без выхода из строя электрической части (отказ электрической части ротора), влекут резкое изменение скорости вращения винта, что не является нормой в стандартном режиме полета. Более того, подобная неисправность провоцирует вращение аппарата с последующим неконтролируемым падением. Тем не менее, наличие этих признаков может быть определено полетным контроллером с высокой точностью не более чем за 0,5 секунды.

Таким образом, спасательный алгоритм предполагает осуществление мер, направленных на управление высотой аппарата для его посадки. Как было сказано ранее, реализация данной методологии может основываться как на внешнем воздействии (человек), так и на внутренней конфигурации (автоматическое переключение из нормального режима управления в аварийный и последующее управление в соответствии с отказоустойчивым алгоритмом). Численное моделирование аварийной ситуации с «ручным» управлением описано в п. 3.2. Симуляция внештатного положения с автоматизированной реализацией спасательного алгоритма показана в п. 3.3.

2.3 Алгоритм воздействия ПИД-контроллера на квадрокоптер

Сформулируем методологию отказоустойчивого приземления для случая программной реакции на возникновение аварийной ситуации. Для

стабилизации положения аппарата относительно высоты было принято решение использовать алгоритм на основе ПИД-контроллеров. Поэтому вначале рассмотрим уравнение выходного сигнала ПИД-регулятора [20]:

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

где K_p, K_i, K_d — коэффициенты усиления пропорциональной, интегрирующей и дифференцирующей составляющих регулятора.

Каждый из параметров ПИД-контроллера имеет ряд особенностей, которые важно учитывать при настройке регулятора под квадрокоптер с конкретной конфигурацией:

1. **Пропорциональная составляющая K_p:**
   - Усиление влияния пропорциональной составляющей путем увеличения коэффициента K_p обеспечивает большую устойчивость. Однако важно заметить, что слишком большое значение K_p может привести как к резким колебаниям, так и к потере управляемости.
   - Ослабление влияния пропорциональной составляющей путем уменьшения коэффициента K_p уменьшает силу реакции аппарата на управляющее воздействие.
2. **Интегральная составляющая K_i:**
   - Повышение влияния интегральной составляющей путем увеличения коэффициента K_i обеспечивает усиление курсовой устойчивости (аппарат приобретает улучшенную инертность) и ослабляет дрейф. Однако вместе с этим возрастает длительность возврата в начальное положение. Также важно заметить, что при постоянном K_p увеличение K_i влечет уменьшение влияния K_p на управляющее воздействие.
   - Ослабление влияния интегральной составляющей путем уменьшения коэффициента K_i увеличивает скорость реакции квадрокоптера на управляющее воздействие, но вместе с этим усиливает дрейф и ухудшает способность удерживать стабильное положение.

3. **Дифференциальная составляющая K_d:**
   - Усиление влияния дифференциальной составляющей путем увеличения коэффициента K_d повышает скорость стабилизации при изменении положения аппарата в пространстве после внешнего воздействия или управления. Также возрастание K_d значительно увеличивает влияние пропорциональной составляющей, что способствует повышению вероятности появления чрезмерного регулирования и осцилляций.
   - Ослабление влияния дифференциальной составляющей путем уменьшения коэффициента K_d уменьшает скорость и размер колебаний в процессе возвращения аппарата в стабилизированное положение.

Указанные выше особенности были учтены при подборе параметров ПИД-контроллера в рамках численного моделирования в п. 3.3. Общий план использования ПИД-регуляторов при реализации автоматического алгоритма отказоустойчивого приземления с целью безопасной посадки аппарата имеет следующий вид:

1. На участке снижения ($t \in [t_1,t_2]$) ПИД-регулятор имеет набор параметров для стабилизации скорости снижения к заданной величине;
2. На участке торможения ($t \in [t_3,t_4]$) ПИД-регулятор имеет набор параметров для гашения вертикальной скорости к моменту касания земли ($z' = 0$ при $z = 0$).

2.4 Метод оценки расстояния посадки квадрокоптера

БПЛА имеют широкое распространение как среди гражданского населения, так и в промышленной среде. Поэтому количество различных ситуаций, в которых может находиться аппарат, весьма разнообразно. В случае возникновения аварии важно понимать, в какой среде находится аппарат, что его окружает. Квадрокоптер может оказаться неисправным,

находясь над поверхностью с водой, или даже огнём (если это пожарный дрон [27]). Любая подобная ситуация влечёт ограничения на посадку аппарата, которые, если их не учитывать при реализации алгоритма отказоустойчивого приземления, могут значительно повлиять на конечный исход. С учётом этого, становится очевидно, что минимизация вероятности бесконтрольного падения не является достаточным условием для успешного проведения операции спасения как квадрокоптера, так и полезной нагрузки (которая может не иметь защиты от воды/огня). В связи с этим предлагается подход к оценке расстояния посадки БПЛА, основанный на проведении ряда численных экспериментов.

В рамках математической модели (п. 1.1) для понимания динамики перемещения аппарата целесообразно провести численное моделирование с изменением некоторых параметров. При анализе аварийных ситуаций такими параметрами могут являться:

1. Величина Ω_i^{em} оказавшегося неисправным ротора вследствие одной из возможных причин поломки [22];
2. Время начала реакции на возникновения аварийной ситуации (алгоритм «ручной» посадки или реализация отказоустойчивого алгоритма с использованием ПИД-регуляторов). Обозначим данный параметр как $t_{reaction}$.

Изменяя определённым образом данные параметры, в силу теоремы о непрерывности решений систем дифференциальных уравнений от правых частей, мы сможем получить совокупность точек приземления аппарата, которая и будет определять ожидаемую зону падения квадрокоптера в аварийной ситуации. Реализация данного метода представлена в п. 3.4.

Глава 3. Численные эксперименты

Данная глава состоит из описания численных экспериментов, демонстрирующих возможности как математической модели, так и подходов к взаимодействию с ней. В среде MATLAB с дополнением Simulink разработан программный пакет (см. приложения А и Б), который позволяет моделировать аварийные ситуации и решать проблемы управления с целью минимизации последствий аварийных ситуаций.

В первом разделе предлагается исследовать последствия полной потери тяги на одном из двигателей квадрокоптера при отсутствии каких-либо алгоритмов спасения [1]. Результатом моделирования будет картина возможных исходов такого отказа. Для моделирования выбраны следующие аварийные ситуации:

1. С ухудшением работоспособности одного двигателя при горизонтальном полете,
2. С полной потерей функционирования одного винта при снижении аппарата.

Моделирование обеих ситуаций происходит при $0 < t < 15$ с отказом в момент t^*.

Второй раздел содержит демонстрацию спасательной методологии [2; 3]:

1. Алгоритм «ручной» безопасной посадки квадрокоптера при участии внешнего управления (оператор БПЛА) с учётом соображений по обнаружению неисправностей, описанных в п. 2.2.
2. Автоматический отказоустойчивый алгоритм на основе ПИД-регуляторов, описанных в п. 2.3.

Целью третьего раздела является формализация метода оценки ожидаемой области падения (или безопасного приземления) квадрокоптера (см. п. 2.4) при возникновении аварийной ситуации (или реализации спасательного алгоритма при обнаружении неисправности аппарата).

При реализации численных экспериментов используются параметры квадрокоптера (1.5), соответствующие конфигурации, описанной в п. 1.2.

3.1 Моделирование аварийных ситуаций

3.1.1 Авария при горизонтальном движении

Проведем моделирование частичной потери работоспособности одного из двигателей (уменьшения тяги) в процессе горизонтального перемещения квадрокоптера. Аварии подобного характера могут происходить в разных ситуациях, например в процессе нахождения в непосредственной близости с исследуемыми объектами. Даже в случае касания одним из винтов дрона неподвижного объекта высока вероятность нарушения целостности его лопастей, что в лучшем случае влечёт уменьшение тяги повреждённого двигателя, а в худшем — полную потерю работоспособности.

В рамках математической модели квадрокоптера угловые скорости вращения Ω_i определяются на основе ранее описанного в п. 2.2 способа задания управлений аппаратом, т.е. проводится приближение сигнала Ω_i к оптимальному. Для этого (2.3) строятся в форме:

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

1. В момент времени $t_0 = 0$ с все двигатели вращаются с необходимой для компенсации силы тяжести скоростью $C = 304{,}1691$ $^{\text{об}}/_{\text{с}}$
2. Угловые скорости $\Omega_{1,2}$ до момента $t^* = 4$ с задаются указанным полиномом, далее определены как C;
3. В момент времени $t = t^*$ происходит частичный отказ третьего двигателя ($\Omega_3 = 100$ $^{\text{об}}/_{\text{с}}$ при $t^* \geqslant 4$ с).

На рис. 3.1 представлен график угловых скоростей винтов аппарата. Движение аппарата при данном способе задания Ω_i, начиная с момента времени t^*, представляет собой падение по спирали (рис. 3.4).

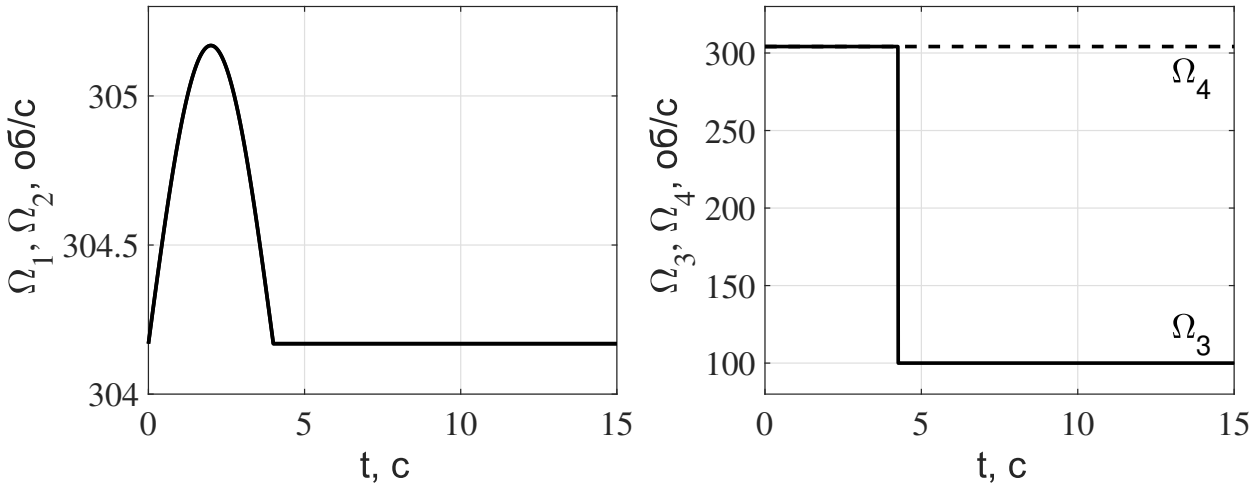


Рисунок 3.1 — Авария при горизонтальном полете: угловые скорости Ω_i

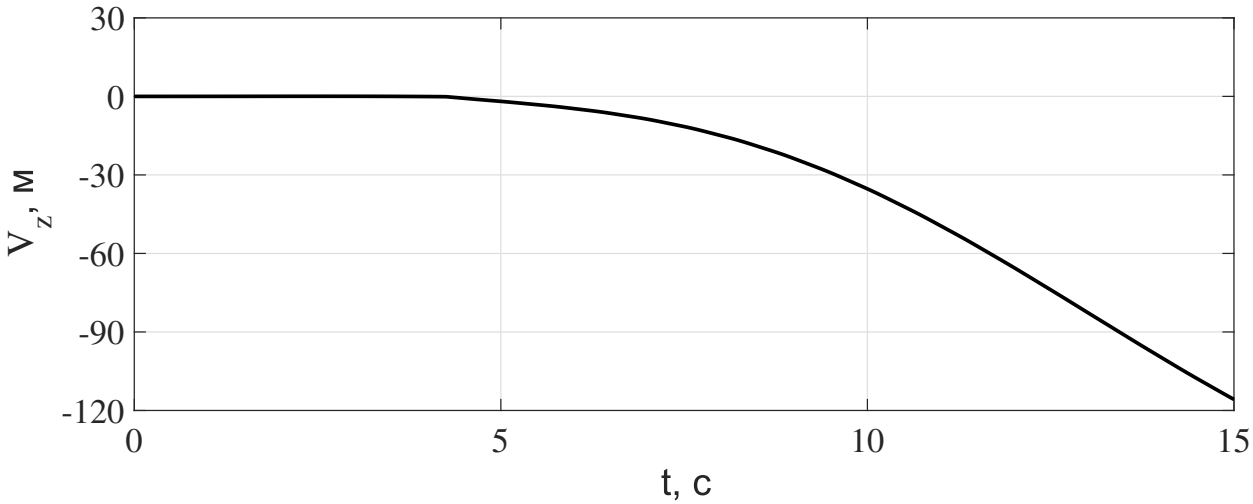


Рисунок 3.2 — Авария при горизонтальном полете: скорость V_z

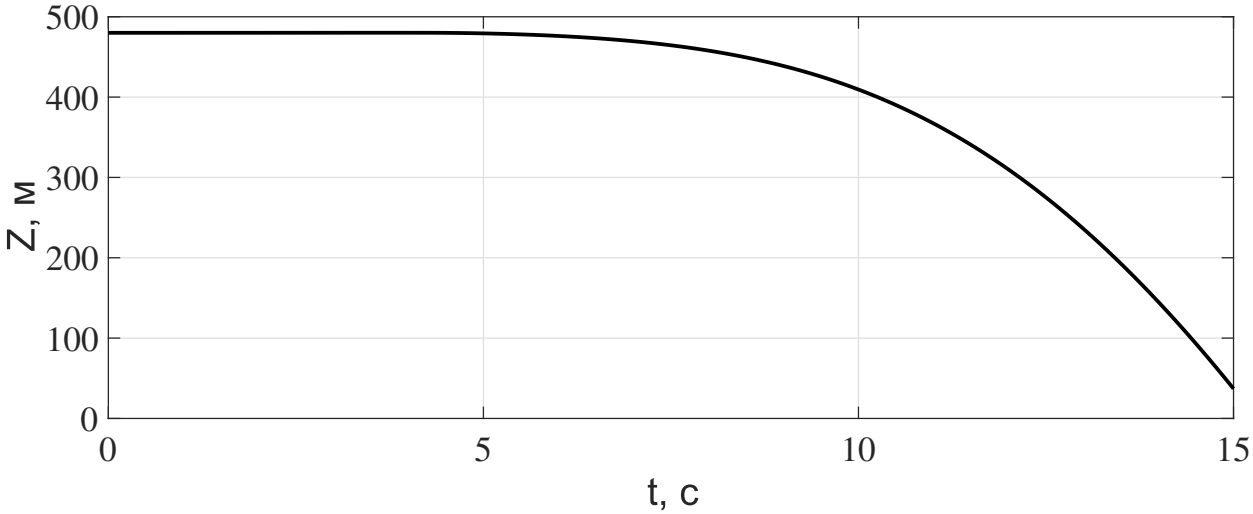


Рисунок 3.3 — Авария при горизонтальном полете: высота Z

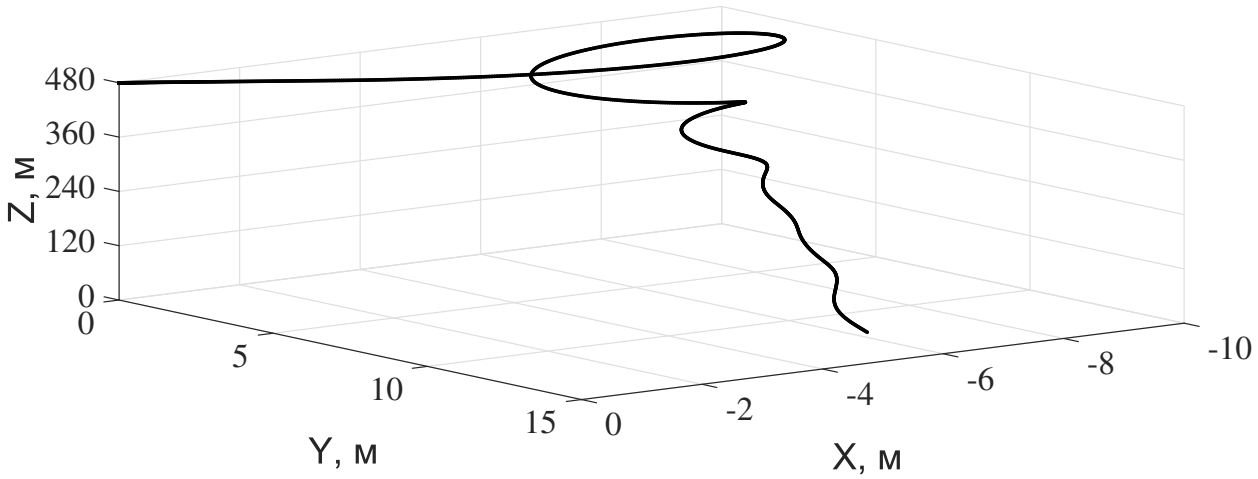


Рисунок 3.4 — Авария при горизонтальном полете: траектория перемещения аппарата

3.1.2 Авария при вертикальном перемещении

Пусть аварийная ситуация наступила в процессе снижения аппарата после неподвижного висения в точке $(x_0,\ y_0,\ z_0) = (0,\ 0,\ 250)$. С точки зрения математической модели получим следующие изменения угловых скоростей Ω_i:

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

1. При $t \in [0;\ 4]$ с все двигатели вращаются с необходимой для компенсации силы тяжести скоростью $C = 304{,}1691$ об/с;
2. При $t \in [4;\ 5]$ с аппарат получает отрицательное вертикальное ускорение вследствие снижения угловых скоростей;
3. При $t \in [5;\ 6]$ с проводится возвращение угловых скоростей к необходимым для висения значениям (происходит компенсация отрицательного ускорения);
4. При $t = t^* = 6$ с происходит аварийная ситуация — обнуление угловой скорости Ω_1.

На рис. 3.5 представлен график угловых скоростей винтов аппарата. Движение аппарата при данном способе задания Ω_i, начиная с момента времени t^*, представляет собой бесконтрольное падение (рис. 3.8).

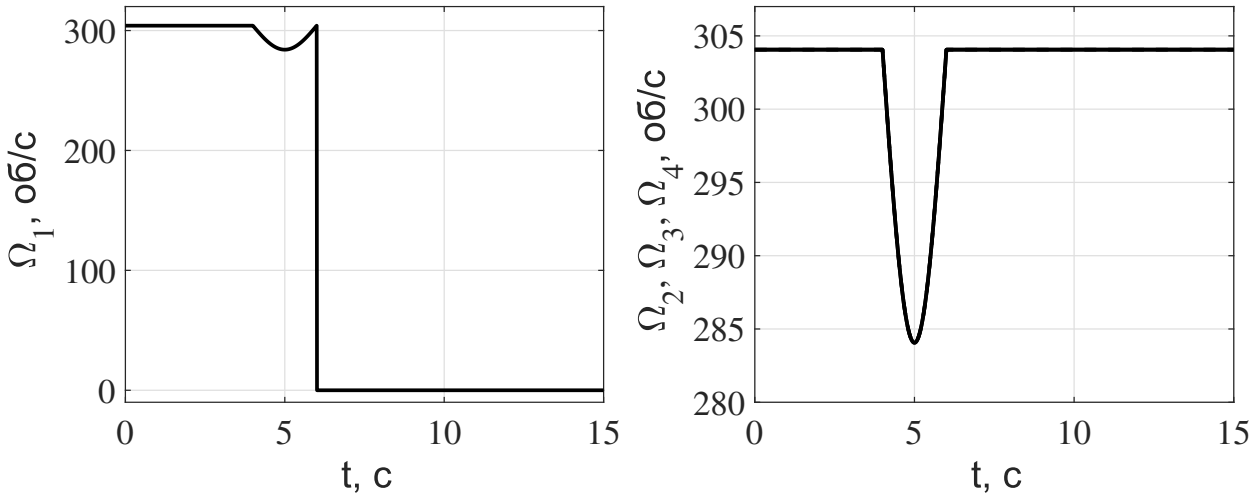


Рисунок 3.5 — Авария при снижении: угловые скорости Ω_i

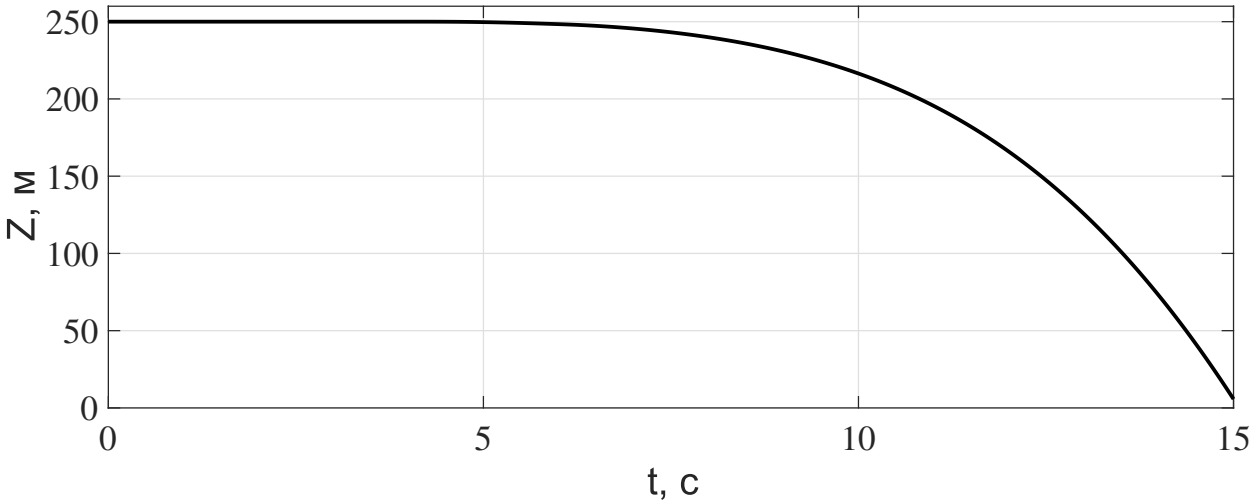


Рисунок 3.6 — Авария при снижении: высота Z

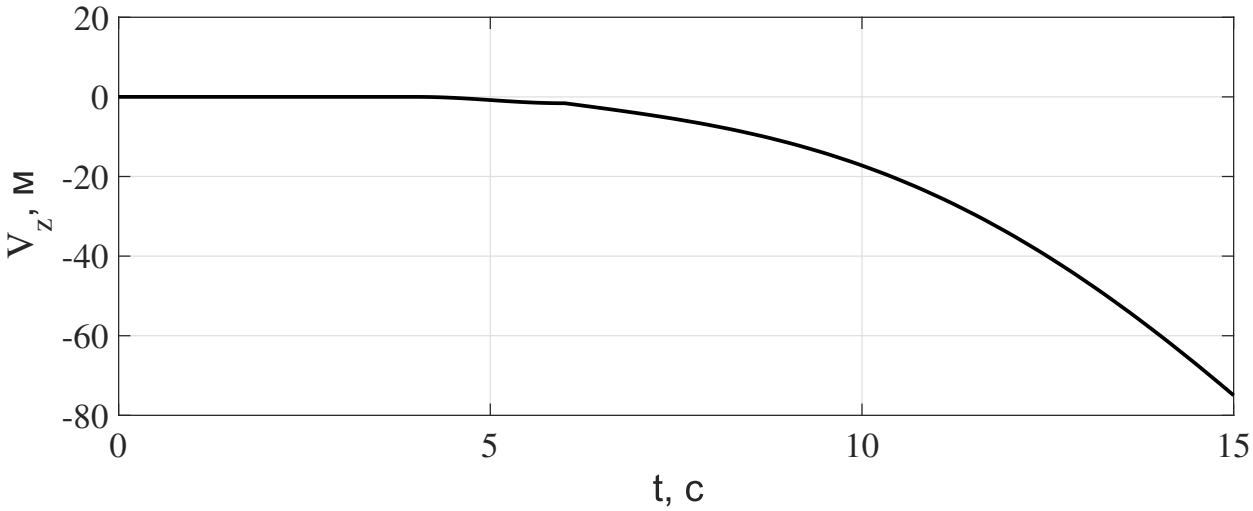


Рисунок 3.7 — Авария при снижении: скорость V_z

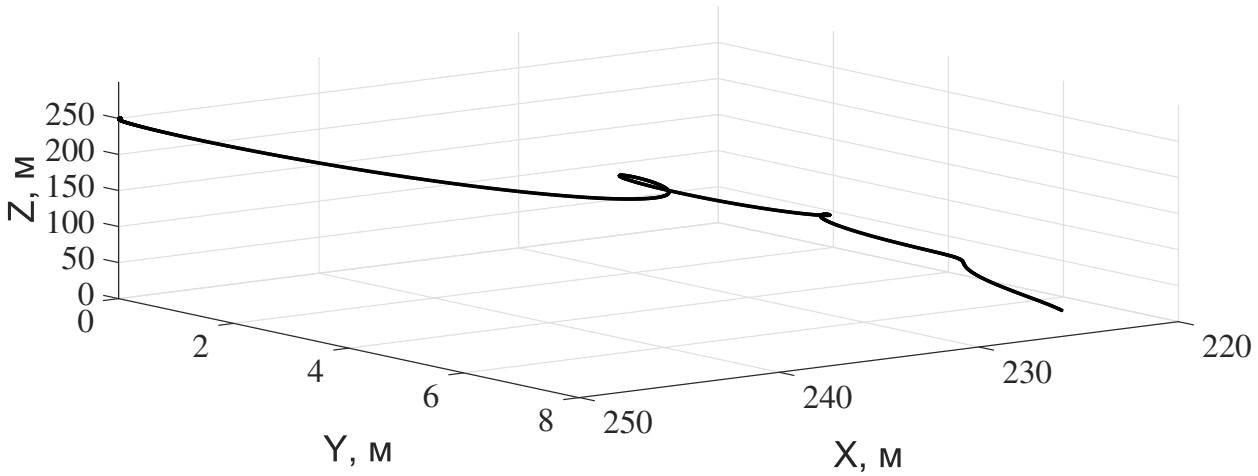


Рисунок 3.8 — Авария при снижении: траектория перемещения аппарата

Согласно полученным результатам, **можно сделать следующие выводы:**

- Потеря тяги хотя бы на одном винте квадрокоптера приводит к значительному и слабо прогнозируемому изменению его положения;
- При аварийной ситуации перед соприкосновением с поверхностью земли квадрокоптер набирает значительную скорость. Целесообразно полагать, что данное обстоятельство влечёт огромный ущерб как для аппарата, так и для полезной нагрузки. Наиболее вероятно, что после подобного падения устройство перестанет быть пригодным для дальнейшего использования (без основательной реконструкции).

3.2 Моделирование аварийных ситуаций с использованием спасательного алгоритма

3.2.1 Моделирование аварийных движений квадрокоптера с алгоритмом «ручной» посадки

Рассмотрим моделирование алгоритма, предполагающего автоматическое переключение из нормального режима управления в аварийный и последующее «ручное» управление высотой аппарата для его посадки. Для этого приведем подробную стратегию управления двигателями аппарата в соответствии с п. 2.2. Как было отмечено ранее, наличие признаков неисправности аппарата может быть определено полетным контроллером с высокой точностью не более чем за $0{,}5$ секунды. Пусть на 5-й секунде полета возникает аварийная ситуация — полный отказ второго двигателя ($\Omega_2 = 0$ об/с при $t > 5$ с). Алгоритм «ручной» посадки в данной ситуации предполагает следующую последовательность действий:

1-й шаг: Согласно п. 2.2, в момент идентификации неисправности управление аппаратом переходит в «ручной» аварийный режим, а также происходит информирование оператора БПЛА через интерфейс пульта управления. Сразу после обнаружения дефектного мотора (в данном примере при $t = 5{,}5$ с) принудительно отключается ротор, располагающийся по диагонали с вышедшим из строя винтом ($\Omega_4 = 0$ об/с, см. рис 3.11);

2-й шаг: Совместно с действиями 1 шага (в момент отключения моторов поврежденной диагональной пары) увеличивается тяга работающих двигателей на второй диагонали. Результирующее значение тяги работающей пары винтов должно приближаться к совокупной тяге аппарата для поддержания висения (значение $\Omega_1 = \Omega_3 = 400$ об/с, см. рис. 3.11);

3-й шаг: Управление тягой на двух работающих винтах с измененным нулевым положением ручки управления (относительно необходимой тяги для висения с учётом шагов 1 и 2) переходит к оператору;

4-й шаг: На основе доступных средств контроля оператор БПЛА обеспечивает управление высотой аппарата для достижения безопасной посадки. В рамках данного примера на интервале $t \in [5{,}5;\ 11{,}7]$ проводится уменьшение вертикальной скорости. По мере приближения аппарата к поверхности земли осуществляется снижение вертикальной скорости до нуля.

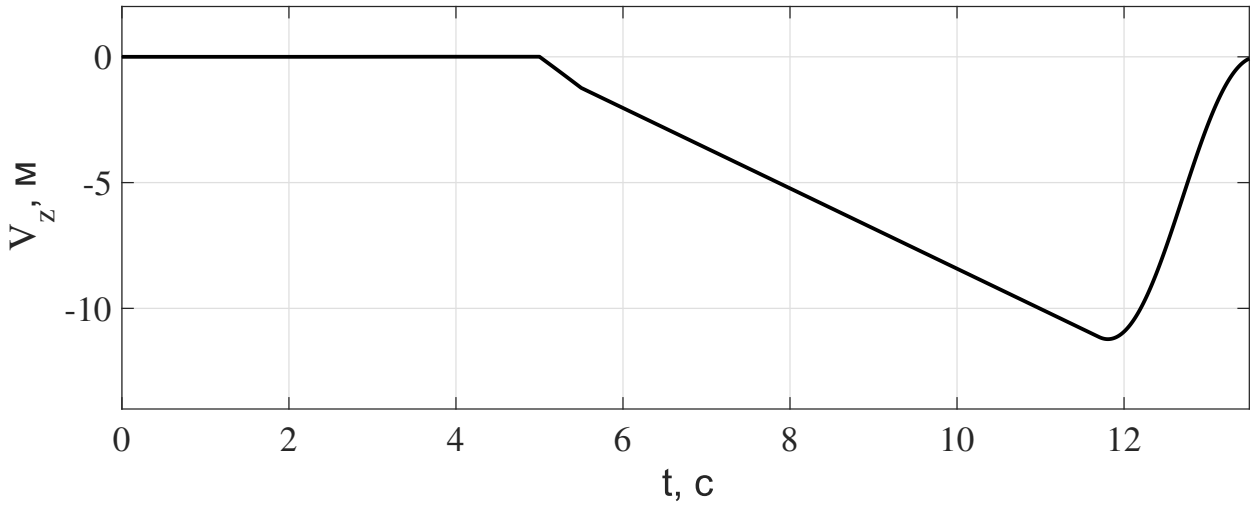


Рисунок 3.9 — «Ручной» спасательный алгоритм: скорость V_z

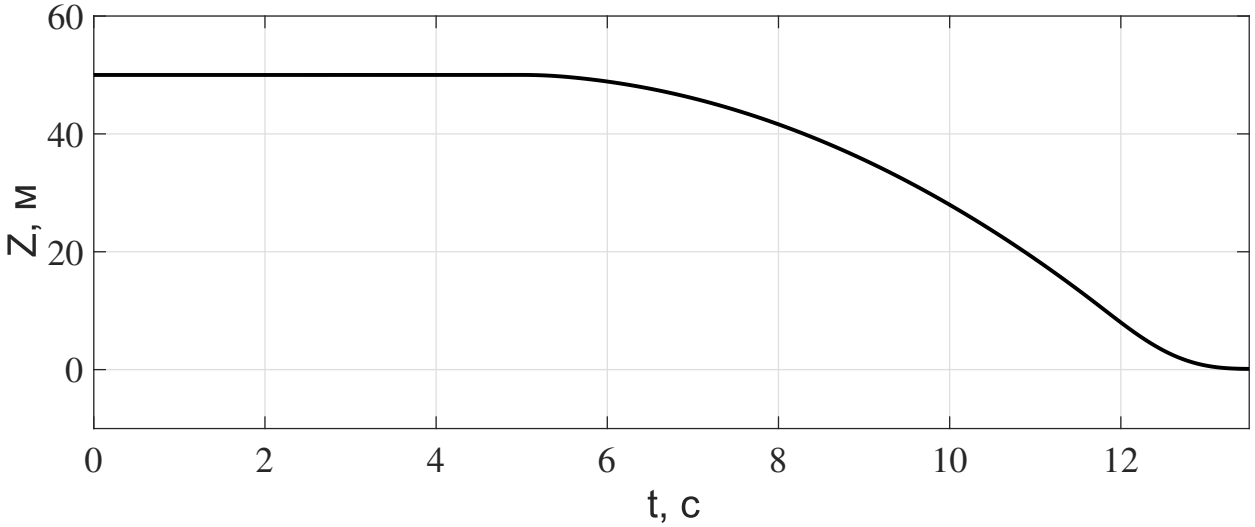


Рисунок 3.10 — «Ручной» спасательный алгоритм: высота Z

В результате получим следующую траекторию перемещения квадрокоптера (см. рис. 3.12). Значения угловых скоростей Ω_i, скорости V_z и высоты Z показаны на графиках 3.11, 3.9 и 3.10 соответственно.

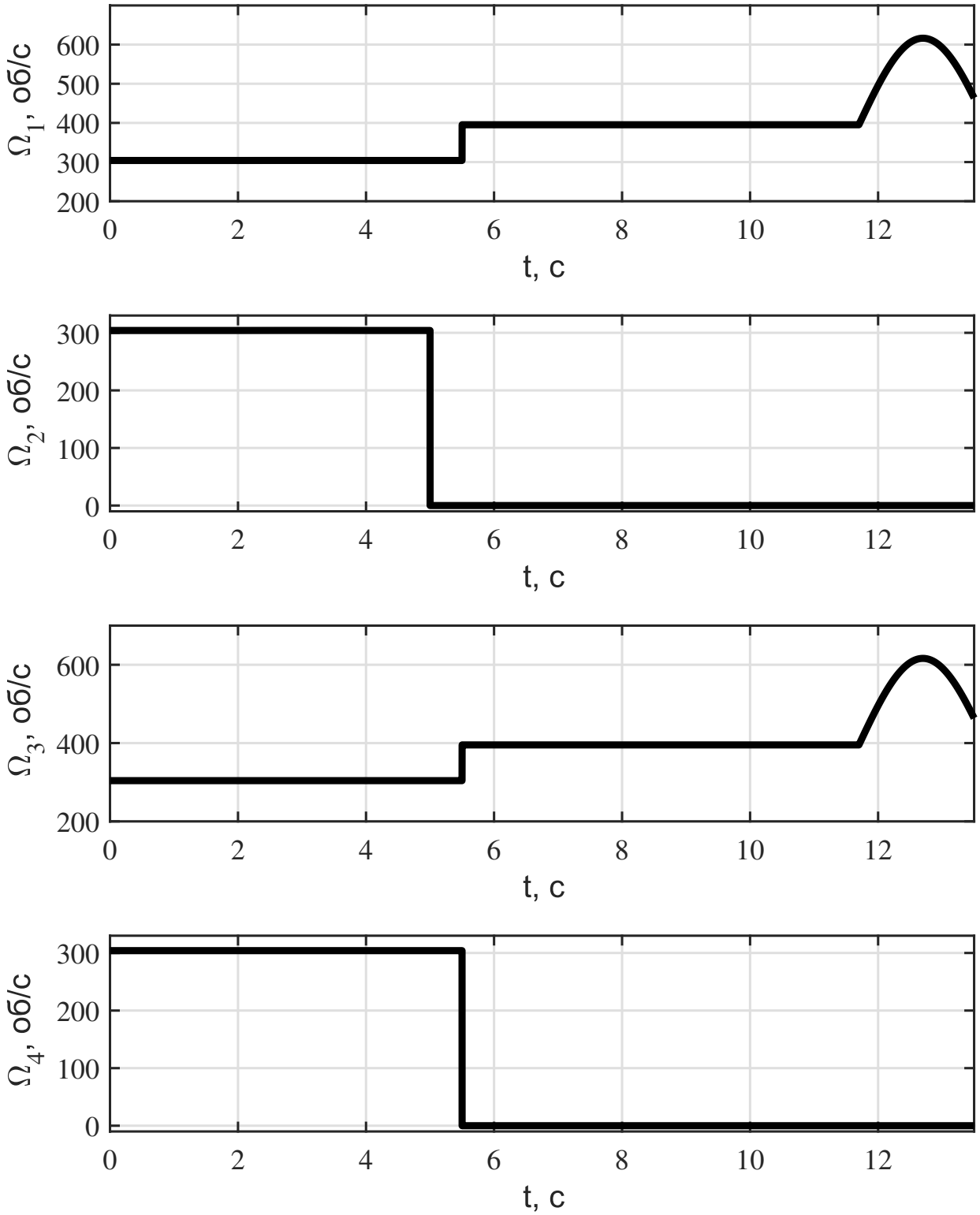


Рисунок 3.11 — «Ручной» спасательный алгоритм: угловые скорости Ω_i

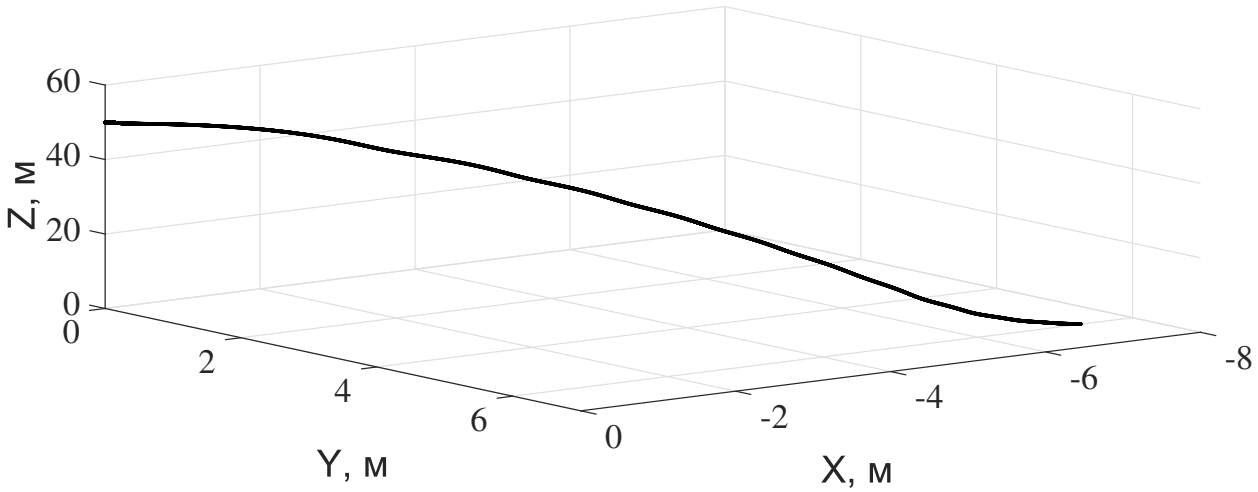


Рисунок 3.12 — «Ручной» спасательный алгоритм: траектория перемещения аппарата

Ключевым условием для осуществления безопасной посадки является гашение скорости приземления аппарата по мере приближения к земле ($V_z \to 0$ при $Z \to 0$). Действуя по указанному алгоритму, можно свести риск повреждения полезной нагрузки аппарата к минимуму.

3.2.2 Моделирование аварийных движений квадрокоптера с использованием ПИД контроллера

Рассмотрим автоматизированный подход к безопасной посадке квадрокоптера, основанный на соображениях из п. 2.3. Предположим, что при перемещении аппарата в горизонтальной плоскости в момент времени $t^* = 5$ с возникает аварийная ситуация — Ω_2 падает до нуля из-за поломки винта (см. рис. 3.13).

В соответствии с п. 2.2, за $0{,}5$ с происходит идентификация аварийной ситуации с последующим автоматическим переключением из нормального режима управления в аварийный с использованием ПИД регулятора: $[K_p,\, K_i,\, K_d] = [50,\, 8,\, 20]$; высота перехода на участок торможения $z_0 = 5$ м.

Скорость вращения рабочих винтов контролируется двумя ПИД-регуляторами в соответствии с п. 2.3. Первый обеспечивает заданную вертикальную скорость (-1 м/с) на интервале $t \in [5.5, 29.3]$, пока аппарат

находится на высоте $Z > 5$ м (см. рис. 3.14). Далее происходит переключение на ПИД-регулятор, сводящий в 0 скорость снижения аппарата при достижении поверхности земли.

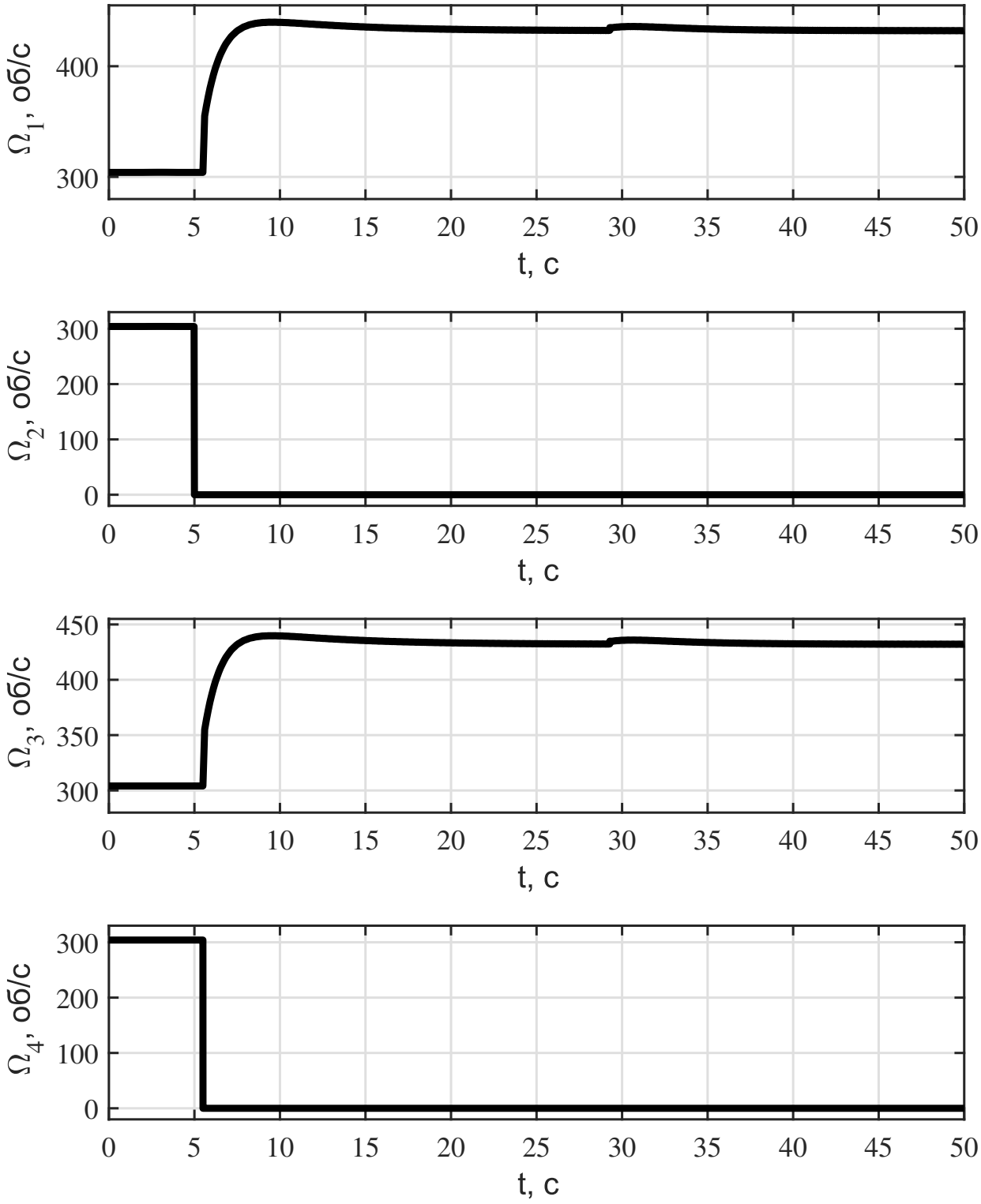


Рисунок 3.13 — Автоматизированный спасательный алгоритм: угловые скорости Ω_i

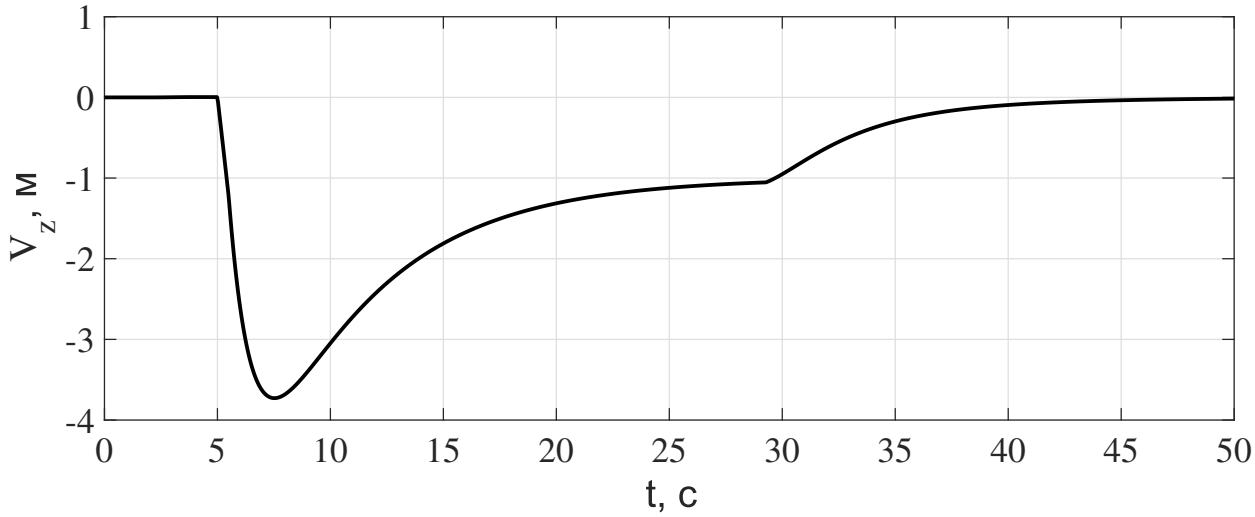


Рисунок 3.14 — Автоматизированный спасательный алгоритм: скорость V_z

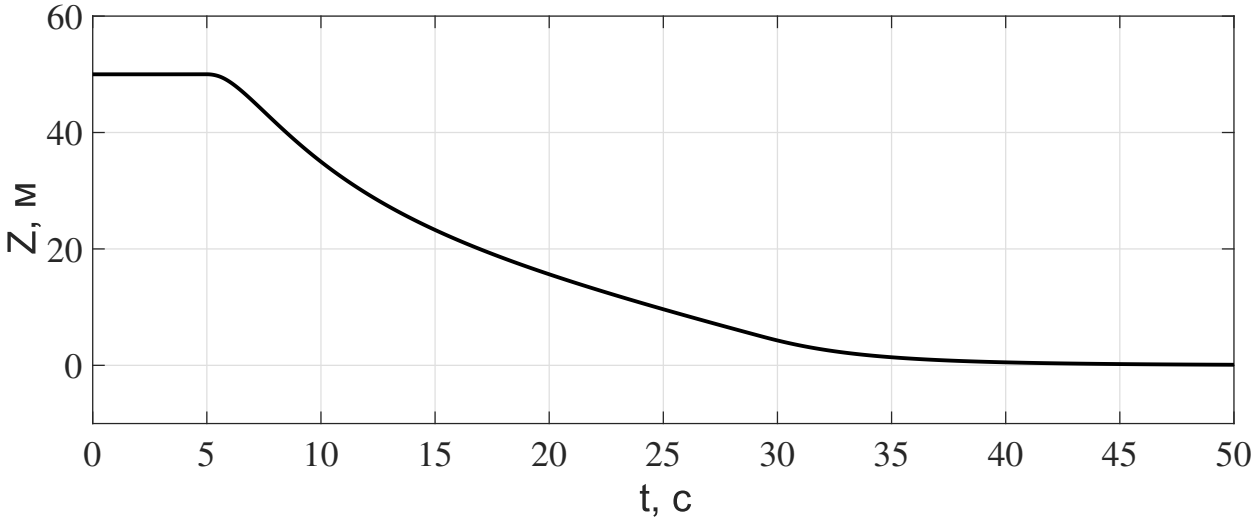


Рисунок 3.15 — Автоматизированный спасательный алгоритм: высота Z

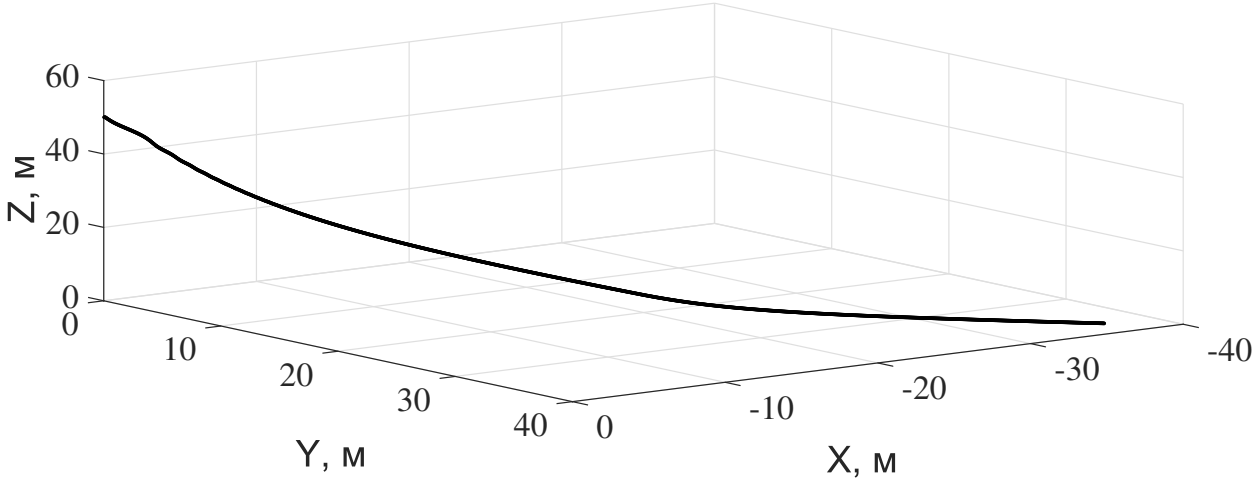


Рисунок 3.16 — Автоматизированный спасательный алгоритм: траектория перемещения аппарата

В результате мы имеем безопасную посадку (см. рис. 3.15) с нулевой вертикальной скоростью при $Z = 0$. При $t = 50$ с и $Z = 0.09$ м, имеем $V_z = -0.01$ м (см. рис. 3.14). График траектории перемещения аппарата представлен на рис. 3.16.

Следует отметить, что работа алгоритма была бы невозможна, если бы не были учтены рекомендации по конфигурации квадрокоптера (см. табл. 1).

3.3 Эксперименты по оценке расстояния приземления БПЛА в аварийной ситуации

Данный раздел призван продемонстрировать зависимость траектории перемещения аппарата от величины неисправности одного из роторов аппарата. Важно понимать, что автоматическое управление помимо преимуществ от безукоризненной точности исполнения спасательной методологии имеет также и большую слабость в виде недопустимости отклонений от заранее заданной последовательности действий. Поэтому подобный «анализ в деталях» как аварийной ситуации, так и внедрение автоматического метода спасения, может пролить свет на зависимость динамики перемещения квадрокоптера от величины повреждённого винта (или его электрической части) Ω_i^{em} и времени начала действия алгоритма безопасной посадки.

В начале проведем ряд моделирований аварийных ситуаций, взяв в качестве начального положения аппарата $(x_0, y_0, z_0) = (0, 0, 50)$. Будем рассматривать следующие значениях угловой скорости повреждённого винта Ω_2:

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

Значение $\Omega_2^{em_{100}} = 300$ об/с соответствует слабому повреждению пропеллера, так как необходимое значение угловой скорости каждого винта для поддержания состояния покоя (в рамках данной модели квадрокоптера с параметрами (1.5)) $\Omega_{hovering} = 304.1691$ об/с.

Таким образом, получим следующие результаты (рис. 3.17). Каждая кривая соответствует траектории падения аппарата в i-й аварийной ситуации. Для удобства анализа полученных данных оставим точку приземления аппарата в каждом отдельном случае. Получим следующий график (рис. 3.18).

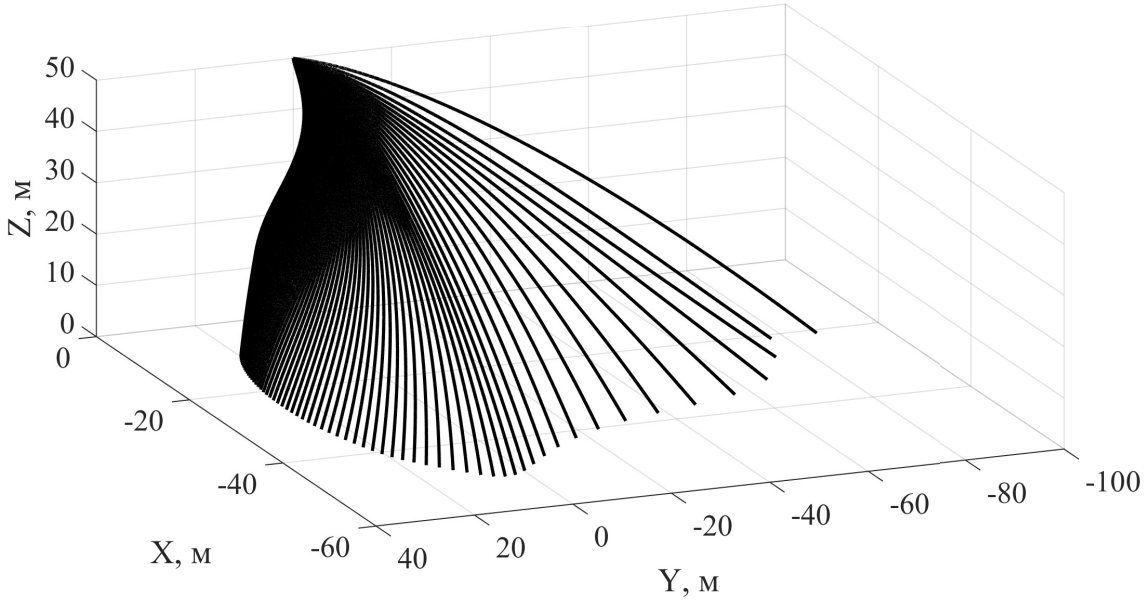


Рисунок 3.17 — Множество траекторий посадки аппарата при разных значениях Ω_2^{em}

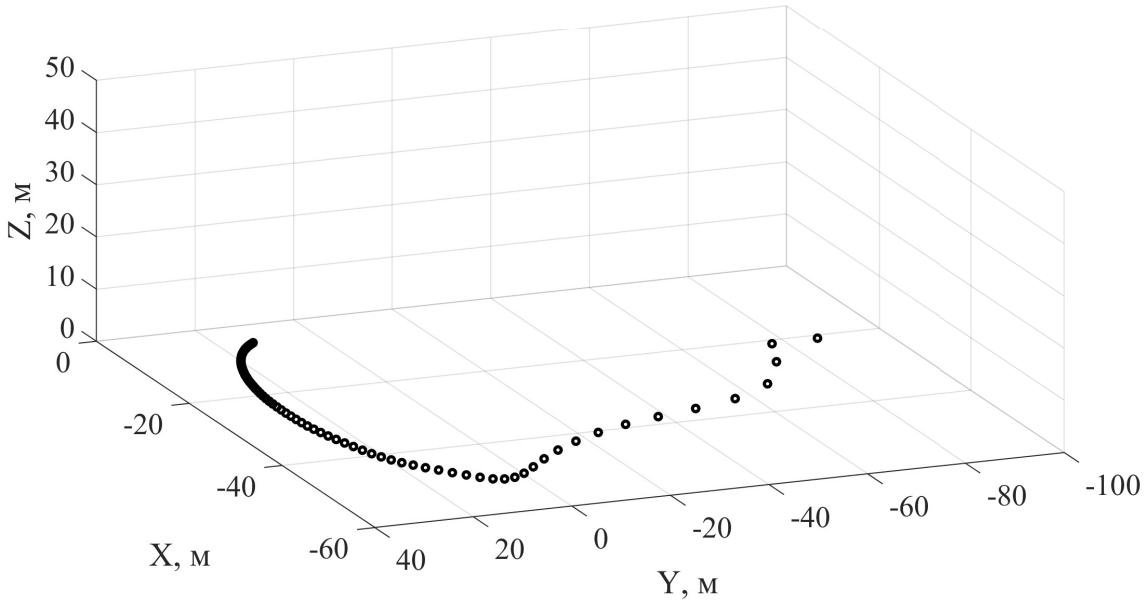


Рисунок 3.18 — Множество точек приземления аппарата при разных значениях Ω_2^{em}

Исходя из полученных результатов можно сделать вывод, что малые значения $\Omega_2^{em_i}$ не сильно влияют на расстояние от точки приземления аппарата до начального положения, в то время как между точками со значениями угловой скорости, соответствующими слабому повреждению аппарата, изменение расстояния заметно больше. Это можно объяснить тем, что при слабом нарушении работоспособности винта квадрокоптер длительное время не достигает критического наклона, ведущего к бесконтрольному падению. Иными словами, чем меньше поврежден винт, тем дальше пролетит аппарат прежде чем упадёт на землю.

Теперь исследуем воздействие алгоритма отказоустойчивого приземления квадрокоптера (п. 3.2.2) в тех же условиях моделирования, введя в рассмотрение дополнительный, измеряемый в секундах, параметр $t_{reaction} = 0{,}5 + 0{,}1i;\ i = \overline{0{,}5}$.

Данная переменная несёт следующий практический смысл: время обнаружения неисправности системой с последующим переключением в аварийный режим. Случаи с отложенной реакцией ($t_{reaction} > 5$ с) представляют больший интерес, чем ситуации с ($t_{reaction} < 5$ с), так как ранее уже была продемонстрирована эффективность отказоустойчивого алгоритма с задержкой в $0{,}5$ с [3] и, кроме этого, для более скорой реакции БПЛА целесообразно ожидать повышение требований к конфигурации самого аппарата, что не имеет смысла в рамках данной математической модели. Осуществление более позднего запуска спасательного алгоритма, вопреки вышесказанному, не зависит от устройства напрямую — можно программно определить необходимый момент времени.

Важно заметить, что в данной работе метод аварийной посадки не полностью обнуляет скорости повреждённого и симметричного ему винтов, а сохраняет угловые скорости обоих пропеллеров одинаковыми. В данном случае на время действия отказоустойчивого алгоритма $\Omega_4 = \Omega_2^{em_i}$, что устраняет наклон аппарата и позволяет сохранить некоторую тягу, облегчая работу по посадке полностью исправной паре винтов.

Таким образом можно исследовать зависимость перемещения квадрокоптера от времени начала действия спасательного алгоритма после возникновения аварии (рис. 3.19). Как и в предыдущем эксперименте, для

удобства анализа полученных данных оставим точку приземления аппарата в каждом отдельном случае. Получим следующий график (рис. 3.20).

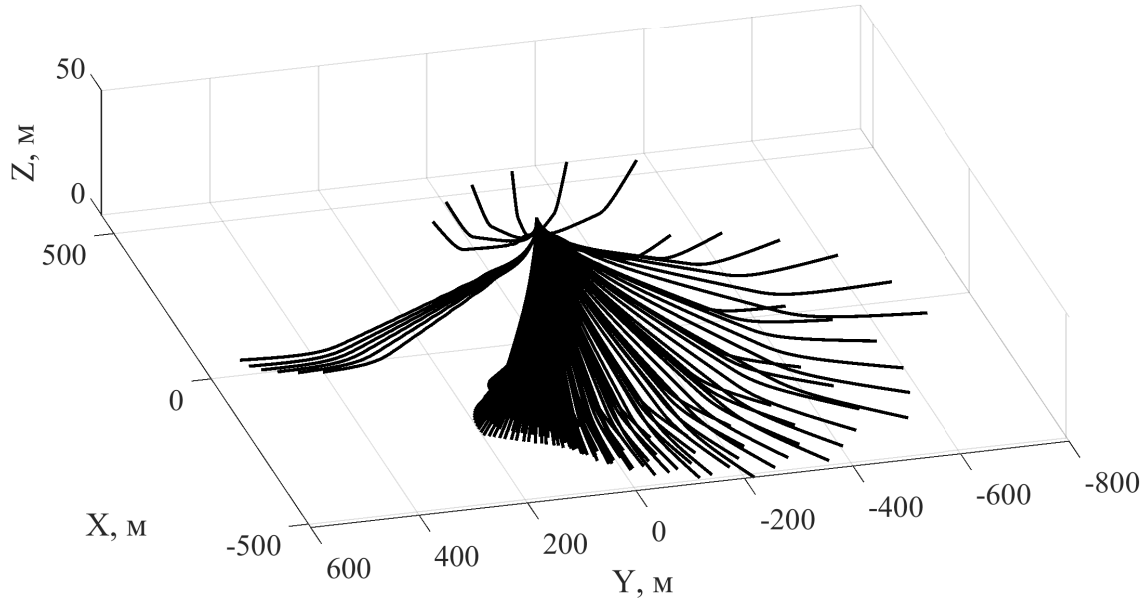


Рисунок 3.19 — Множество траекторий посадки аппарата при разных $\Omega_2^{em}, t_{reaction}$

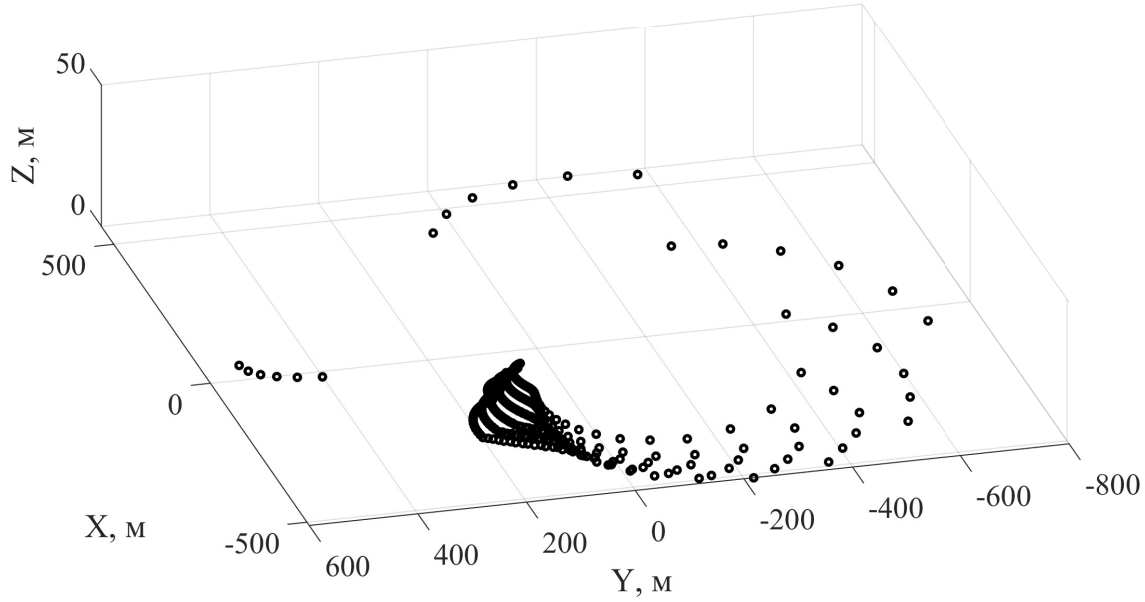


Рисунок 3.20 — Множество точек приземления аппарата при разных $\Omega_2^{em}, t_{reaction}$

На графике видно, что при работе отказоустойчивого метода аппарат летит тем дальше, чем больше значение $\Omega_2^{em_i}$. Тем не менее, в сравнении с предыдущим экспериментом, введение параметра $t_{reaction}$ усложнило визуальный анализ. В связи с этим сгруппируем полученное множество точек

по времени начала работы алгоритма посадки после возникновения аварии $t_{reaction}$ (см. рис. 3.21).

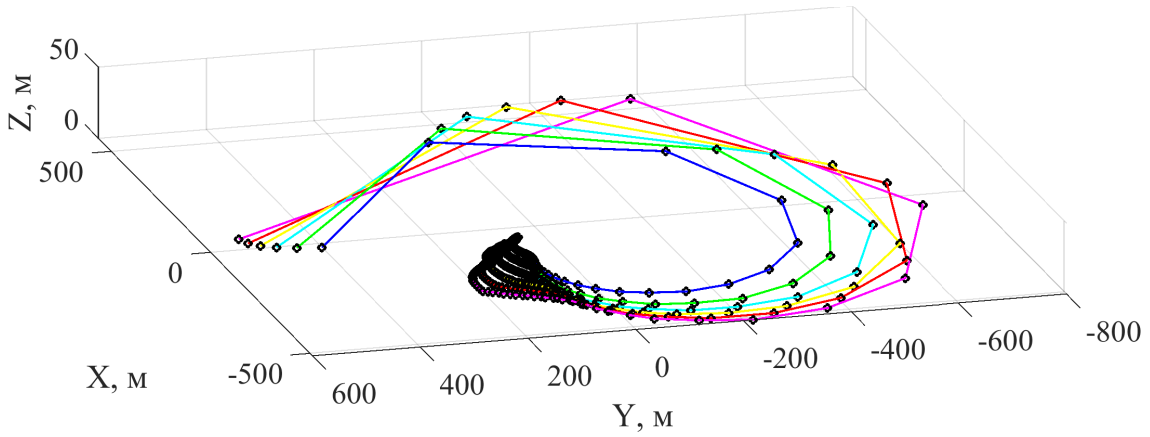


Рисунок 3.21 — Множество точек приземления аппарата при разных $\Omega_2^{em}, t_{reaction}$ (группировка по $t_{reaction}$

Исходя из полученных результатов можно сделать вывод, что время начала действия спасательного алгоритма в некоторой степени влияет на направление посадки квадрокоптера — множество точек приземления образует «витки», в которых каждая точка соответствует определенному времени переключения на спасательный алгоритм.

Таким образом, данный подход можно использовать в следующих направлениях:

1. При возникновении аварийной ситуации можно определить область падения аппарата или, в случае подключения отказоустойчивого алгоритма, определить примерную дальность и направление посадки, что позволяет избежать перемещение в нерекомендуемую для посадки зону, варьируя время запуска программы после потери тяги на одном из винтов.
2. Более подробное изучение динамики квадрокоптера может позволить на практике осуществить контроль в автоматическом режиме над дальностью и направлением полёта неисправного аппарата.

Иными словами, полученные данные потенциально могут быть использованы для усовершенствования работы алгоритмов посадки [2; 3], продемонстрированных в п. 3.2.

3.4 Выводы из результатов моделирования

Данная глава демонстрирует моделирование лишь частных случаев аварийных ситуаций. Однако важно заметить, что внештатные ситуации с неисправностями любого рода фактически влекут одинаковые последствия — потеря контроля над квадрокоптером, неизбежный переворот аппарата с последующим падением.

По-настоящему «частным» случаем является «ручное» спасение аппарата, так как результирующие действия осуществляются в рамках определённых обстоятельств и только могут быть в некоторой степени обобщены.

Автоматизированный алгоритм посадки лишен данного недостатка в виду полного программного контроля при любой аварии, последствия которой позволяют управлять двумя диагональными винтами. Тем не менее, это также является и проблемой данного подхода — фиксированные заранее параметры (желаемая скорость снижения, высота перехода на «гасящий» скорость приземления режим, время начала работы) зачастую определяют как время посадки, так и расстояние, которое аппарат пролетит перед достижением поверхности земли. Данные обстоятельства не позволяют получить соответствующую реакцию алгоритма на требования или изменения внешней среды (ограниченная область посадки, оставшаяся величина заряда аккумуляторов).

Решать указанные проблемы автоматизированного ПО предлагается усложнением спасательного алгоритма, однако эта рекомендация не сможет покрыть весь спектр возможных ситуаций без использования просчёта дальнейшей траектории аппарата в каждый момент времени.

Выводы

Данные, полученные в численных экспериментах с использованием математической модели квадрокоптера, дают ясное представление о поведении БПЛА определенной конфигурации, заданной параметрически в терминах моделирования аппарата. Используя описанные в работе методологии и принципы синтеза аварийных ситуаций, а также алгоритмы, направленные на минимизацию негативных последствий внештатных случаев, можно добиться конструктивного и при том автоматического контроля аппарата в условиях исправности диагональной пары винтов. Однако важно заметить, что методика автоматизированного спасательного алгоритма нуждается в более подробном анализе для нахождения путей устранения недостатков, описанных в п. 3.4.

Таким образом, использование ПО на основе описанной теории может существенно упростить решение задач, требующих проведения тестов на конкретном аппарате, в виду уменьшения временных и денежных затрат. Созданный пакет программ снабжен описанием задаваемых параметров, а также функциями, автоматизирующими процессы моделирования и обработку полученных результатов (см. приложения А и Б).

Заключение

Основные результаты работы заключаются в следующем:

1. Описана математическая модель квадрокоптера и определены параметры, соответствующие реальной конфигурации аппарата (п. 1.1 и 1.2);
2. С точки зрения математической модели определены аварийная ситуация, спасательная методология и метод оценки ожидаемой области падения (приземления) квадрокоптера с учётом определения возможных перемещений аппарата (п. 2.1 — 2.4);
3. Проведено численное моделирования различных аварийных ситуаций (п. 3.1);
4. Осуществлено внедрение методологии спасательного приземления в математическую модель квадрокоптера (п. 3.2);
5. Поставлены эксперименты, демонстрирующие возможности подхода к оценке ожидаемой области падения (п. 3.3);
6. Осуществлён анализ полученных результатов (п. 3.4);
7. Для реализации поставленных целей и задач было разработано ПО на основе программного пакета MATLAB R2019b с дополнением Simulink v.10.0 (см. приложения А и Б).

Основные результаты докладывались и обсуждались на конференциях «Процессы управления и устойчивость» (2019 и 2021 гг.) [1], «13 th International Symposium on Intelligent Distributed Computing (IDC)» (2019 год) [2]; опубликованы в журнале «Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)» (Том 11, №2) [3].

Список литературы

1. Жолобов Е. В., Баранов О. В. Моделирование и анализ аварийных движений квадрокоптера // Процессы управления и устойчивость. — 2019. — Т. 6, № 1. — С. 213–217.

2. 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.

3. 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.

4. 16 способов применения квадрокоптеров: спасение жизни, времени и денег [Электронный ресурс]. — URL: https://www.moyo.ua/news/16_unikalnyh_sposobov_primeneniya_kvadrokopterov.html (дата обращения: 29.03.2021).

5. Заказать съемку с квадрокоптера - цены на услууги [Электронный ресурс]. — URL: https://4vision.ru/aerosemka.html (дата обращения: 07.04.2021).

6. Сферы применения дронов, квадрокоптеров, БПЛА, готовые решения [Электронный ресурс]. — URL: https://brlab.ru/scopes/ (дата обращения: 01.04.2021).

7. Доставка дронами: кто будет первым? [Электронный ресурс]. — URL: https://dronomania.ru/faq/dostavka-dronami-kto-budet-pervym.html (дата обращения: 01.04.2021).

8. Как дроны и нейросети ищут пропавших людей? И насколько они реально эффективны? [Электронный ресурс]. — URL: https://tass.ru/obschestvo/6757981 (дата обращения: 14.01.2021).

9. «Укроборонпро» представил первый отечественный разведывательный квадрокоптер для военных «Берегиня» [Электронный ресурс]. — URL: https://itc.ua/news/ukroboronprom-predstavil-pervyj-otechestvennyj-razvedyvatelnyj-kvadrokopter-dlya-voennyh-bereginya-video/ (дата обращения: 14.01.2021).

10. В Китае начали использовать дроны с огнеметами [Электронный ресурс]. — URL: https://www.ixbt.com/news/2017/02/21/v-kitae-nachali-ispolzovat-drony-s-ognemetami-.html (дата обращения: 15.01.2021).

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

12. Verdan Sikiric. Control of Quadrocopter : Master's thesis / Sikiric Verdan ; School of Computer Science and Communication. — Stockholm, Sweden, 2008. — 88 pp.

13. Weibel Roland E., Hansman John R. Safety consideration for operation of unmanned aerial vehicles in the national airspace system : Ph. D. thesis / Roland E. Weibel, John R. Hansman ; Massachusetts Institute of Technology. — USA, 2005. — march.

14. Tsach S., Penn D., Levy A. Advanced technologies and approaches for next generation uavs // International congress of aeronautical sciences. — 2002.

15. Allison Sam, Bai he, Jayaraman Balaji. Wind estimation using quadcopter motion: A machine learning approach // Aerospace Science and Technology. — 2020. — 03. — Vol. 98. — P. 105699. DOI: 10.1016/j.ast.2020.105699.

16. Radiansyah S, Kusrini Mirza, Prasetyo Lilik. Quadcopter applications for wildlife monitoring // IOP Conference Series: Earth and Environmental

Science. — 2017. — 01. — Vol. 54. — P. 012066. DOI: 10.1088/1755-1315/54/1/012066.

17. Альсевич В. В., Габасов Р., Глушенков В. С. Оптимизация линейных экономических моделей: статические задачи. — Минск : Изд-во БГУ, 2000. — 211 с.

18. Скляров А. А., Скляров С. А. Синергетический подход к управлению беспилотным летательным аппаратом в среде с внешними возмущениями // Изв. Южн. фед. ун-та. Технические науки. — 2012. — № 8. — С. 159–170.

19. Ефимов В. Программируем квадрокоптер на Arduino (ч. 1) [Электронный ресурс]. — URL: http://habrahabr.ru/post/227425/ (дата обращения: 16.02.2021).

20. Жмудь В. А., Заворин А. Н., Ядрышников О. Д. Неаналитические методы расчета ПИД-регуляторов : учебное пособие. — Изд-во НГТУ, 2013. — 39 с.

21. Зубов В. И. Лекции по теории управления. — М. : Наука, 1975. — 496 с.

22. Баранов О. В. Управление квадрокоптером в аварийных режимах функционирования // Вестн. С.-Петерб. ун-та. Прикладная математика. Информатика. Процессы управления. — 2016. — С. 69–79.

23. Купить комнатный квадрокоптер - Quadrone.ru [Электронный ресурс]. — URL: https://quadrone.ru/komnatnye/ (дата обращения: 18.04.2021).

24. Квадрокоптеры с большой грузоподъемностью – характеристики, особенности и устройство | digbox.ru [Электронный ресурс]. — URL: https://digbox.ru/reviews/s-bolshoy-gruzopodemnosty/ (дата обращения: 18.04.2021).

25. Корченко А. Г., Ильяш О. С. Обобщённая классификация беспилотных летательных аппаратов // Збірник наукових праць Харківського національного університету Повітряних Сил. — 2012. — Т. 4, № 33. — С. 27–36.

26. Попков А. С., Баранов О. В. Об оптимальном управлении вращательным движением вала электродвигателя // Процессы управления и устойчивость. — 2014. — Т. 1, № 17. — С. 31–36.

27. EHang запускает новое решение для тушения пожаров в небоскрёбах [Электронный ресурс]. — URL: https://dronomania.ru/news/ehang-zapuskaet-novoe-reshenie-dlya-tusheniya-pozharov-v-neboskryobah.html (дата обращения: 14.04.2021).

Приложение А

Структура математической модели в Simulink (MATLAB)

Математическая модель квадрокоптера состоит из двух главных блоков:

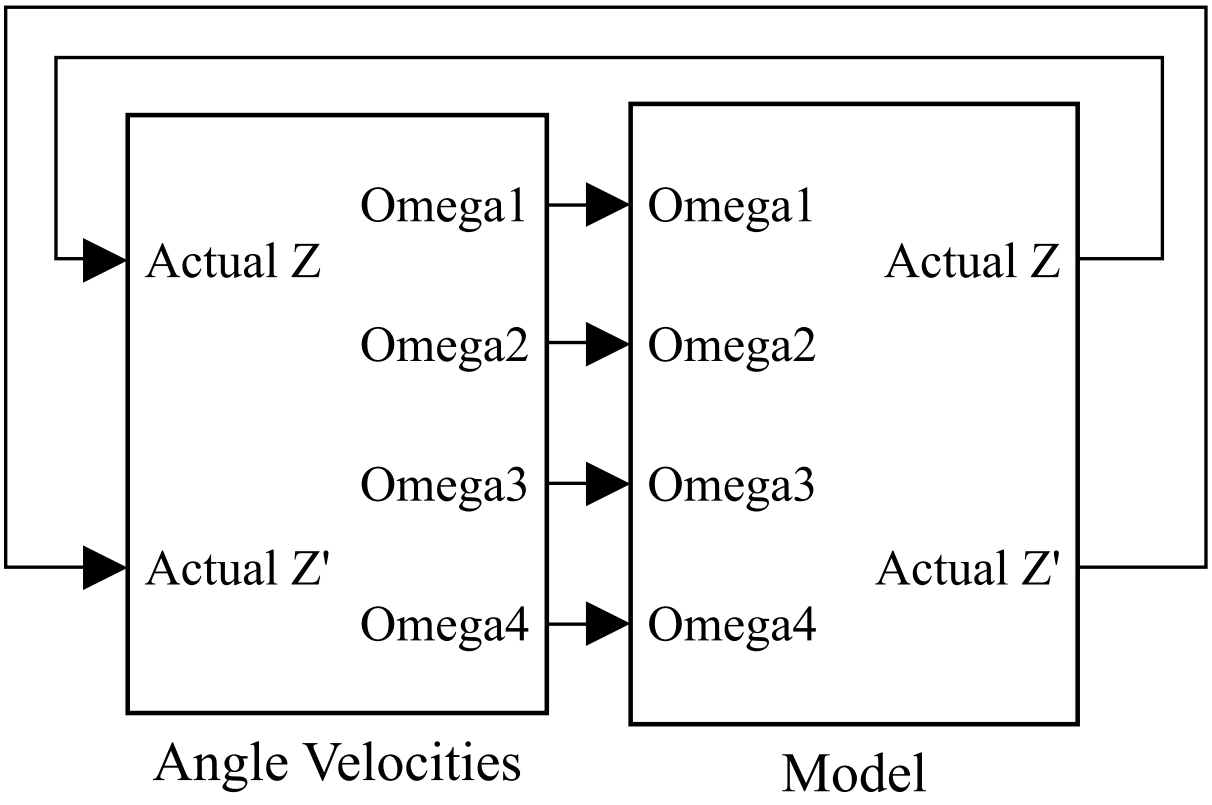


Рисунок А.1 — Структура модели в Simulink

А.1 Структура блока Angle Velocities

Внутренняя структура блока Angle Velocities разбита на несколько частей:

1. Симуляция аварии без включения спасательного алгоритма (красная область);

2. Симуляция аварии с внедрением отказоустойчивого алгоритма (зеленая область);
3. Приемные компоненты для передачи сигналов (бирюзовая область).

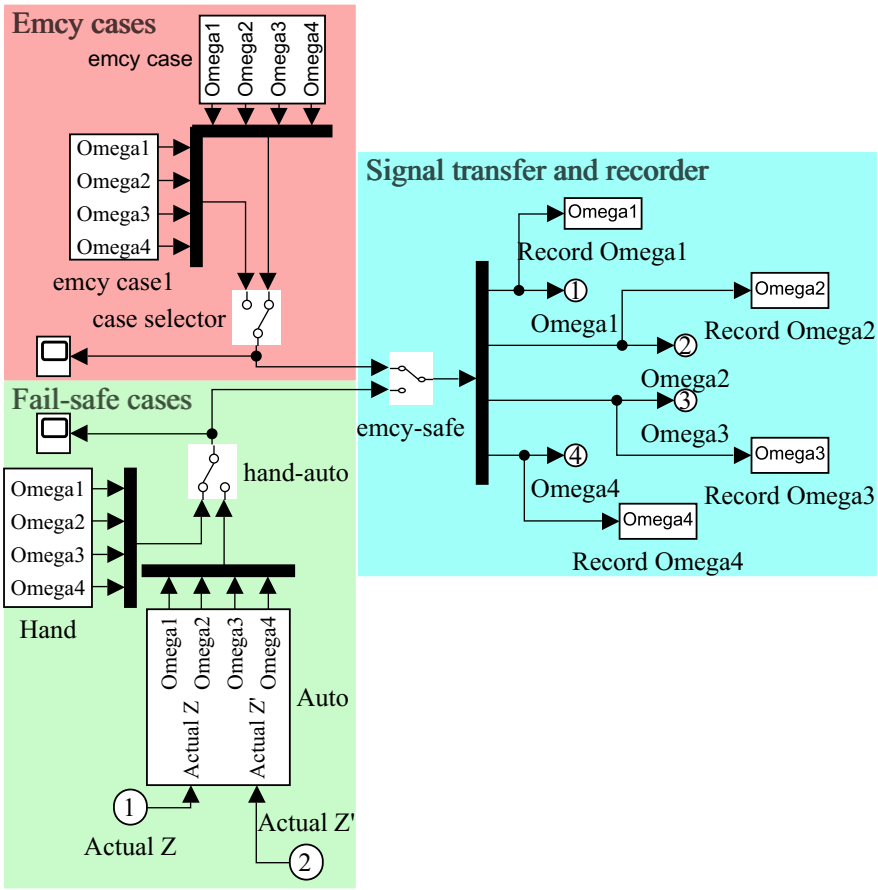


Рисунок А.2 — Структура блока Angle Velocities

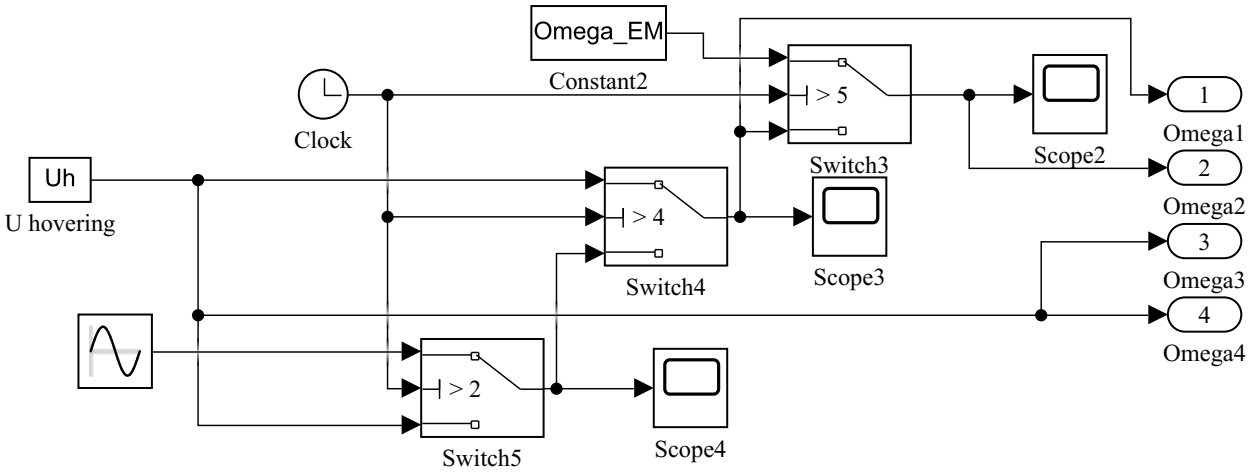


Рисунок А.3 — Структура внутренней системы emcy case (случай разных значений Ω^{em})

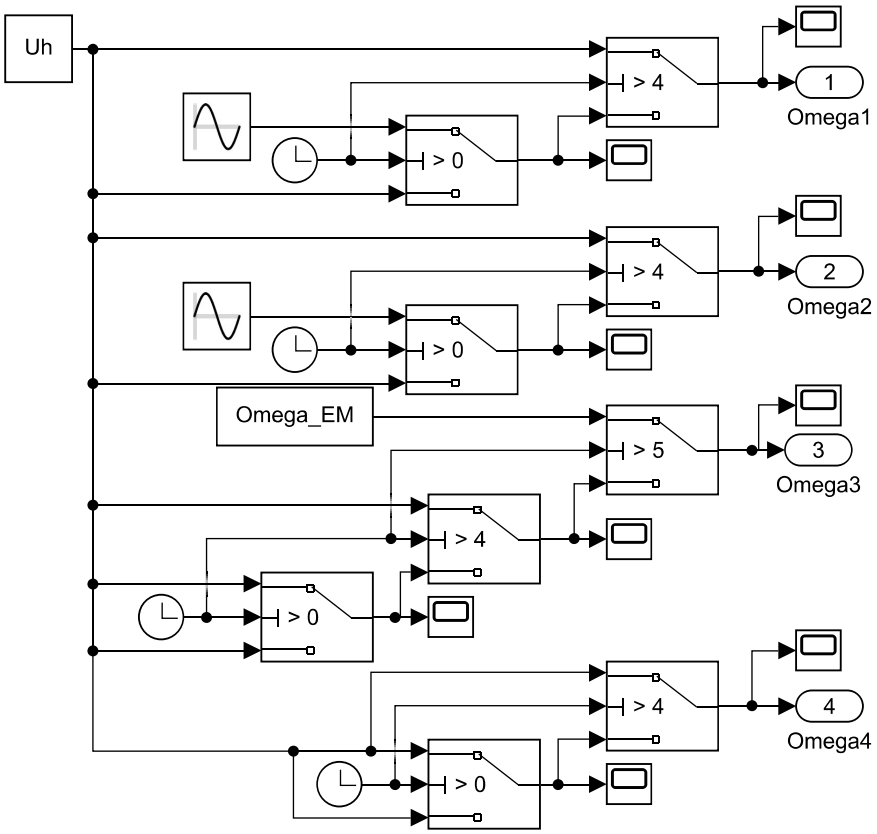


Рисунок А.4 — Структура внутренней системы emcy case 1 (случай отказа винта при горизонтальном движении)

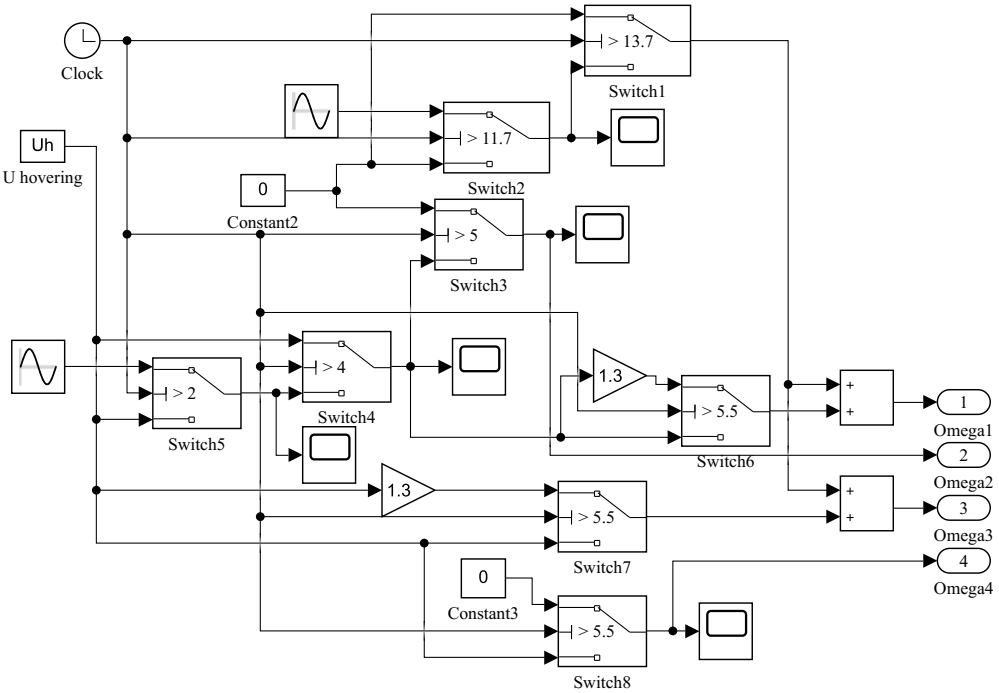


Рисунок А.5 — Структура внутренней системы Hand

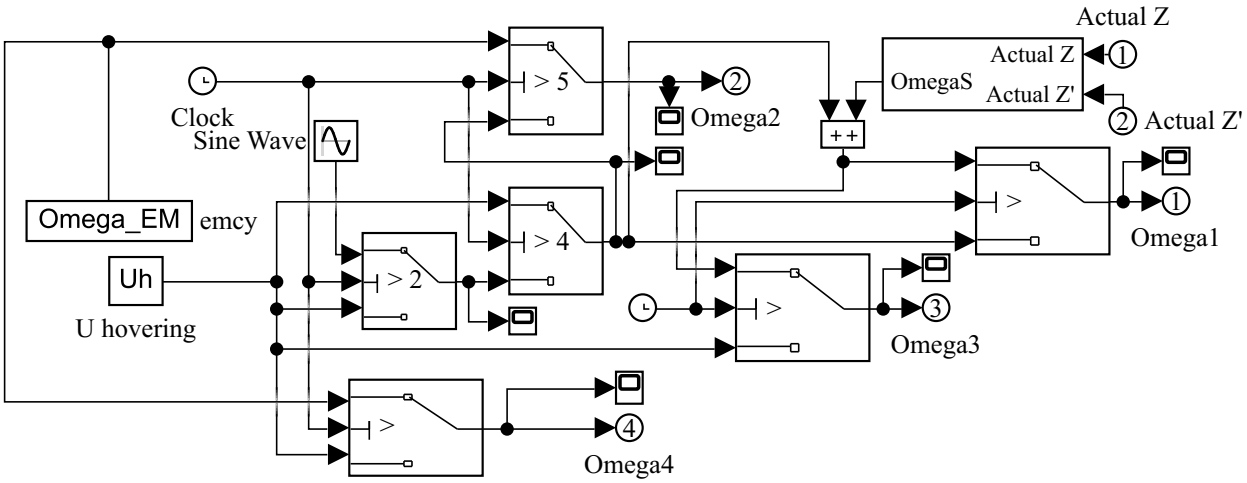


Рисунок А.6 — Структура внутренней системы Auto

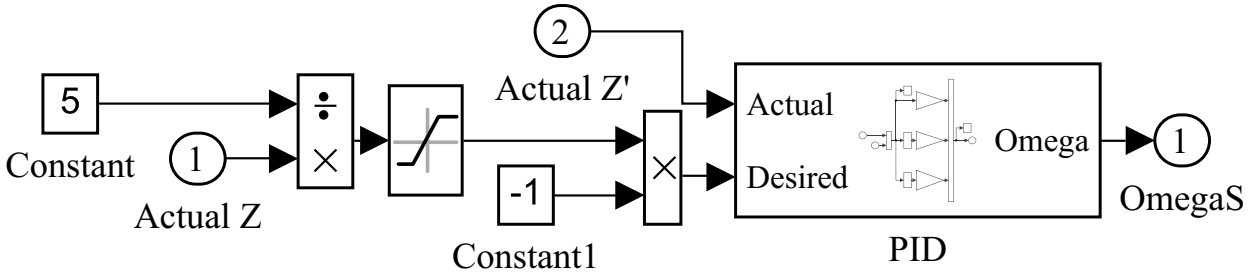


Рисунок А.7 — Структура системы ПИД контроллера в Auto

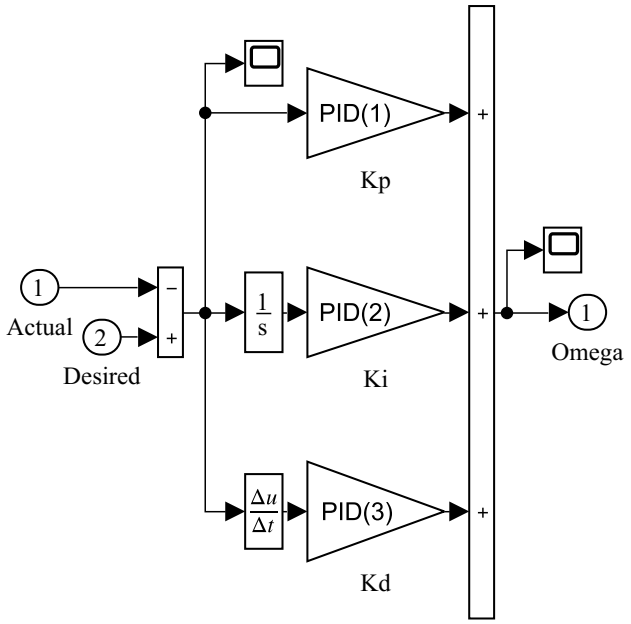


Рисунок А.8 — Структура системы ПИД контроллера в Auto

A.2 Структура блока Model

Блок Model содержит математическую модель квадрокоптера и блоки ввода/вывода данных:

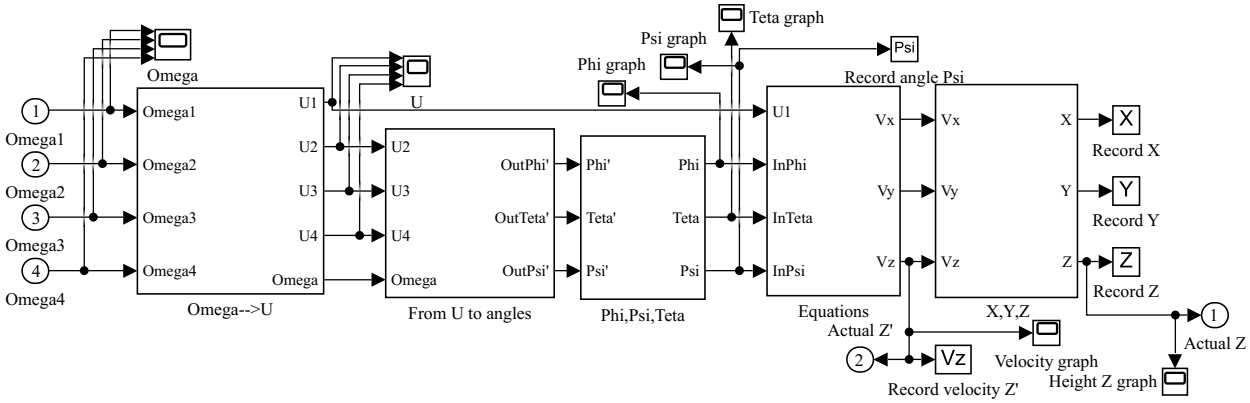


Рисунок А.9 — Структура блока Model

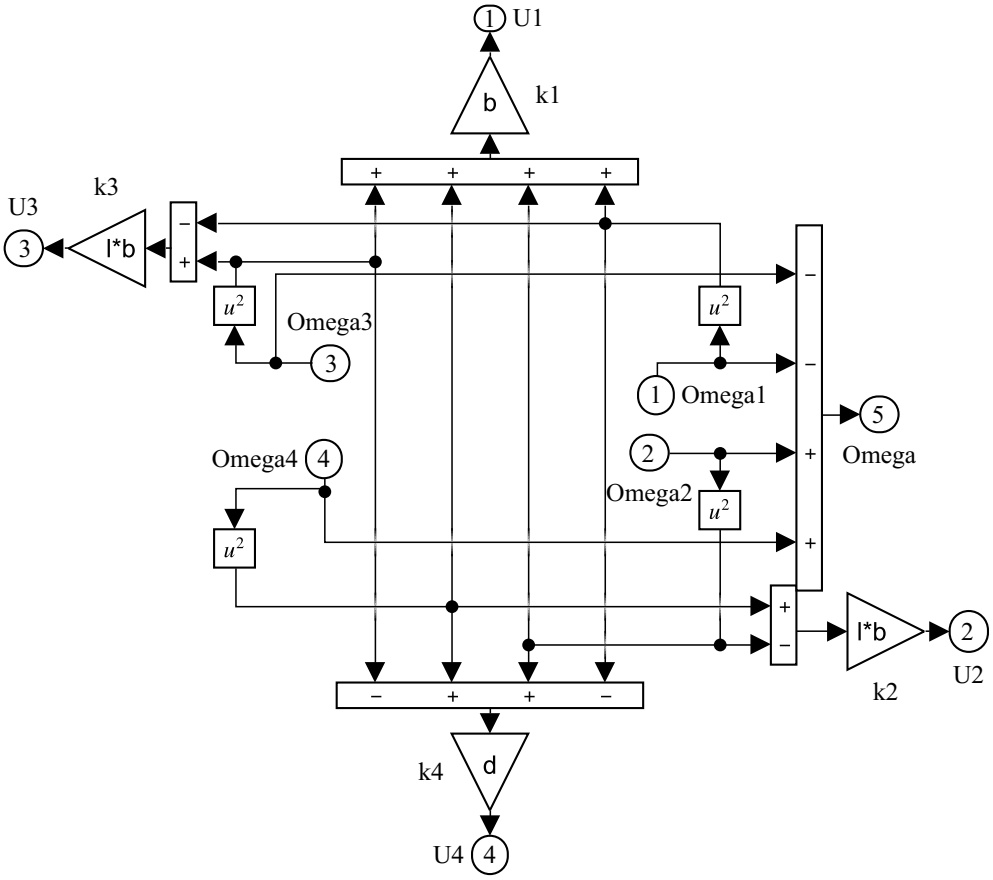


Рисунок А.10 — Структура блока Omega–>U. Реализация формулы 1.3

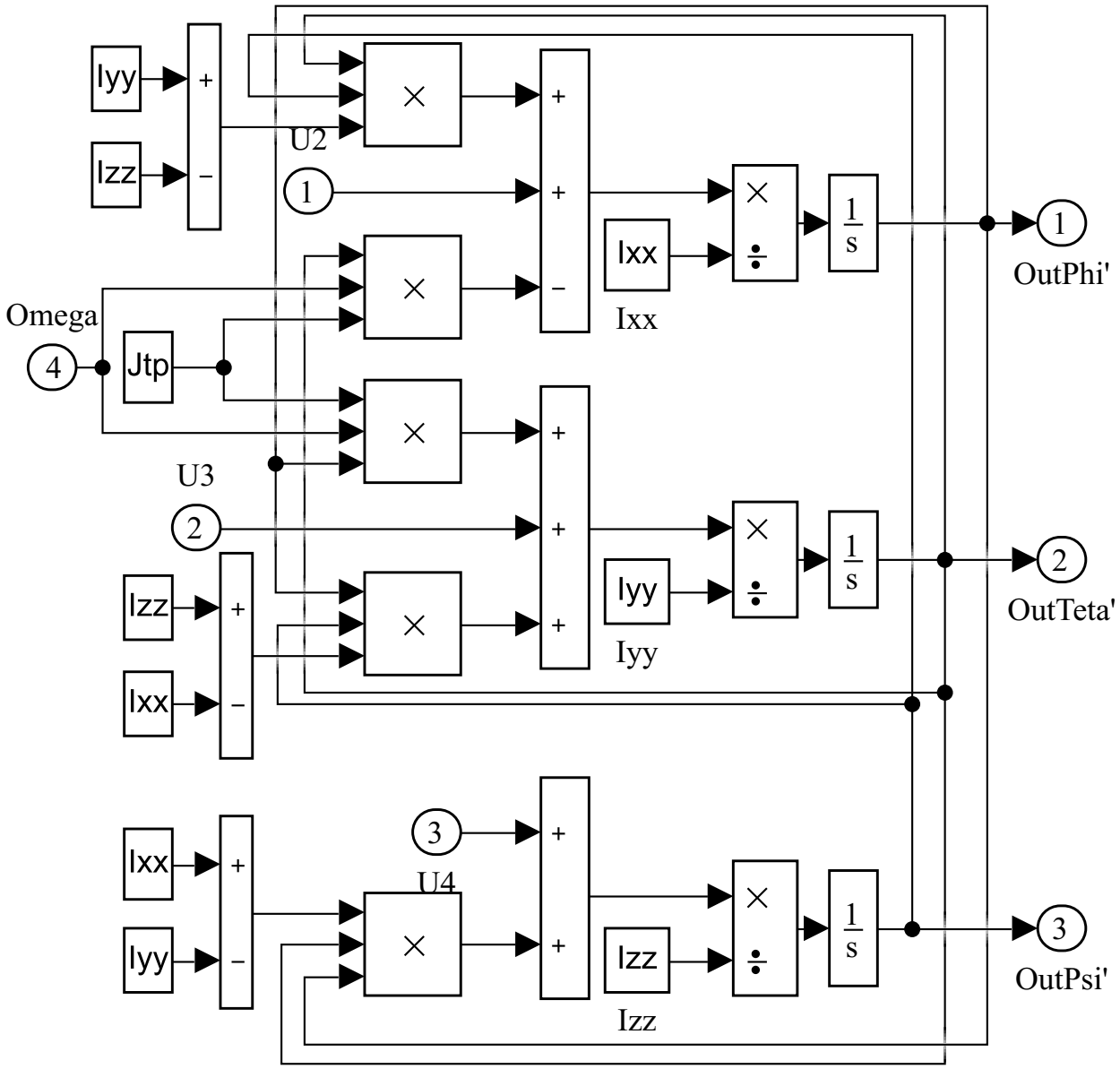


Рисунок А.11 — Структура блока From U to angles

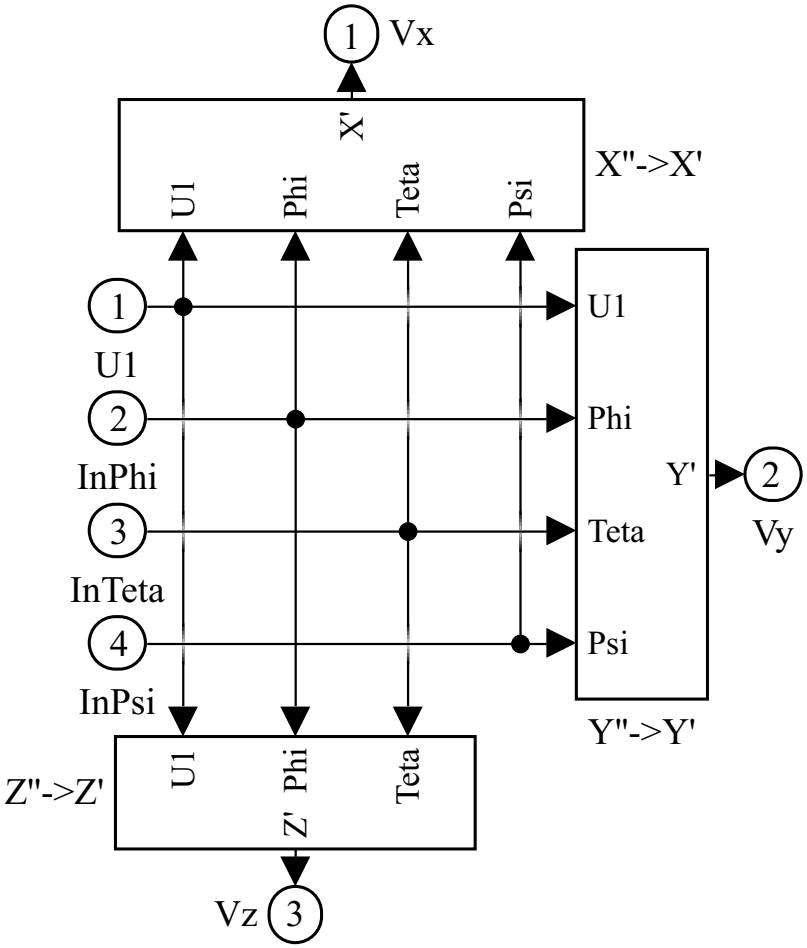


Рисунок А.12 — Структура блока Equations

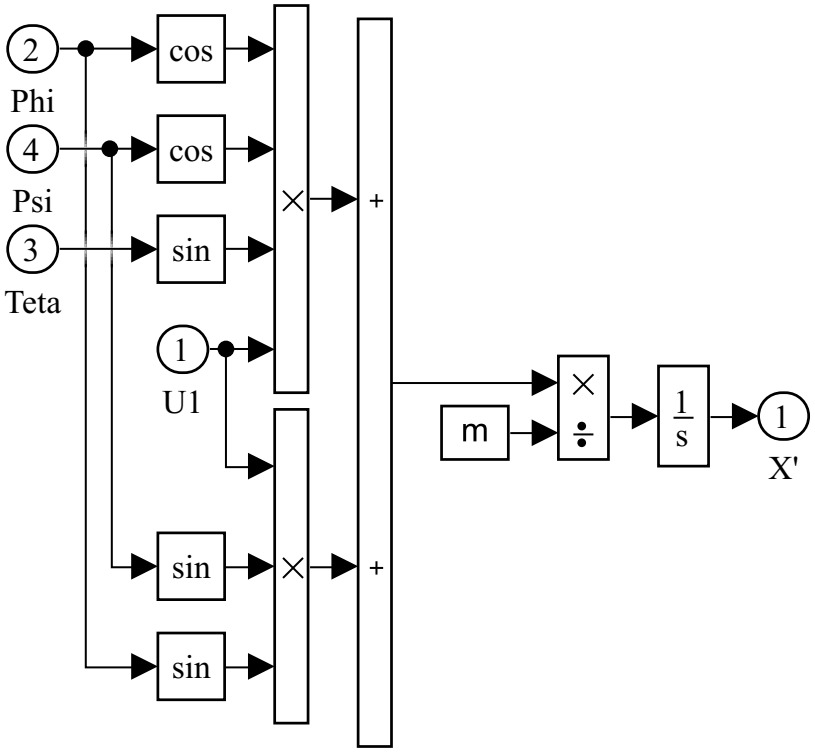


Рисунок А.13 — Структура блока Equations (X)

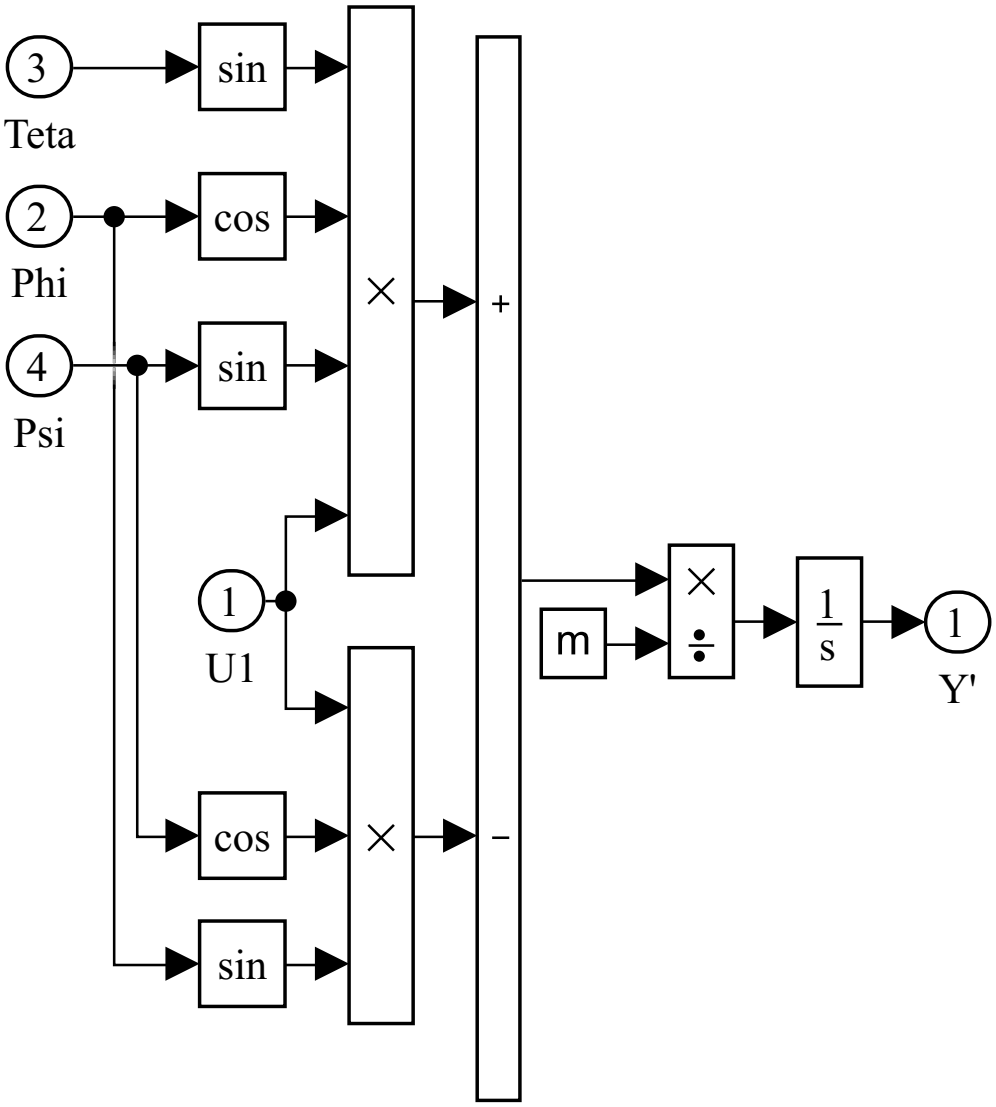


Рисунок А.14 — Структура блока Equations (Y)

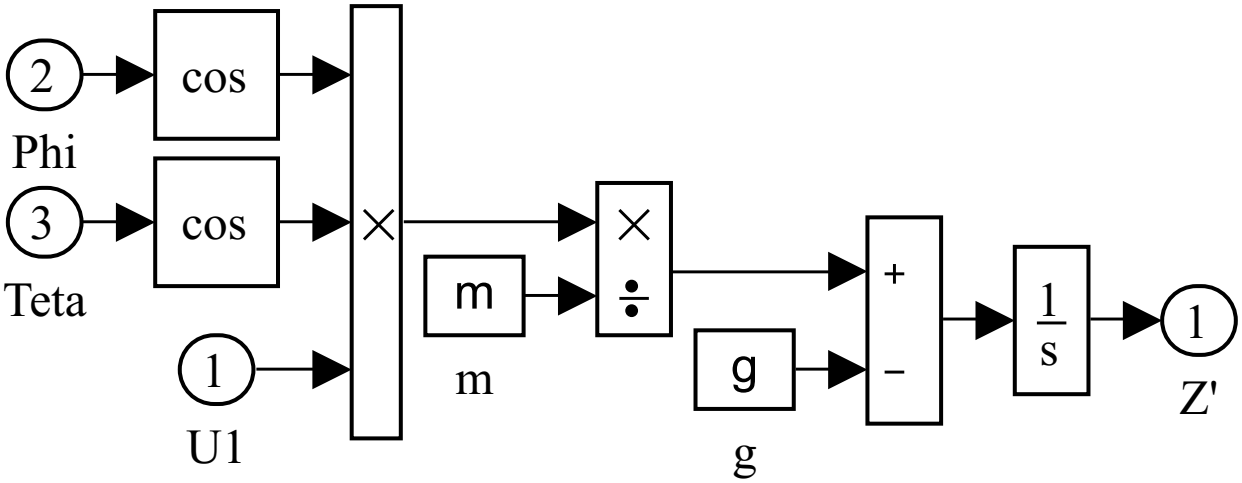


Рисунок А.15 — Структура блока Equations (Z)

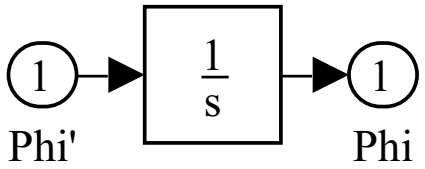
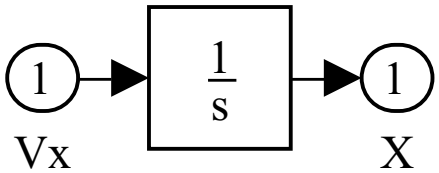
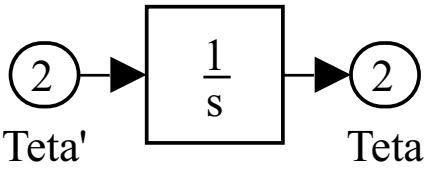
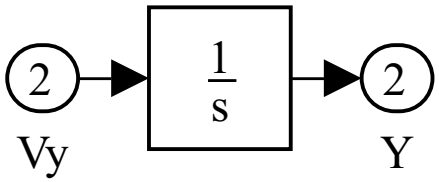
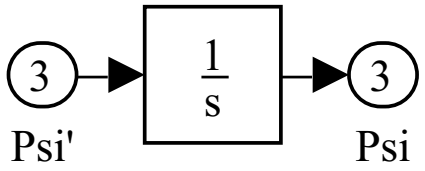


Рисунок А.16 — Структура блока Phi,Psi,Theta









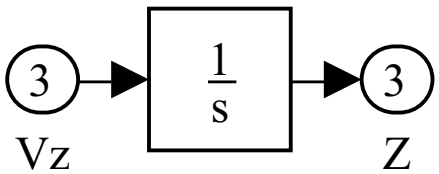


Рисунок А.17 — Структура блока X,Y,Z

Приложение Б

Моделирование аварийных ситуаций и случаев с внедрением методологии безопасной посадки

Б.1 Скрипт запуска симуляции с последующим построением графиков аварии

```
clear
clc

%% Задаем параметры модели
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 симуляцию (для аварийной ситуации)
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');

%% Строим графики отслеживаемых величин
print_Figmod(res,'Po')
```

Б.2 Скрипт запуска симуляции аварии с ручной посадкой с последующим построением графиков

```
clear
clc

%% Задаем параметры модели
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;

%% Запускаем симуляцию (для аварийной ситуации с ручным спасен
    ием)
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');

%% Строим графики отслеживаемых величин
print_Figmod(res,'Povzp')
```

Б.3 Скрипт запуска симуляции с отказоустойчивым алгоритмом с внедрением ПИД регуляторов и построения графиков

```
clear
clc

%% Задаем параметры модели
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;
```

```
%% Задаем коэффициенты ПИД регулятора
PID=[50 8 20];

%% Запускаем симуляцию (для аварийной ситуации с автоматическим спасением)
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');

%% Строим графики отслеживаемых величин
print_Figmod(res,'Povzp')
```

Б.4 Вспомогательная функция вывода графиков print_Figmod

```
function print_Figmod(res,names) %на входе структура с данными и переменная-строка (для выбора нужных графиков)
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

%% График аварийного движения
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('Траектория перемещения аппарата')
    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

%% График скоростей оборотов винтов
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('Угловые скорости \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

%% Строим графики 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('График скорости 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('График высоты 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

```

```
%% График угла 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('График угла \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
```

Б.5 Скрипт для построения множества траекторий перемещения аппарата при разных Ω_i^{em}

```
clear
clc

%% Задаем параметры модели
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));

%% Задаем коэффициенты ПИД регулятора
PID=[50 8 20];

%% Задаем матрицу начальных данных (генерация набора параметро
   в)
X0 = x0;
Y0 = y0;
Z0 = z0;
omega_em = linspace(0,300,101);
om_l = size(omega_em,2);

%% Запускаем симуляцию для каждого случая (перебор)
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;

%% Строим графики отслеживаемых величин
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('Множество траекторий перемещения с различным значением
   \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';

%% График конечных точек
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('Множество точек приземления с различным значением \
   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';
```

Б.6 Скрипт для построения множества траекторий перемещения аппарата при разных Ω_i^{em}, $t_{reaction}$

```
clear
clc

%% Задаем параметры модели
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));

%% Задаем коэффициенты ПИД регулятора
PID=[50 8 20];

%% Задаем матрицу начальных данных (генерация набора параметро
   в)
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_l = size(time_r,2);

%% Запускаем симуляцию для каждого случая (перебор)
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;

%% Строим графики отслеживаемых величин
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('Множество траекторий перемещения с различным значением
   \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');

%% Выбираем множества точек, соответствующих конкретному време
   ни переключения 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

%% График конечных точек
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('Множество точек приземления с различным значением \
   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');
```